



**COVID-19 OLASI VAKALARININ DERİN
ÖĞRENME TEKNİKLERİ KULLANARAK
TESPİTİ**

Çinare OĞUZ

Danışman Dr. Öğr. Üyesi Mete YAĞANOĞLU
Yüksek Lisans Tezi
Bilgisayar Mühendisliği Ana Bilim Dalı
2021
(Her hakkı saklıdır.)

T.C.
ATATÜRK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANA BİLİM DALI

**COVID 19 OLASI VAKALARININ DERİN ÖĞRENME TEKNİKLERİ
KULLANARAK TESPİTİ**

(Determination of Covid-19 Possible Cases by Using Deep Learning Techniques)

YÜKSEK LİSANS TEZİ

Çinare OĞUZ

Danışman: Dr. Öğr. Üyesi Mete YAĞANOĞLU

Erzurum
Mayıs, 2021

KABUL VE ONAY TUTANAĞI

Tez yazarının adını girmek için tıklayın tarafından hazırlanan “Tezin başlığını girmek için tıklayın” başlıklı çalışması .. / .. / 20.. tarihinde yapılan tez savunma sınavı sonucunda başarılı bulunarak jürimiz tarafından Ana Bilim Dalı adını girmek için tıklayın Ana Bilim Dalı, Bilim Dalı adını girmek için tıklayın Bilim Dalında Tez türünü giriniz. Ör. yüksek lisans veya doktora tezi olarak kabul edilmiştir.

Jüri Başkanı: Unvan Ad SOYAD
Üniversite Adı

Danışman: Unvan Ad SOYAD
Üniversite Adı

Jüri Üyesi: Unvan Ad SOYAD
Üniversite Adı

Bu tezin Atatürk Üniversitesi Lisansüstü Eğitim ve Öğretim Yönetmeliği'nin ilgili maddelerinde belirtilen şartları yerine getirdiğini onaylarım.

Unvan Ad SOYAD

Enstitü Müdürü

Not: Bu tezde kullanılan özgün ve başka kaynaklardan yapılan bildiriş, çizelge, şekil ve fotoğrafların kaynak olarak kullanımı, 5846 sayılı Fikir ve Sanat Eserleri Kanunundaki hükümlere tabidir.

**FEN BİLİMLERİ ENSTİTÜSÜ**Graduate School of Natural and
Applied Sciences**T.C.
ATATÜRK ÜNİVERSİTESİ REKTÖRLÜĞÜ
FEN BİLİMLERİ ENSTİTÜSÜ MÜDÜRLÜĞÜ****ETİK BİLDİRİM VE İNTİHAL BEYAN FORMU**

Yüksek Lisans Tezi olarak Dr. Öğr. Üyesi Mete YAĞANOĞLU danışmanlığında sunulan “COVID-19 Olası Vakaların Derin Öğrenme Teknikleri Kullanarak Tespiti ” başlıklı çalışmanın tarafımızdan bilimsel etik ilkelere uyularak yazıldığını, yararlanılan eserlerin kaynakçada gösterildiğini, Fen Bilimleri Enstitüsü tarafından belirlenmiş olan Turnitin Programı benzerlik oranlarının aşılmadığını ve aşağıdaki oranlarda olduğunu beyan ederiz.

Tez Bölümleri	Tezin Benzerlik Oranı (%)	Maksimum Oran (%)
Giriş	7	30
Kuramsal Temeller	20	30
Materyal ve Yöntem	9	35
Bulgular ve Tartışma	3	20
Sonuçlar ve Öneriler	5	20
Tezin Geneli	10	25

Not: Yedi kelimeye kadar benzerlikler ile Başlık, Kaynakça, İçindekiler, Teşekkür, Dizin ve Ekler kısımları tarama dışı bırakılabilir. Yukarıdaki azami benzerlik oranları yanında tek bir kaynaktan olan benzerlik oranlarının %5'den büyük olmaması gerekir.

Sunulan bilgilerin doğru olduğunu, aksi halde doğacak hukuki sorumlulukları kabul ettiğimizi beyan ederiz.

Tez Yazarı (Öğrenci)	Tez Danışmanı
Çinare OĞUZ	Dr. Öğr. Üyesi Mete YAĞANOĞLU
21.6.2021	21.6.2021
İmza:	İmza:

* Tez ile ilgili YÖKTEZ'de yayınlamasına ilişkin bir engelleme var ise aşağıdaki alanı doldurunuz.

☐ Tezle ilgili patent başvurusu yapılması / patent alma sürecinin devam etmesi sebebiyle Enstitü Yönetim Kurulunun/....../.... tarih ve sayılı kararı ile teze erişim 2 (iki) yıl süreyle engellenmiştir.

☐ Enstitü Yönetim Kurulunun/....../.... tarih ve sayılı kararı ile teze erişim 6 (altı) ay süreyle engellenmiştir.

Not: Bu form, Tezin son şekline uygun olarak bilgisayar ortamında doldurulmalı, çıktısı imzalanıp Tezin sonuna eklenmelidir.

TEŞEKKÜR

Tez çalışmam süresince yardımını esirgemeyen tez danışmanım Sayın Dr. Öğr. Üyesi Mete YAĞANOĞLU'na içtenlikle teşekkürlerimi sunarım.

COVID-19 konusunda bilgisinden yararlandığım Sayın Op. Dr. Hasay GULİYEV'e teşekkürlerimi sunarım.

Tez çalışmamda kullandığım Bilgisayarlı Tomografi görüntülerini sağladığı için Siirt Eğitim ve Araştırma Hastanesi'ne teşekkürlerimi sunarım.

Her türlü konuda daima bana destek olan ve beni asla yalnız bırakmayan sevgili eşime ve aileme en içten teşekkürlerimi sunarım.

Çinare OĞUZ



ÖZET

YÜKSEK LİSANS TEZİ COVID 19 OLASI VAKALARININ DERİN ÖĞRENME TEKNİKLERİ KULLANARAK TESPİTİ

Çinare OĞUZ

Danışman: Dr. Öğr. Üyesi Mete YAĞANOĞLU

Amaç: COVID-19 tanısında kullanılan PCR testinin sonuçlanma sürecinde hasta karantinaya alınmadığı için hastalığı yaymaya devam etmektedir. Bu çalışmada Bilgisayarlı Tomografi kullanımı ile COVID-19 şüphesi nedeniyle hastaneye başvuran hastaların tanı süresini kısaltarak hastalığın bulaşma süresini ve miktarını azaltmak amaçlanmıştır. Ayrıca COVID-19 tespitinde radyoloji uzmanlara karar destek sistemi sağlamak amaçlanmıştır.

Yöntem: Bu çalışmada Siirt Eğitim ve Araştırma Hastanesinin radyografi veri tabanından elde edilen 1345 tane BT görüntüsü üzerinde ResNet-50, ResNet-101, AlexNet, Vgg-16, Vgg-19, GoogLeNet, SqueezeNet, Xception gibi derin öğrenme modelleri ile derin özellikler çıkarıldı. Bu derin özellikler Destek Vektör Makinesi(DVM), k en yakın komşu (kNN), Rastgele Orman (RO), Karar Ağaçları (KA) ve Naive Bayes (NB) gibi sınıflandırma yöntemlerine verilerek test görüntüleriyle performansları değerlendirildi.

Bulgular: Başarı kriteri olarak doğruluk değeri, F1-skor ve ROC eğrisi ele alınmıştır. Uygulama sonucunda elde edilen verilere göre en iyi performans ResNet-50 ve DVM yöntemi ile elde edildi. Doğruluk değeri %96,296, F1-skor %95,868 ve AUC değeri 0,9821 olarak hesaplandı.

Sonuç: Bu çalışmada incelenen ve performansı yüksek bulunan derin öğrenme modeli ve sınıflandırma metodu COVID-19 hastalığı için gereksiz testleri önleyerek yardımcı karar destek sistemi olarak kullanılabilir.

Anahtar Kelimeler: derin öğrenme, COVID-19, sınıflandırma, ResNet, destek vektör makinesi.

Nisan 2021, 85 sayfa

ABSTRACT

MASTER'S THESIS

DETERMINATION OF COVID 19 POSSIBLE CASES BY USING DEEP LEARNING TECHNIQUES

Çinare OĞUZ

Supervisor: Assist. Prof. Dr. Mete YAĞANOĞLU

Purpose: Since the patient is not quarantined during the conclusion of the PCR test used in the diagnosis of COVID 19, the disease continues to spread. In this study, it was aimed to reduce the duration and amount of transmission of the disease by shortening the diagnosis time of patients admitted to the hospital due to the suspicion of COVID 19 with the use of Computed Tomography. In addition, it is aimed to provide radiology experts with a decision support system in the detection of COVID-19.

Method: In this study, deep features were extracted with deep learning models such as ResNet-50, ResNet-101, AlexNet, Vgg-16, Vgg-19, GoogLeNet, SqueezeNet, Xception on 1345 CT images obtained from the radiography database of Siirt Training and Research Hospital. These deep features are given to classification methods such as Support Vector Machine (SVM), k Nearest Neighbor (kNN), Random Forest (RF), Decision Trees (DT), Naive Bayes (NB), and their performance is evaluated with test images.

Findings: Accuracy value, F1-score and ROC curve were considered as success criteria. According to the data obtained as a result of the application, the best performance was obtained with ResNet-50 and DVM method. The accuracy was 96,296%, the F1-score was 95,868%, and the AUC value was 0,9821.

Results: The deep learning model and classification method examined in this study and found to be high performance can be used as an auxiliary decision support system by preventing unnecessary tests for COVID-19 disease.

Keywords: deep learning, COVID-19, classification, ResNet, support vector machine .

April 2021, 85 pages

İÇİNDEKİLER

KABUL VE ONAY TUTANAĞI.....	i
ETİK BİLDİRİM VE İNTİHAL BEYAN FORMU	ii
TEŞEKKÜR	iii
ÖZET	iv
ABSTRACT	v
İÇİNDEKİLER.....	vi
TABLolar DİZİNİ.....	viii
ŞEKİLLER DİZİNİ	ix
KISALTMALAR VE SİMGELER DİZİNİ.....	xi
GİRİŞ.....	1
KURAMSAL TEMELLER.....	4
Temel Kavramlar	4
Yapay sinir ağları	5
Derin öğrenme.....	7
Bilgisayarlı Torax Tomografi Görüntüsü.....	10
Literatür Çalışması	11
MATERYAL VE YÖNTEM	19
Materyal	19
Veri seti	19
Yöntem.....	20
Derin öğrenme modelleri	20
Sınıflandırma yöntemleri	32
ARAŞTIRMA BULGULARI VE TARTIŞMA.....	42
Performans Metrikleri	42
ResNet-101 deney sonuçları	43
ResNet-50 Deney Sonuçları.....	45
AlexNet Deney Sonuçları	47
Vgg-19 Deney Sonuçları.....	49
Vgg-16 Deney Sonuçları.....	51
GoogLeNet deney sonuçları.....	53
SqueezeNet Deney Sonuçları	55
Xception Deney Sonuçları	57
SONUÇLAR VE ÖNERİLER	62
KAYNAKLAR.....	65

EKLER	71
EK 1. Atatürk Üniversitesi Girişimsel Olmayan Klinik Araştırmalar Etik Kurulu Onayı	71
ÖZGEÇMİŞ.....	72



TABLÖLAR DİZİNİ

Tablo 1. Veri Setindeki Eğitim ve Test Setine Ayrılan Görüntü Sayısı	20
Tablo 2. ResNet Katmanları.	24
Tablo 3. Karışıklık Matrisi.....	42
Tablo 4. Resnet-101 Deney Sonuçları	45
Tablo 5. ResNet-50 Deney Sonuçları	46
Tablo 6. AlexNet Deney Sonuçları.....	49
Tablo 7. Vgg-19 Deney Sonuçları	51
Tablo 8. Vgg-16 Deney Sonuçları	52
Tablo 9. GoogLeNet Deney Sonuçları	55
Tablo 10. SqueezeNet Deney Sonuçları	57
Tablo 11. Xception Deney Sonuçları.....	59
Tablo 12. Deney Sonuçları	60
Tablo 13. ResNet-50 ile Sınıflandırma Deney Sonuçları	61
Tablo 14. Önerilen Modelin BT Görüntüleri Kullanan Çalışmalarla Karşılaştırılması	63

ŞEKİLLER DİZİNİ

Şekil 1. Dünya’da COVID-19 vaka dağılımı.	2
Şekil 2. Asya’da COVID-19 vakalarının en fazla görüldüğü ülkelerdeki toplam vakalar.....	2
Şekil 3. (Solda) COVID 19 virüsüyle enfekte olmuş bir bireyin akciğerinin BT görüntüsü ve (sağda) sağlıklı bir bireyin akciğerinin BT görüntüsü.....	3
Şekil 4. Yapay zekâ venn şeması	5
Şekil 5. Yapay sinir ağı mimarisi	7
Şekil 6. Derin öğrenme mimarisi.	8
Şekil 7. Veri setinden örnek görüntüler.....	20
Şekil 8. AlexNet ağ yapısı.....	22
Şekil 9. ResNet kısa yol bağlantıları.	23
Şekil 10. ResNet ağ yapısı.....	25
Şekil 11. Vgg ağ yapısı.	26
Şekil 12. Inception modülü.	27
Şekil 13. GoogLeNet ağ yapısı.	28
Şekil 14. Yangın modülü.....	29
Şekil 15. SqueezeNet ağ yapısı.	30
Şekil 16. Xception ağ yapısı.....	32
Şekil 17. Destek vektörleri ve ayırım düzlemi.	34
Şekil 18. Doğrusal olmayan DVM.....	35
Şekil 19. Karar ağacı.	36
Şekil 20. Gaussian NB yöntemi	39
Şekil 21. Çalışmanın blok diyagramı.	41
Şekil 22. ResNet-101 ile sınıflandırma yöntemlerinin doğruluk oranı.	44
Şekil 23. ResNet-101 ROC-eğrisi	45
Şekil 24. ResNet-50 ile sınıflandırma yöntemlerinin doğruluk oranı.	46
Şekil 25. ResNet-50 ROC-eğrisi	47
Şekil 26. AlexNet ile sınıflandırma yöntemlerinin doğruluk oranı.	48
Şekil 27. AlexNet ROC-eğrisi.....	49
Şekil 28. Vgg-19 ile sınıflandırma yöntemlerinin doğruluk oranı.	50
Şekil 29. Vgg-19 ROC-eğrisi	51
Şekil 30. Vgg-16 ile sınıflandırma yöntemlerinin doğruluk oranı.	52
Şekil 31. Vgg-16 ROC-eğrisi	53
Şekil 32. GoogLeNet ile sınıflandırma yöntemlerinin doğruluk oranı.	54

Şekil 33. GoogLeNet ROC-eğrisi	55
Şekil 34. SqueezeNet ile sınıflandırma yöntemlerinin doğruluk oranı.	56
Şekil 35. SqueezeNet ROC-eğrisi	57
Şekil 36. Xception ile sınıflandırma yöntemlerinin doğruluk oranı.....	58
Şekil 37. Xception ROC-eğrisi.....	59
Şekil 38. ResNet-50 eğitim ve test süreci	61



KISALTMALAR VE SİMGELER DİZİNİ

BT	: Bilgisayarlı Tomografi
CoV	: Korona Virüs
ÇZ	: Çalışma Zamanı
DSÖ	: Dünya Sağlık Örgütü
DVM	: Destek Vektör Makinesi
ESA	: Evrişimli Sinir Ağı
ILSVRC	: ImageNet Büyük Ölçekli Görsel Tanıma
KA	: Karar Ağaçları
kNN	: k En Yakın Komşu
NB	: Naive Bayes
nCoV	: Yeni Nesil Korona Virüs
PCR	: Poliimeraz Zincir Reaksiyonu
RO	: Rastgele Orman

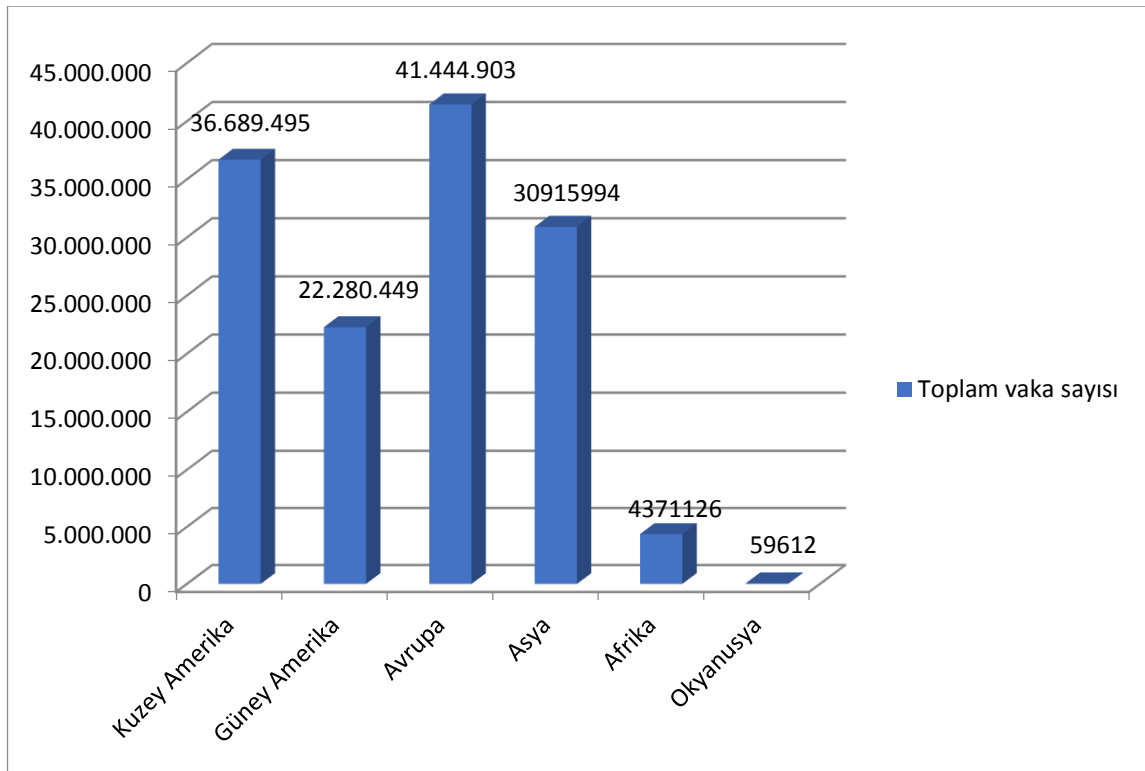
GİRİŞ

Dünya Sağlık Örgütü'nün (DSÖ) belirttiği gibi koronavirüsler (CoV), soğuk algınlığı gibi toplumda yaygın görülen hafif enfeksiyondan; daha ciddi enfeksiyona neden olabilen büyük bir virüs ailesidir. 2019 yılının son aylarında Çin'in Hubei eyaletinin başkenti Vuhan şehrinde etiyojisi bilinmeyen pnömoni vakaları tespit edilmiştir. Daha sonra insanlarda görülmemiş yeni bir koronavirüs (2019- nCoV) olduğu açıklandı. 11 Şubat 2020'de DSÖ'nün Genel Müdürü Dr. Tedros Adhanom Ghebreyesus, bu virüsü "2019 korona virüs hastalığı"nın kısaltması olan "COVID-19" olarak adlandırdı. COVID-19 yeni bir virüsün neden olduğu iltihaplı bir hastalıktır. Hastalık, daha ciddi vakalarda soğuk algınlığı, öksürük, ateş gibi semptomlarla birlikte solunum sıkıntısına neden olmaktadır (Sujath *et al.* 2020).

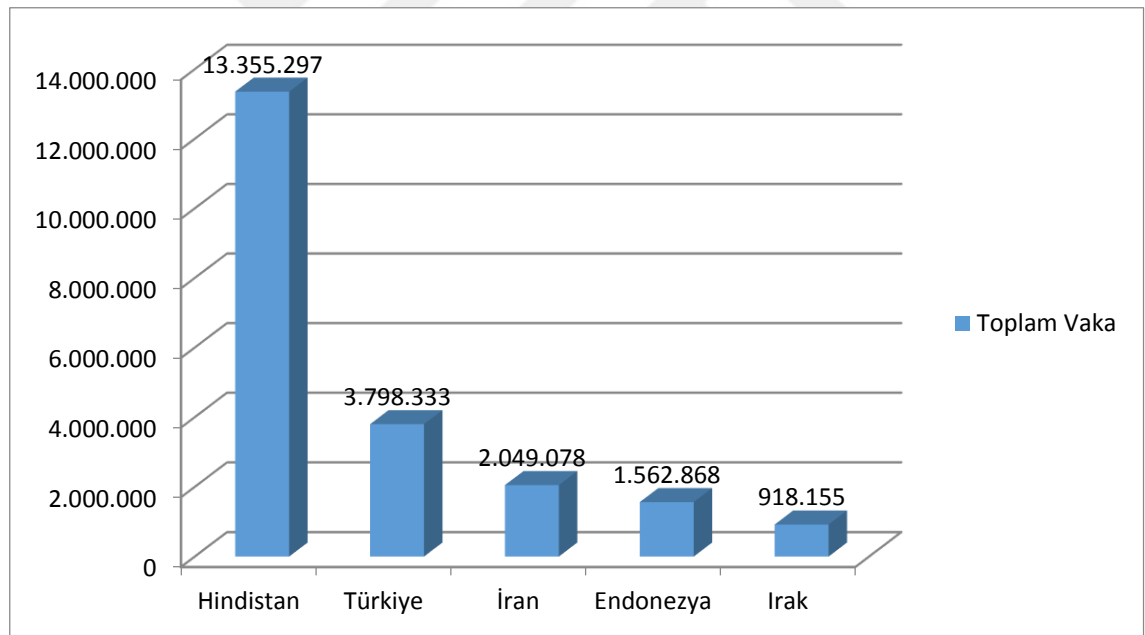
DSÖ'nün verilerine göre bu hastalık nedeni ile kişisel koruyucu ekipman ve solunum cihazlarına talep artmıştır. Virüsün yayılmasını takip eden süreçte maske kullanımı normal seviyelere göre 100 kat artmış ve ilgili materyallerin fiyatları yaklaşık 20 kat yükselmiştir (Mahase 2020).

10 Nisan 2021 tarihi itibarıyla dünya genelinde 130 milyondan fazla vaka ve yaklaşık 3 milyon ölüm bildirilmiştir (Anonymous 2021a). Şekil 1'de dünyada COVID-19 vakalarının kıtalara göre dağılımı gösterilmektedir. Şekilde görüldüğü üzere Kuzey Amerika ve Avrupa kıtalarından sonra vakaların en fazla görüldüğü kıta Asya kıtasıdır. Şekil 2'de Asya kıtasında vakaların en fazla görüldüğü 5 ülke verilmiş olup Türkiye ikinci sıradadır.

Türkiye'de ilk COVID-19 vakası 11 Mart 2020 tarihinde görülmüş olup, virüsün yayılmasıyla birlikte bir yıl içerisinde 4 milyona yakın vaka tespit edilmiştir. COVID-19 nedeniyle 30 binden fazla kişi hayatını kaybetmiştir (Anonymous 2021a).



Şekil 1. Dünya’da COVID-19 vaka dağılımı (Anonymous 2021a).



Şekil 2. Asya’da COVID-19 vakalarının en fazla görüldüğü ülkelerdeki toplam vakalar (Anonymous 2021a).

COVID-19’un tespiti için henüz %100 kesin tanı sağlayabilen altın standart bir test olmadığından hızlı antijen testi, Polimeraz Zincir Reaksiyonu (PCR) testi yapılarak ve Bilgisayarlı Tomografi (BT) görüntüsü alınarak tanıya ulaşılmaya çalışılmaktadır.

COVID-19 şüphesi nedeniyle hastaneye başvuran kişilerin ilk değerlendirme sonucunda sağlık bakanlığı tarafından belirtilen vaka tanımına uyan hastalar daha önceden

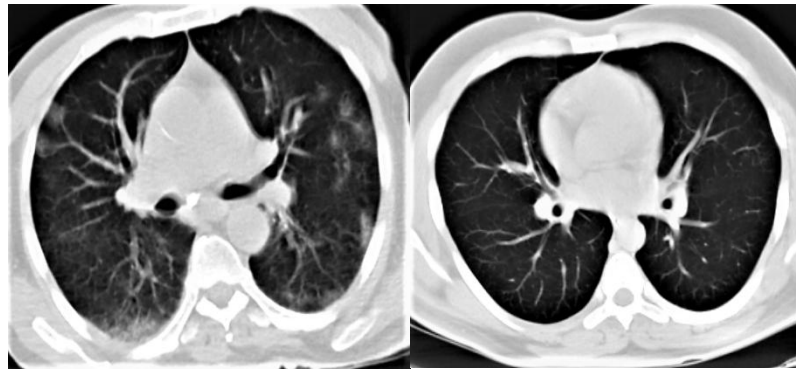
belirlenen izolasyon alanına alınmaktadır. Ardından muayenesi yapılarak vital bulgularına bakılmaktadır.

COVID-19 test kitlerinin sınırlı sayıda olması nedeniyle şüpheli tüm COVID-19 hastalarına bu test kitleri uygulanmamaktadır. Bu yüzden COVID-19 ile enfekte olduğu şüphelenilen bireylerde tanı amaçlı ilk yapılan tekniklerden biri "Bilgisayarlı Toraks Tomografi" dir. Çekilen BT'deki pozitif bulgular neticesinde hastaya PCR testi yapılmaktadır. Böylece test kitlerini verimli bir şekilde kullanılması sağlanmaktadır. BT cihazı hemen hemen tüm hastanelerde mevcuttur, özel test kitleri olmadan COVID-19'u test etmek için BT görüntülerini kullanmak mümkün olabilmektedir.

COVID-19, solunum yolumuzu örten epitel hücrelerine saldırdığından, hastanın akciğerlerinin tutulumunu değerlendirmek için BT kullanılabilir. Doktorlar pnömoni, akciğer iltihabı, apseler ve genişlemiş lenf düğümlerini teşhis etmek için sıklıkla BT görüntülerini kullanmaktadır.

Şekil 3'de COVID-19 virüsü ile enfekte bir bireyin akciğerinin BT görüntüsü ve sağlıklı bir bireyin akciğerinin BT görüntüsü gösterilmektedir.

Çok hızlı yayılan COVID-19; pnömoni, böbrek yetmezliği gibi ciddi organ fonksiyon bozukluğuna neden olmakla birlikte; bazen de ölüm ile sonuçlanmaktadır. Dolayısıyla COVID-19 erken teşhis hayati önem taşımaktadır. COVID-19 tanısında kullanılan PCR testinin sonuçlanma sürecinde hasta karantinaya alınmadığı için hastalığı yaymaya devam etmektedir. Bu çalışmada Bilgisayarlı Tomografi kullanımı ile COVID-19 şüphesi nedeniyle hastaneye başvuran hastaların tanı süresini kısaltarak hastalığın bulaşma süresini ve miktarını azaltmak amaçlanmıştır. Ayrıca COVID-19 tespitinde radyoloji uzmanlara karar destek sistemi sağlamak amaçlanmıştır.

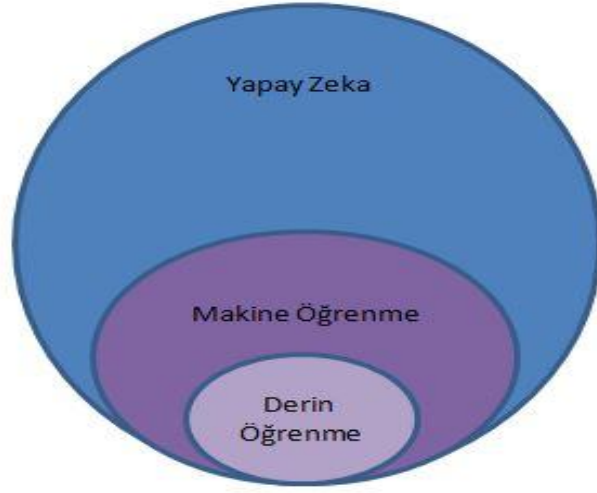


Şekil 3. (Solda) COVID 19 virüsüyle enfekte olmuş bir bireyin akciğerinin BT görüntüsü ve (sağda) sağlıklı bir bireyin akciğerinin BT görüntüsü

KURAMSAL TEMELLER

Temel Kavramlar

Yapay zekâ, başlangıçtan bu yana bir makine tarafından herhangi bir zekâ özelliğini taklit etmeyi amaçlamış bir bilgisayar programıdır. Yapay zekâ, sinir ağlarının bir bileşiminden oluşmuştur (Kutsurelis 1998). Kökenlerini saptamak zor olmakla birlikte, yapay zekânın tarihi Aristo'ya kadar uzandığı düşünülmektedir. Modern anlamda yapay zekâ İngiliz matematikçi Turing (1950) tarafından 1950 yılında, akıllı makinelerin nasıl yaratılacağını ve özellikle de zekâlarının nasıl test edileceğini anlattığı makalesinde ortaya konulmuştur (Turing 1950). Turing Testi bugün hala bir yapay modelin zekâsını belirlemek için bir ölçüt olarak kabul edilmektedir. Eğer bir insan, makine ile etkileşime girdiğinde makineyi insan olarak algılıyorsa, bu makinenin akıllı olduğu söylenmektedir. 1964-1966 yılları arasında Joseph Weizenbaum, insanlarla konuşmayı taklit eden doğal bir dil işleme aracı olan bilgisayar programı geliştirmiştir. Bu program Turing Testi üzerinde test edilmiş ilk programlardan biriydi. Yapay zekâ ile ilgili ilk çalışma, 1956'da yapay sinir hücreleri, fizyoloji ve Turing'in hesaplama kavramından yararlanan mantık modellemeye dayalı bir hesaplama modeli aracılığıyla McCulloch ve Pitts tarafından gerçekleştirildi. 1973'te ABD Kongresi, yapay zekâ araştırmalarına yapılan yüksek harcamaları eleştirmeye başladı. Böylece yapay zekâ duraklama dönemine girdi. Bu dönem yapay zekâ kışı olarak adlandırıldı. Google tarafından geliştirilen bir program olan AlphaGo, 2015 yılında Go tahta oyununda dünya şampiyonunu yenmeyi başardıktan sonra yapay zekâ Derin Öğrenme biçiminde bir geri dönüş yaptı (Silver *et al.* 2016). Makine öğrenme bir yapay zekâ yöntemidir (Kim 2017). Aralarındaki ilişki Şekil 4'te gösterilmektedir.



Şekil 4. Yapay zekâ venn şeması (Anonymous 2021b).

Makine öğrenimi terimi ilk kez 1959'da, Arthur Samuel tarafından ortaya atıldı (Samuel 1959). 1960'larda makine öğreniminde tahmini çıkarım için Bayes yöntemleri tanıtıldı. 1980'lerde Geri yayımlı sinir ağlarının keşfi, makine öğrenme araştırmalarında yeniden canlanmaya neden oldu.

Makine öğrenimi, karmaşık verilerin otomatik olarak çıkarılması ve analizi yoluyla açık programlama olmadan öğrenmesine olanak sunmaktadır (Silver *et al.* 2016). Makine öğrenimi teknikleri, eğitim yöntemlerine bağlı olarak üç gruba ayrılmaktadır:

Denetimli öğrenme: Makineye bir "öğretmen" tarafından girdi ve doğru çıktı çiftleri verilmektedir. Doğru çıktı, modelin verilen girdi için üretmesi gereken çıktıdır (Kim 2017). Böylece girdileri çıktılarla eşleştiren genel bir kural öğretilmektedir.

Denetimsiz öğrenme: Genellikle verilerin özelliklerini araştırmak ve verileri önceden işlemek için kullanılmaktadır. Öğrenme sürecinde eğitim verileri yalnızca doğru çıktıları olmayan girdileri içermektedir. Girdi verilerine göre çıkan benzer sonuçlar kümelenmekte ve her küme bir sınıfı belirlemektedir.

Takviyeli öğrenme: Bir bilgisayar programına, girdilere göre herhangi bir çıktı değeri verilmemektedir. Sonuç olarak girdi verisine göre elde edilen çıktı bir öğretmen tarafından kontrol edilerek doğru veya yanlış sonuç ürettiğini bildirmektedir. Yanlış sonuç elde edilmişse tekrar güncelleme yapmaktadır (Kim 2017). Makine öğrenme modellerine yapay sinir ağları, kNN, KA, RO ve DVM örnek olarak verilebilir.

Yapay sinir ağları

Biyolojik sinir ağları temel alınarak yapay sinir ağları geliştirilmiştir. Yapay sinir ağlarını anlamak için biyolojik sinir ağlarının yapısını ve nasıl çalıştığını bilmek gerekir.

Biyolojik sinir ağıları "sinir hücresi" olarak da bilinen nöronların, diğer hücelere hızlı ve hassas bir şekilde sinyal göndermelerini sağlayan özel yapılardır. Bu sinyaller, akson adı verilen ince lifler boyunca ilerleyen elektrokimyasal dalgalar şeklinde gönderilmektedir. Kimyasalların sinaps adı verilen bağlantı noktalarında salınmasıyla sinyaller beyne iletilmektedir. Beyne iletilen sinyaller (veriler) sayesinde öğrenme gerçekleşmektedir. Nöronlar arasındaki bağlantılar; sinir yolları, sinir devreleri ve daha büyük ağları oluşturabilmektedir (Anonymous 2021c).

Biyolojik sinir sistemi; bilgiyi alan, yorumlayan ve buna göre kararlar veren bir kontrol merkezidir. Nöronlar, bu bilgiyi işleme ve iletme gibi özel yeteneklerle donatılmış çok özel hücrelerdir (Staub *et al.* 2011).

Yapay sinir ağıları, insan beynindeki nöronlarla aynı şekilde bir araya getirilen yapay nöronlar aracılığıyla numuneler hakkında bilgi toplayan sistemlerdir. Problemlerle karşılaştıklarında öğrendiklerini kullanarak karar verme yeteneğine sahiptirler. Yapay sinir ağıları, insan beyninin iki şekilde modellenmesiyle geliştirilmektedir. Birinci olarak ağlar aracılığıyla bilgi elde edilmektedir. İkinci olarak, nöronlar arasındaki bağlantı ile bilgi depolanmaktadır. Ağ yapıları oluşturulduktan sonra görsel girdiler sınıflandırılmaktadır (Simard *et al.* 2003). İlk yapay sinir ağı McCulloch ve Pitts tarafından insan beyninin hesaplama kapasitesine dayanarak elektrik devreli bir sinir ağı olarak ortaya konuldu.

Yapay sinir ağıları; giriş, gizli ve çıktı katmanlarından oluşmaktadır. Tek bir gizli katmana sahip sinir ağına basit sinir ağı veya vanilya sinir ağı denir. Şekil 5'de basit bir yapay sinir ağı örneği verilmiştir. Birden fazla gizli katmandan oluşan çok katmanlı bir sinir ağına derin sinir ağı denir (Kim 2017).

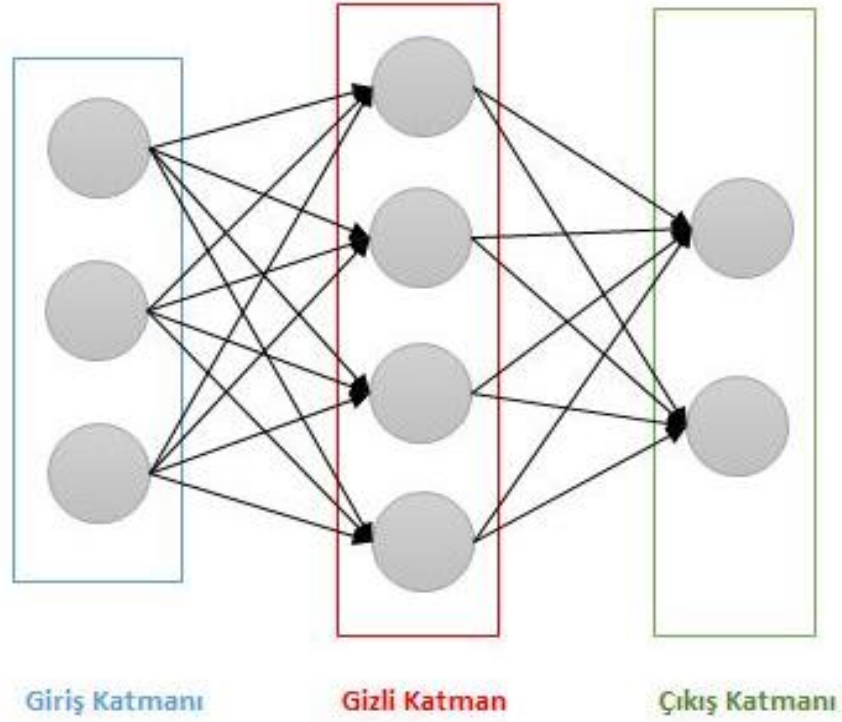
Giriş Katmanı: Veri kümesinin ağa tanıtıldığı katmandır. Giriş katmanı, giriş sinyallerini sonraki katmanlara iletmektedir. Bu nedenle, ağırlık toplamı ve aktivasyon fonksiyonu hesaplanmamaktadır.

Gizli Katman: Giriş ve çıkış katmanları arasındaki katmanlara gizli katmanlar denir. Sinir ağının dışından erişilemedikleri için onlara bu isim verilmiştir (Kim 2017). Giriş katmanından gelen girdilere ağırlık uygulanmakta ve bir aktivasyon fonksiyonu aracılığıyla bunları çıktı katmanına yönlendirilmektedir.

Çıktı katmanı: Şekil 5'de en sağdaki düğümler grubuna çıktı katmanı denir (Kim 2017). Öğrenme bu katmanda gerçekleşmektedir. Çıktı katmanı, kendisinden önceki katmanlardan geçirilen girdileri alarak, nöronları aracılığıyla hesaplamaları yapmaktadır ve

ardından çıktı hesaplanmaktadır. Bu katmandaki düğümlerden gelen çıktı, sinir ağının nihai sonucu olmaktadır.

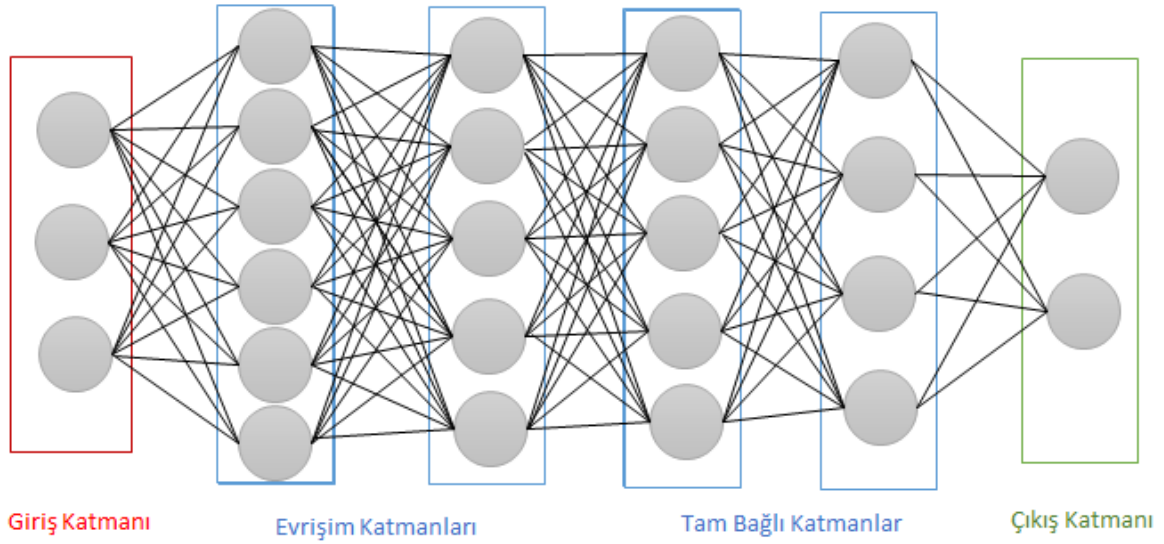
Yapay sinir ağlar; ileri beslemeli ağlar, geri bildirim ağları, bellek tabanlı ağlar ve radyal tabanlı ağlar gibi mimari yapılarına bağlı olarak gruplara ayrılabilir.



Şekil 5. Yapay sinir ağı mimarisi (Anonymous 2021c).

Derin öğrenme

Derin öğrenmedeki “derin” kelimesi yapay sinir ağındaki girdi ve çıktı katmanları arasındaki katmanların çok sayıda olmasından kaynaklanmaktadır (Schmidhuber 2015). Şekil 6’da basit bir derin öğrenme mimarisi gösterilmiştir. Derin öğrenme ilk kez Hinton ve Salakhutdinov (2006) makalelerinde sunulmuştur. Genellikle derin öğrenme modelleri Evrimsel Sinir Ağları’na (ESA) dayanmaktadır.



Şekil 6. Derin öğrenme mimarisi.

Derin öğrenme modellerinde, daha iyi sonuçlar elde etmek için büyük miktarda veri gerektirmektedir. Verileri işlerken, derin öğrenme modelleri, karmaşık matematiksel hesaplamalar içermektedir. Derin öğrenmede her katmanda girdi verilerini bileşik bir temsile dönüştürerek öğrenme işlemini gerçekleştirir. Örneğin, bir yüz tanıma programı ilk katmanda yüzün kenarlarını ve çizgilerini, ardından ikinci katmanda gözler gibi yüzün daha önemli kısımlarını ve son olarak görüntünün yüz içerdiğini algılamayı ve tanımayı öğrenerek çalışmaktadır. (Bengio 2013; Hinton 2015).

Makine öğrenmede önceden yapılan ön işleme ve özellik çıkarma gibi işlemler derin öğrenmede katmanlar içerisinde otomatik yapılmaktadır.

Derin öğrenme katmanları

Derin öğrenme modelindeki bir katman, modelin mimarisindeki önceki katmanlardan bilgi alan ve ardından bilgileri bir sonraki katmana aktaran bir yapıdır.

Giriş katmanı

Giriş katmanında, önceden tanımlanmış filtreler ile ön işleme gerçekleştirilmektedir. Böylece, ham girdi görüntüsünün yanı sıra, kenarlar ve gradyanlar gibi ek bilgiler ağa sağlanabilmektedir (Ciresan 2011). Bu yüzden bu katmana görüntü işleme katmanı da denilmektedir.

Evrişim katmanı

Evrişim katmanı, özellik haritaları adı verilen yeni görüntüler üretmektedir. Özellik haritası, orijinal görüntünün benzersiz özelliklerini vurgulamaktadır. Evrişim katmanı, diğer

sinir ağı katmanlarına kıyasla çok farklı bir şekilde çalışmaktadır. Bu katman bağlantı ağırlıkları ve ağırlık toplamı kullanmamaktadır. Bunun yerine, görüntüleri dönüştüren filtreler içermektedir. Bu filtrelere evrişim filtreleri adı verilmektedir (Kim 2017). Filtreleme ile görüntü üzerinde ölçeklendirme yapılmaktadır. Farklı derin öğrenme modellerine göre filtreleme boyutu değişiklik göstermektedir.

Aktivasyon katmanı

Evrişim filtresinin oluşturduğu özellik haritası aktivasyon fonksiyonu aracılığıyla işlenmektedir. Doğrusal olmayan özellikler ağı kazandırılmaktadır. Bu çalışmada kullanılan derin öğrenme modellerinde aktivasyon fonksiyonu olarak RELU aktivasyon fonksiyonundan yararlanılmaktadır. RELU diğer aktivasyon fonksiyonlarından daha az hesaplama yüküne sahiptir. Bu yüzden derin öğrenme modellerinde daha çok tercih edilmektedir. Genellikle ara katmanlarda kullanılmaktadır. RELU aktivasyon fonksiyonu Denklem 1’de gösterilmektedir.

$$f(x) = \begin{cases} x, & x \geq 0 \\ 0, & x < 0 \end{cases} \quad (1)$$

ReLU aktivasyon fonksiyonu ile sıfırdan küçük herhangi bir değer sıfır olarak ayarlanmaktadır. Girişin her ögesine bir eşik işlemi uygulamaktadır.

Havuzlama katmanı

Havuzlama katmanında görüntünün boyutu küçültülmektedir. Bu işlem, görüntünün belirli komşu piksellerini tek bir temsili değere dönüştürülmesi ile gerçekleşmektedir. Havuzlama, diğer birçok görüntü işleme şemasının hâlihazırda kullanmakta olduğu tipik bir tekniktir. Havuzlama katmanındaki işlemleri gerçekleştirmek için, görüntüden havuz piksellerinin nasıl seçileceğini ve temsili değerin nasıl ayarlanacağı belirlenmektedir (Kim 2017). Seçilen piksellerin ortalaması ya da maksimum değeri alınarak temsili değerler ayarlanmaktadır. Havuzlama işlemlerinin en yaygın yöntemleri ortalama ve maksimum havuzlama yöntemleridir.

Tam bağı katman

Önceki katmanlardan elde edilen veriler, bu katmanda tek boyutlu matrise dönüştürülmektedir. Örneğin hangi sınıfa ait olduğunu belirleyen derin özellikler bu katmandan elde edilmektedir (Lin *et al.* 2013)

Dropout katmanı

Dropout katmanının amacı yapay sinir ağının ezberlenmesine engel olmaktır. Dropout tekniği ağın ezberlenmesini önlemek için ağda ezber yapan nöronları belli bir oranda ortadan kaldırılmasından ibarettir. Atılan bu nöronlar geri yayılmaya katılmamaktadır. Yani her girdi ağa sunulduğunda, sinir ağı farklı bir mimariyi örneklemektedir, ancak tüm bu mimariler ağırlıkları paylaşmaktadır. Bu şekilde diğer nöronlar birçok farklı rastgele alt kümesiyle bağlantılı olarak yararlı olan daha sağlam özellikleri öğrenmeye zorlanmaktadır (Krizhevsky 2012).

Sınıflandırma katmanı

Sınıflandırma katmanı, ağın son katmanıdır. Derin öğrenme modellerine göre değişiklik gösterse bile genelde softmax katmanından sonra gelmektedir. Bu katmanın çıktı değeri sınıf sayısına eşittir (Ciresan 2011).

Softmax katmanı

Derin öğrenme modellerinin çıkışındaki olasılık dağılımını sağlamaktadır. Her sınıfa ondalık olasılıklar atamaktadır. Bu ondalık olasılıkların toplamı 1 olmalıdır.

Normalizasyon katmanı

Normalleştirme, genellikle modeli eğitmeden önce verileri hazırlamak için kullanılan bir yöntemdir. Normalleştirmenin temel amacı, sayısal değerler için tek tip bir ölçek sağlamaktır. Veri kümesi çok büyük bir aralıkta değişen sayısal veriler içerirse, öğrenme sürecini çarpıtarak kötü bir modele neden olmaktadır. Normalizasyon katmanı, bilgi kaybının olmamasını ve hatta değer aralığının etkilenmemesini sağlamaktadır (Anonymous 2021d). Bu süreç sayesinde ağın eğitimi daha hızlı gerçekleşmektedir.

Bilgisayarlı Torax Tomografi Görüntüsü

BT görüntüleme, hastanın vücudundaki belirlenen alanın birden fazla açıdan alınan röntgen görüntülerini birleştirilerek oluşturulur. COVID-19 teşhisinde kullanılan PCR test sonucu, özellikle hastalığın erken evresinde veya viral yük az olduğunda negatif olabilir. Bu evrede hastadan BT görüntülerinin alınması COVID-19 tanısında önemli rol oynamaktadır

(Kang *et al.* 2020). Fang *et al.* (2020) çalışmalarında erken dönemde COVID-19 tespitinde PCR testinin duyarlılığını %71, BT görüntüleri alınarak COVID-19 tespitinin duyarlılığını %98 olarak saptanmıştır. Amerika Radyoloji Koleji, Mart 2020'den itibaren COVID-19 tanısında BT görüntülerinin alınmasını ilk basamak testi olarak önermektedir (Anonymous 2021e).

COVID-19 tanısında X-ray görüntülemeye kıyasla BT görüntüleme daha çok tercih edilmektedir. Çünkü COVID-19 hastalığının erken evrelerinde akciğerde buzlu cam görüntüsü X-ray görüntülerinde görülmeyebilir. X-ray görüntülemenin duyarlılığı, BT görüntülemeye göre düşüktür (Islam *et al.* 2021).

Literatür Çalışması

Derin öğrenme, tanı amaçlı tıp alanında uygulanan son tekniklerden biridir (Shukla *et al.* 2020). Derin öğrenme temelli destek sistemlerinde COVID-19 teşhisi için hem BT hemde X-ray görüntüleri kullanılmaktadır. Bu çalışmalarda kullanılan derin öğrenme modelleri ve sınıflandırma yöntemleri incelenmiştir.

Panwar *et al.* (2020a) yaptıkları çalışmada COVID-19 hastalarının teşhisi için bir derin transfer öğrenme modeli önerilmiştir. Önerilen derin öğrenme modelinde COVID-19 pozitif vakaları tespiti için kullanılan RT-PCR testlerinden daha hızlı tespit edebildiği gösterilmiştir. Yapılan çalışmada 3 çeşit veri seti kullanılmıştır. İlk veri seti 526 radyolojik görüntüden oluşmaktadır. İkinci halka açık veri seti COVID-19 pozitif hastaların 1252 BT görüntüsü ve COVID19 negatif olan ancak başka akciğer hastalıkları olabilen vakalar için 1230 BT görüntüsü içermektedir. Üçüncü veri setinden pnömoni hastalarının ve sağlıklı bireylerin X-ray görüntülerinden oluşan toplam 5856 görüntü kullanılmıştır. Önerilen modelin %94,04 duyarlılık değerine ve %95,86 özgüllük değerine ulaştığı tespit edilmiştir..

Ezzat *et al.* (2020) yaptıkları çalışmada göğüs röntgeni görüntüleri ile COVID-19'u teşhis etmede yüksek düzeyde doğruluk elde etmek için bir optimizasyon algoritması kullanılmıştır. Hibrit ESA mimarisine dayalı GSA-DenseNet121-COVID-19 adlı yeni bir yaklaşım önerilmiştir. Yapılan çalışmada ESA mimarisine DenseNet121 adı verilmiştir. DenseNet121 mimarisinin hiper parametrelerinde en iyi değerleri belirlemek için kullanılan optimizasyon algoritmasına yer çekimsel arama algoritması (GSA) adı verilmiştir. GSA, SSD-DenseNet121 adlı yaklaşımla karşılaştırılmıştır. Elde edilen sonuçlara göre önerilen yaklaşımın doğruluk oranını %98,38 olarak tespit edilmiştir. Veri seti, iki farklı veri setinin birleştirilmesinden elde edilmiştir. Kullanılan veri seti, toplam 306 tane X-ray ve BT görüntüsünden oluşmaktadır. Önerilen yaklaşım ölçeklenebilir, yani DenseNet121'in hiper

parametrelerinin deęerlerini manuel olarak belirtmek zorunda kalmadan eęitim, test ve doęrulama örneklerinin sayısı artırılabilidięinden sınıflandırmanın genişletebilir olduęu gösterilmiştir.

Minaee *et al.* (2020) çalışmalarında COVID-19 vakalarının teşhisi için ResNet18, ResNet50, SqueezeNet ve DenseNet-121 gibi derin öğrenme modelleri incelenmiştir. İlk olarak, halka açık veri setlerinden 5000 akcięer röntgeni içeren veri seti hazırlanmıştır. 71 adet COVID-19 hastalığı varlığını gösteren görüntüler, kurul onaylı radyolog tarafından belirlenmiştir. Bu modeller kalan 3000 görüntü ile test edilmiştir. Bu aęların çoęu yaklaşık %90 özgüllük oranına %98 duyarlılık oranına ulaştığı tespit edilmiştir. COVID-19 ile potansiyel olarak enfekte olmuş akcięer bölgelerinin ısı haritalarını oluşturmak için bir teknik uygulanmıştır.

Ouchichaa *et al.* (2020) yaptıkları çalışmada COVID-19 tespiti için artık sinir aęı tabanlı CVDNet modeli önerilmiştir. Önerilen model, farklı çekirdek boyutlarında olan iki paralel seviye kullanılarak oluşturulmuştur. Veri seti olarak 219 COVID-19, 1345 pnömoni ve 1341 normal akcięer röntgeni görüntülerini içeren halka açık bir veri seti kullanılmıştır. Bu veri setinin %20'si test verisi olarak ayrılmıştır. İkili sınıflandırmada %97,2 doęruluk oranı, üçlü sınıflandırmada %96,69 doęruluk oranı elde edilmiştir.

Singh *et al.* (2020) yaptıkları çalışmada COVID-19 hastalarını tespit etmek için derin öğrenme teknięi ESA kullanılmıştır. Bu modelde, çok amaçlı diferansiyel evrim (MODE) ile ESA'nın başlangıç parametreleri ayarlanmıştır. Veri seti, 75 COVID-19 pozitif ve 75 COVID-19 negatif BT görüntüsünden oluşmaktadır. Veri setini eęitim ve test için ayırırken farklı varyasyonları kullanılarak deneyler yapılmıştır. Önerilen modelde en iyi varyasyon eęitim için %90 ve test için %10 oranı olduęu saptanmıştır. Önerilen model, %90,72 özgüllük, %90,7 duyarlılık, %93,25 doęruluk ve %89,96 F1 skor elde etmiştir.

Uçar and Korkmaz (2020) bu çalışmada X-ray görüntüleri kullanarak COVID-19 teşhisi için Bayes-SqueezeNet temelli bir COVIDdiagnosis-Net önerilmiştir. Hafif ve daha verimli olduęu için SqueezeNet derin öğrenme modeli kullanılmıştır. Kullanılan veri seti 45 COVID-19 vakası, 1203 normal ve COVID-19 olmayan 1591 pnömoni vakası görüntülerinden oluşmuştur. Veri seti, halka açık üç veri setinin birleşiminden oluşmaktadır. Elde edilen sonuçlara göre önerilen model %98.26 doęruluk oranı, %98.25 F1-skor deęeri tespit edilmiştir.

Pandit *et al.* (2020) çalışmalarında bir vakanın COVID-19 ile enfekte olup olmadığını belirlemek için önceden eęitilmiş VGG-16 derin öğrenme modeli incelenmiştir. Bu çalışmadaki veri seti; doęrulanmış COVID-19 pozitif (224), pnömoni (700) ve sağlıklı

vakalar (504) ile 1428 göğüs radyografisini içermektedir. Derin öğrenme modelinin nispeten küçük akciğer radyografileri üzerinde etkili bir şekilde eğitmek için ince ayarlı transfer öğrenimi kullanılmıştır. İkili sınıflandırmada %96 başarı oranı elde ederken, üçlü sınıflandırmada ise %92,5 başarı oranı elde etmiştir.

Wu *et al.*(2020) yapılan çalışmada COVID-19 tanısına ulaşmak için ResNet50 derin öğrenme tekniğine dayalı COVNet adlı bir model önerilmiştir. BT görüntülerinden oluşan veri seti, 1296 adet COVID-19, 1735 pnömoni ve 1325 normal olmak üzere 4336 görüntü içermektedir. Veri setinin %90'ı eğitim için ve %10'u test için ayrılmıştır. Deneysel sonuçlarına göre, önerilen model %90 duyarlılık, %96 özgüllük ve %96 AUC değeri elde etmiştir.

Panwar *et al.*(2020b) çalışmalarında X-ray görüntüleri kullanarak COVID-19 hastalarını tespit etmek için kullanılabilecek alternatif bir hızlı tarama yöntemi olan derin öğrenme sinir ağı tabanlı nCOVnet önermiştir. Veri setinde 142 X-ray görüntüsü COVID-19 pozitif olarak ve Kaggle veri setinden elde edilen 142 X-ray görüntüsü COVID-19 negatif olarak seçilmiştir. Bu veri setinden %30'u test verisi olarak kullanılmıştır. Önerilen model, COVID-19 pozitif hastaları %97 doğruluk oranı ile tespit ederken genel doğruluk oranı %88 olarak kaydedilmiştir.

Ozturk *et al.* (2020) çalışmalarında, COVID-19 vakalarının teşhisi için bir model sunulmuştur. Önerilen model, ikili sınıflandırmada COVID ve Bulgu Yok sınıfları olarak; çok sınıflı sınıflandırmada ise COVID, Bulgu Yok ve Pnömoni sınıfları olarak belirlenmiş ve doğru teşhis sağlamak üzere geliştirilmiştir. Bu model, her katmanında farklı filtreleme yapılan 17 evrişim katmanından oluşmaktadır. İkili sınıflandırmada %98,08 doğruluk değerine ve çok sınıflı durumlarda %87,02 doğruluk değerine ulaşılmıştır.

Rahimzadeh and Attar (2020) çalışmalarında, X-ray görüntülerini üç sınıfa ayırmak için çeşitli derin evrişimli ağlar eğitilmiştir. Veri seti, 180 adet COVID-19 pozitif X-ray görüntüsü içermektedir. Önerilen model, Xception ve ResNet50V2 modellerinin birleşiminden oluşmaktadır. 11302 görüntü, test seti olarak kullanılmıştır. COVID-19 vakalarını tespit etmek için önerilen ağın ortalama doğruluk oranı %99,5 ve tüm sınıflar için genel ortalama doğruluk oranı %91,4 olarak tespit edilmiştir.

Abbas *et al.* (2020) çalışmalarında akciğer röntgeni görüntülerini kullanarak COVID-19 tespiti için ESA modeline dayalı yeni bir derin öğrenme modeli olan DeTraC'ı önermiştir. Veri seti, 105 adet COVID-19 görüntüsü, 11 SARS virüs hastalığı görüntüsü ve 80 normal akciğer röntgeni görüntüsü içermektedir. Yaptıkları deneyler sonucu, önerilen model ile %95,12 doğruluk oranı elde edilmiştir.

Oh *et al.* (2020) yaptıkları çalışmada az sayıda eğitilebilir parametrelere sahip yama tabanlı bir ESA önermiştir. Veri seti, 180 adet COVID-19, 6012 adet pnömoni ve 8851 adet normal vakaların görüntüsünü içermektedir. Önerilen model, yapılan deneyde %88,9 doğruluk oranı tespit edilmiştir.

Zheng *et al.* (2020) çalışmada COVID-19 tespiti için 3D derin ESA tipli bir model (DeCoVNet) önerilmiştir. Kullanılan veri seti hastane ortamından toplanan 630 BT görüntülerinden oluşmaktadır. Veri setinin %80'i eğitim için %20'si test için ayrılmıştır. Yapılan deneylerde, %90,1 doğruluk, %90,7 duyarlılık, %91,1 özgüllük, %95,9 AUC değeri elde edilmiştir.

Apostolopoulos and Mpesiana (2020) yaptıkları çalışmada COVID-19 tespiti için önceden eğitilmiş birkaç ESA modeli incelemiştir. Önerilen model, iki veri seti üzerinde gerçekleştirilmiştir. İlk veri seti, 224 adet COVID-19 X-ray görüntüsü, 700 adet bakteriyel pnömoni X-ray görüntüsü ve 504 adet normal X-ray görüntüsü içermektedir. İkinci veri seti ise 224 adet COVID-19 X-ray görüntüsü, 714 adet bakteriyel ve viral pnömoni görüntüsü ve 504 adet normal görüntü içermektedir. Önceden eğitilmiş modeller arasında en iyi performansı gösteren MobileNetV2 ile ikinci veri kümesi üzerinde %96,78 doğruluk oranı elde edilmiştir.

Sethy and Behara (2020) çalışmalarında COVID-19 vakalarının tespitinde en iyi sınıflandırma modelini seçmek için istatistiksel analiz uygulanmıştır. Sınıflandırma modeli, ResNet-50 diğer sekiz modele (AlexNet, DenseNet201, GoogleNet, Inceptionv3, ResNet18, ResNet50, ResNet101, VGG16, VGG19, XceptionNet, Inceptionresnetv2) kıyaslandığında istatistiksel olarak daha üstün olduğu görülmüştür. COVID-19 tespiti için önerilen sınıflandırma modeli %95,38 doğruluk oranı tespit edilmiştir.

Mahmud *et al.* (2020) yaptıkları çalışmada COVID-19 ve diğer pnömoni vakalarının tespiti için akciğer röntgeni görüntülerini veri seti olarak kullanılmıştır. CovXNet adlı derin öğrenme modeli önerilmiştir. CovXNet'de, akciğer röntgeni görüntülerinden yüksek seviyeli özellikler elde etmek için derinlemesine evrişim uygulanmıştır. Önerilen modele yığınlama algoritması ve gradyan tabanlı ayırt edici yerelleştirme uygulanmıştır. Veri seti, 305 adet COVID-19, 2780 adet bakteriyel pnömoni, 1493 adet viral pnömoni ve 1583 adet normal görüntüden oluşmaktadır. COVID-19 görüntülerinin tespiti için %97,4 doğruluk oranı ve genel sınıflandırma için %90,2 doğruluk oranı saptanmıştır.

Tsiknakis *et al.* (2020) çalışmalarında COVID-19 teşhisi için Inception-V3 önermiştir. COVID-19 veri kümesini sınırlı sayıda veri içermesi nedeniyle transfer öğrenimi ile birleştirilmiştir. Veri seti, 122 adet COVID-19 akciğer röntgeni görüntüsü, 150 adet bakteriyel

pnömoni akciğer röntgeni görüntüsü, 150 adet viral pnömoni akciğer röntgeni görüntüsü ve 150 adet normal akciğer röntgeni görüntüsü içermektedir. İkili sınıflandırma için %100 ve dörtlü sınıflandırma için %76 doğruluk oranı elde edilmiştir.

Horry *et al.* (2020) yaptıkları çalışmada COVID-19 vakalarının tespiti için önceden eğitilmiş dört farklı derin öğrenme modelinin kullanıldığı bir şema sunmuştur. Veri seti; 115 adet COVID-19 vakası, 322 adet pnömoni vakası ve 60.361 adet normal vaka örneği içermektedir. Modeller arasında VGG16 ve VGG19 ile yüksek performans elde edilmiştir. VGG19 kullanarak %81 doğruluk oranı elde edilmiştir.

Toğaçar *et al.* (2020) yaptıkları çalışmada COVID 19 tespiti için MobileNetV2 ve SqueezeNet derin öğrenme modelleri ile derin özellikler çıkarmıştır. Ardından bu özellikler Social Mimic optimization yönteminde işlenmiştir. Daha sonra DVM yönteminde kullanılarak veriler sınıflandırılmıştır. Geliştirilen modelde COVID-19, pnömoni ve normal X-ray görüntüleri kullanılmıştır. Doğruluk oranı %99,27 olarak tespit edilmiştir.

Hemdan *et al.* (2020) bu çalışmada COVID-19 teşhisi için VGG19 ve Google MobileNet'in ikinci sürümü gibi yedi farklı derin ESA modeli içeren bir COVIDX-Net modeli önermiştir. Önerilen model, 25 adet doğrulanmış COVID-19 ve 50 adet normal akciğer röntgeni görüntüsü üzerinde test edilmiştir.

Narin *et al.* (2020) çalışmalarında COVID-19 ile enfekte olan bireylerin tespiti için ResNet50, ResNet101, ResNet152, InceptionV3 ve Inception-ResNetV2 gibi derin öğrenme modelleri incelenmiştir. Akciğer röntgeni görüntüleri içeren 3 farklı veri seti kullanılmıştır. 5 kat çapraz doğrulama uygulanmıştır. Elde edilen deney sonuçlarına göre önceden eğitilmiş ResNet50 modeli kullanılan diğer dört modelden daha yüksek sınıflandırma performansı göstermiştir. İlk veri setinde %96,1 doğruluk değeri, ikinci veri setinde %99,5 doğruluk değeri ve üçüncü veri setinde %99,7 doğruluk değeri saptanmıştır.

Karakanis and Leontidis (2021) çalışmalarında Koşullu Üretken Karşıt Ağlar (cGAN'lar) modelini kullanarak sentetik tıbbi görüntüler oluşturan bir yöntem sunmuştur. COVID-19'u tespit etmek için iki derin öğrenme modeli uygulanmıştır. İlk normal vakalara karşı COVID-19'u, ikincisi üç farklı vakayı; yani bakteriyel pnömoni, COVID-19 ve normal vakaları tespit etmek için transfer öğrenimi olmadan hafif bir mimariyi destekleyecek şekilde tasarlanmış bir ESA modeli önermiştir. Veri seti olarak sentetik görüntüler olmadan; 145 adet COVID-19, 145 adet bakteriyel pnömoni ve 145 adet normal vaka görüntüleri kullanılmıştır. Sentetik görüntülerle birlikte; 275 (130'u sentetik) COVID-19, 275 bakteriyel pnömoni (yalnızca gerçek görüntüler) ve 270 normal (yalnızca gerçek görüntüler) görüntüler

kullanılmıştır. İki sınıflı sınıflandırmada %98,7 doğruluk oranı, üç sınıflı sınıflandırmada ise %98,3 doğruluk oranı elde edilmiştir.

Saha *et al.* (2021) yaptıkları çalışmada akciğer röntgeni görüntüleri üzerinden COVID-19 teşhisi için ESA tabanlı EMCNet modeli önermiştir. Çıkarılan özellikler rastgele orman, destek vektör makinesi, karar ağacı ve AdaBoost gibi sınıflandırma yöntemlerinde kullanılmıştır. Diğer derin öğrenme modelleriyle kıyaslandığında; EMCNet %98,91 doğruluk oranı ile en iyi performansı gösterdiği bildirilmiştir. Veri seti toplam 4600 görüntüden oluşmaktadır. 3220 görüntü eğitim setinde, 920 görüntü doğrulama setinde ve 460 görüntü test setinde kullanılmıştır.

Gupta *et al.* (2021) yaptıkları çalışmada COVID-19 ile enfekte olan hastaları tespit etmek için entegre yığınlı derin evrişim ağı olan InstaCovNet-19 önermiştir. Önerilen modelde, ResNet101, Xception, InceptionV3, MobileNet ve NASNet gibi derin öğrenme modelleri kullanılmıştır. Önerilen model, üçlü sınıflandırmada (COVID-19, pnömoni, normal) %99,08 doğruluk elde ederken, ikili sınıflandırmada (COVID, NONCOVID) %99,53 doğruluk elde edilmiştir. Önerilen modelde kullanılan veri seti, halka açık iki veri setindeki görüntülerden oluşmaktadır. Bu veri seti, 361 adet COVID-19, 1345 adet pnömoni ve 1341 adet normal akciğer röntgeni görüntüleri içermektedir. Görüntüleri %20'si test için ayrılmıştır.

Wu *et al.* (2021) yaptıkları çalışmada COVID-AL adlı zayıf denetimli bir derin aktif öğrenme çerçevesi önermiştir. COVID-AL, bir 2D U-Net ile akciğer segmentasyonundan eşzamanlı olarak numune çeşitliliğini ve tahmin kaybını dikkate alan yeni bir hibrit aktif öğrenme stratejisi uygulamaktadır. Özel olarak tasarlanmış bir 3D kalıntı ağıyla, önerilen COVID-AL, COVID-19'u verimli bir şekilde teşhis edebilir olduğu gösterilmiştir. COVID-19 teşhisi için veri seti, Çin Toraks BT Görüntü İnceleme Konsorsiyumu'ndan (CC-CCII) toplanan BT görüntülerinden oluşmaktadır. Bu veri setinde 304 adet COVID-19, 316 adet pnömoni ve 342 adet normal vaka görüntüleri kullanılmıştır. %86,6 doğruluk oranı tespit edilmiştir.

Panahi *et al.* (2021) yaptıkları çalışmada X-ray görüntüleri kullanarak COVID-19'un hızlı bir şekilde algılanması için Hızlı COVID-19 Dedektörü (FCOD) adlı yeni bir teknik önermiştir. Inception mimarisine dayalı FCOD, COVID-19 tespiti için derinlemesine ayrılabilir 17 evrişim katmanı kullanmaktadır. Önerilen modeli değerlendirmek için halka açık veri setinde COVID-19 ve normal vakaları içeren 940 adet X-ray görüntüsü kullanılmıştır. Bu veri setinin %80'i eğitim, %20'si test için ayrılmıştır. COVID-19 tespitinde doğruluk oranı %96 olarak saptanmıştır.

Bozkurt (2021) çalışmasında, akciğer X-Ray görüntüleri kullanarak COVID-19, normal ve viral pnömoni olarak sınıflandırabilen 11 farklı derin öğrenme tekniklerinin bir karşılaştırması yapılmıştır. Kullanılan veri seti, Kaggle'nin COVID-19 radyografi veritabanından elde edilmiştir. Bu veri seti 1200 adet COVID-19, 1345 adet viral pnömoni ve 1341 adet normal görüntüden oluşmaktadır. Çalışmada kullanılan veri setinin %70 eğitim, %10 doğrulama ve %20 test için ayrılmıştır. Yapılan incelemeye göre, en yüksek doğruluk değeri %97,17 ile DenseNet121 modeli ile elde edilmiştir.

Ismael and Şengür (2021) yaptıkları bu çalışmada COVID-19 ve normal akciğer röntgeni görüntülerinden derin özellikler çıkarmak için ResNet18, ResNet50, ResNet101, VGG16 ve VGG19 gibi önceden eğitilmiş ESA modeller kullanmıştır. Bu derin özellikler, DVM yönteminin Doğrusal, Kuadratik, Kübik ve Gauss gibi çeşitli çekirdek fonksiyonlar kullanılmıştır. Bu çalışmada uçtan uca eğitimle yeni bir ESA modeli önerilmiştir. COVID-19 teşhisi için kullanılan veri setinde, 80 adet COVID-19 ve 200 adet normal akciğer röntgeni görüntüsü bulunmaktadır. En yüksek doğruluk oranı ResNet-50 modelinden çıkarılan derin özellikleri kullanarak doğrusal çekirdek fonksiyonlu DVM sınıflandırıcısından elde edilmiştir. Doğruluk oranı %94,7 olarak belirlenmiştir.

İncelenen çalışmalar arasında X-ray verilerine dayalı 25 çalışma ve geri kalan 4 çalışma BT verilerine dayalı olarak yapılmıştır. COVID-19 hastalığının tespitinde doktorlar X-ray görüntülemeye kıyasla BT görüntülemeyi daha çok tercih etmektedir. Çünkü COVID-19 hastalığının erken evrelerinde akciğerde buzlu cam görüntüsü X-ray görüntülerinde görülmeyebilir. X-ray görüntülemenin duyarlılığı, BT görüntülemeye göre düşüktür (Islam *et al.* 2021). Bu nedenle bu tez çalışmasında BT görüntüleri kullanılmıştır.

Çalışmaların çoğunluğu birden fazla veri seti kullanmıştır. Kullanılan veri setlerinin çoğu halka açık veri tabanlarından oluşturulmuştur. Literatürdeki bazı çalışmalarda (Panwar *et al.* 2020a; Minae *et al.* 2020; Ouchichaa *et al.* 2020; Rahimzadeh and Attar 2020; Oh *et al.* 2020; Mahmud *et al.* 2020; Ucar and Korkmaz *et al.* 2020; Gupta *et al.* 2021) veri seti çok sayıda görüntü içermekte fakat COVID-19 vakalarına ait görüntü sayısı azdır. Ayrıca bazı çalışmalar (Ezzat *et al.* 2020; Singh *et al.* 2020; Panwar *et al.* 2020b; Ozturk *et al.* 2020; Abbas *et al.* 2020; Sethy and Behara *et al.* 2020; Tsiknakis *et al.* 2020; Toğaçar *et al.* 2020; Hemdan *et al.* 2020; Narin *et al.* 2020; Ismael and Sengur *et al.* 2021) az miktarda görüntü içeren veri setine sahiptir. Her iki durumda da sınırlı sayıda veri eksik veya fazla uyum sorununa neden olarak derin öğrenme modellerinin performans düşürür (Islam *et al.* 2021; Nayak *et al.* 2021). Literatürdeki çalışmalarının bazılarında da (Panwar *et al.* 2020a; Minae *et al.* 2020; Ouchichaa *et al.* 2020; Rahimzadeh and Attar 2020; Oh *et al.* 2020; Mahmud *et al.*

2020; Ucar and Korkmaz *et al.* 2020; Abbas *et al.* 2020; Apostolopoulos and Mpesiana 2020; Horry *et al.* 2020; Toğaçar *et al.* 2020; Ismael and Sengur *et al.* 2021; Gupta *et al.* 2021) veri dengesizliği görülmektedir. Veri dengesizliği de bir sınıfa yanlılığı artırır (Islam *et al.* 2021). Genellikle literatürdeki çalışmalarda ikili sınıflandırma uygulanmıştır. İncelenen çalışmalar arasında en iyi performansı Gupta *et al.* (2021) çalışmalarında göstermiştir. Ancak veri seti hem dengesiz hem de veriler hastane ortamından elde edilmemiştir. Açık kaynaklı veri tabanından veri seti oluşturulmuştur. Literatürdeki çalışmaların çoğusunda kullanılan yöntemlerin çalışma zamanlarından bahsedilmemiştir.



MATERYAL VE YÖNTEM

Materyal

Bu tez çalışmasında MATLAB programlama dili kullanılmıştır. MATLAB büyük verilerle hızlı çalışması ve derin öğrenme için hazır fonksiyonlar sağlaması nedeniyle derin öğrenme uygulamalarında popüler olarak kullanılmaktadır. MATLAB’da derin öğrenme uygulamaları için Deep Learning Toolbox kullanılmaktadır.

Deep Learning Toolbox önceden eğitilmiş modeller ve uygulamalarla derin öğrenme modellerini tasarlamak ve gerçekleştirmek için bir ortam sağlamaktadır (Benco *et al.* 2020). Görüntü ve metin verileri üzerinde sınıflandırma ve regresyon gerçekleştirmek için Deep Learning Toolbox’da bulunan ESA (ConvNets) ve Uzun Kısa Süreli Bellek (Long Short-Term Memory-LSTM) ağları kullanılmaktadır. Ayrıca özel eğitim döngüleri, otomatik farklılaştırma ve paylaşılan ağırlıkları kullanarak Üretici Rakip Ağlar (Generative Adversarial Network-GAN) ve siyam ağları gibi ağ mimarileri oluşturulmaktadır. Deep Network Designer uygulaması; ağları grafiksel olarak tasarlanmasına, analiz edilmesine ve eğitilmesine olanak sağlamaktadır. Deney Yöneticisi uygulaması (Experiment Manager app); çoklu derin öğrenme deneyini yönetmeye, eğitim parametrelerini takip etmeye, sonuçları analiz etmeye ve farklı deneylerdeki kodları karşılaştırmaya yardımcı olmaktadır (Anonymous 2021f). Ayrıca katman etkinleştirmelerini görselleştirmekte ve eğitim ilerlemesini grafik olarak göstermeye olanak sunmaktadır. Toolbox; DarkNet-53, ResNet-50, AlexNet, NASNet, SqueezeNet gibi birçok önceden eğitilmiş modeller ile birlikte transfer öğrenmeyi desteklemektedir.

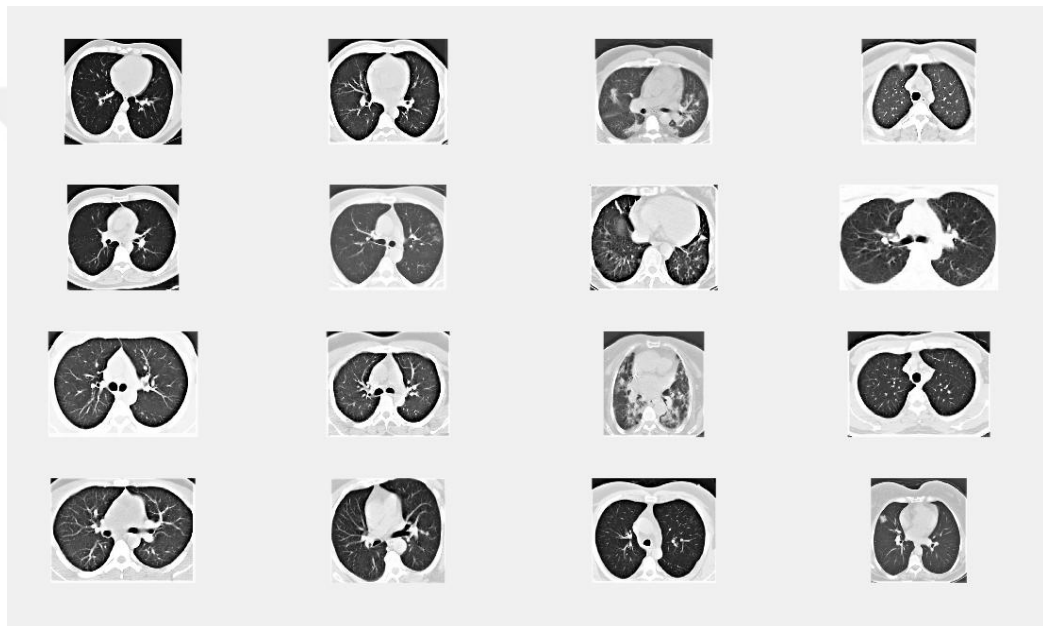
Veri seti

Bu çalışmada kullanılan veri seti için Atatürk Üniversitesi Girişimsel Olmayan Klinik Araştırmalar Etik Kurulu onayı (*bk.* Ek-1) alınmıştır. Bu onay ile Siirt Devlet Hastanesi COVID-19 Hastalıkları polikliniklerine Mayıs 2020 - Haziran 2020 arasında başvuran hastalardan COVID-19 saptanan ve COVID-19 saptanmayan toraks BT verileri, otomasyon sistemi üzerinden alınarak veri seti oluşturulmuştur. Toplam 1345 tane BT görüntüsü kullanarak veri seti oluşturulmuştur. Bunların 738 tanesinde COVID-19 negatif iken; 607 tanesinde ise COVID-19 pozitif idi. Şekil 7’de çalışmada kullanılan Toraks BT aksiyal kesit örnek görüntüleri gösterilmektedir. Şekil 7’de görüldüğü gibi COVID-19 tipik BT bulguları buzlu cam opasiteleridir. Bunun dışında kaldırım taşı, ters halo işareti, konsolidasyon ve

lineer opasitelerdir. Çalışmada toplam 1345 tane BT görüntüsünden 270'i (%20) si test için kullanılmıştır. Veri setinin ayrıntıları Tablo 1 'de verilmektedir.

Tablo 1. Veri Setindeki Eğitim ve Test Setine Ayrılan Görüntü Sayısı

	COVID-19 pozitif	COVID-19 negatif
Eğitim seti	485	590
Test seti	122	148
Toplam	607	738



Şekil 7. Veri setinden örnek görüntüler.

Yöntem

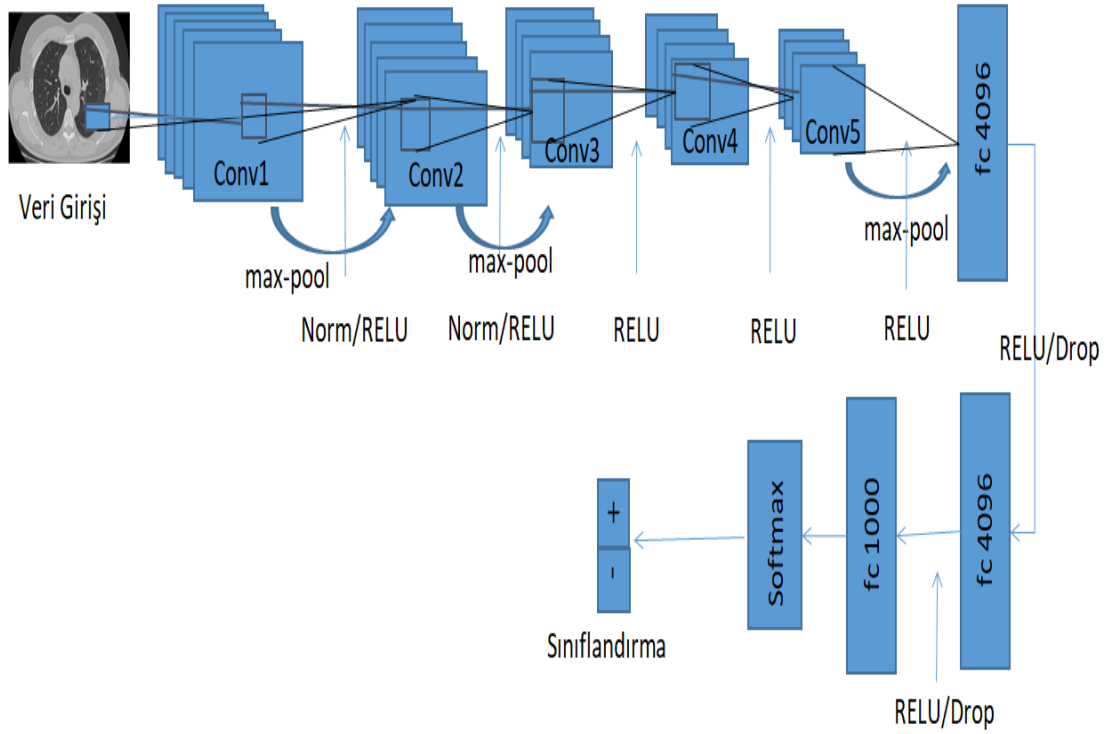
Derin öğrenme modelleri

Alexnet

2012 yılında düzenlenen ImageNet Büyük Ölçekli Görsel Tanıma (ILSVRC) yarışmasında, Alex Krizhevsky (2012) tarafından geliştirilen AlexNet modelinin yarışmasında başarı göstermesi sonrasında derin öğrenmeye olan ilgi artmıştır. Model, ImageNet'te %83,6 doğruluk oranı elde etmiştir.

AlexNet, paralel bir çift GPU üzerinde çalışan ilk modeldir. Böylelikle model daha hızlı eğitilmekte ve hafıza problemleri önlemektedir. 60 milyon parametre ile 650.000 nöron içeren bir sinir ağıdır.

AlexNet ağ yapısı Şekil 8’de gösterilmektedir. Şekil 8’de görüldüğü gibi beşi evrişim ve üçü tam bağlı olmak üzere toplam sekiz öğrenilmiş katmana sahiptir (Krizhevsky *et al.* 2012). Evrişim katmanının temel amacı özellikleri çıkarmaktır. Maksimum havuzlama, özellik eşleştirme gerçekleştirmektedir (Du *et al.* 2016). AlexNet’in görüntü giriş boyutu 227x227’dir. Üçüncü evrişim katmanının çekirdekleri, ikinci katmandaki tüm çekirdek haritalarına bağlanmaktadır. Tam bağlantılı katmanlardaki nöronlar, önceki katmandaki tüm nöronlara bağlıdır. Normalleştirme, birinci ve ikinci evrişim katmanından sonra uygulanmaktadır. Maksimum havuzlama hem normalizasyondan sonra hem de beşinci evrişim katmanından sonra uygulanmaktadır. Maksimum havuzlama, 3x3 piksel pencereler üzerinde gerçekleştirilmektedir. Maksimum havuzlama işlemi için adım sayısı (stride) [2 2], piksel ekleme(padding) [0 0 0 0] olarak ayarlanmıştır. RELU, aktivasyon fonksiyonu olarak kullanılmaktadır. RELU, her evrişim katmanından sonra ve tam bağlı katmanın çıktısına uygulanmaktadır. Bu, eğitim süresinin kısaltılmayı sağlamaktadır. İlk evrişim katmanı, 11x11x3 boyutunda 96 çekirdekli evrişim filtreler içermektedir. İlk evrişim katmanında filtre için adım sayısı [4 4], piksel ekleme [0 0 0 0] olarak ayarlanmıştır. İkinci evrişim katmanı, girdi olarak ilk evrişim katmanından sonra normalleştirilme ve havuzlama işlemlerinin çıktısını almaktadır. 5x5x48 boyutunda 256 çekirdekli evrişim filtreleri içermektedir. İkinci evrişim katmanında filtre için adım sayısı [1 1], piksel ekleme [2 2 2 2] olarak ayarlanmıştır. Üçüncü, dördüncü ve beşinci evrişimli katmanlar, araya giren herhangi bir havuzlama veya normalleştirme katmanı olmadan birbirine bağlanmaktadır. Üçüncü evrişim katmanı, ikinci evrişim katmanından sonra normalleştirilme havuzlama işlemlerinin çıktısını almaktadır. 3x3x256 boyutunda 384 çekirdekli evrişim filtreleri içermektedir. Üçüncü evrişim katmanında filtre için adım sayısı [1 1], piksel ekleme [1 1 1 1] olarak ayarlanmıştır. Dördüncü evrişim katmanı, 3x3x192 boyutunda 384 çekirdekli evrişim filtrelerine sahiptir. Dördüncü evrişim katmanında filtre için adım sayısı [1 1], piksel ekleme [1 1 1 1] olarak ayarlanmıştır. Beşinci evrişim katmanı, 3x3x192 boyutunda 256 çekirdekli evrişim filtrelerine sahiptir. Beşinci evrişim katmanında filtre için adım sayısı [1 1], piksel ekleme [1 1 1 1] olarak ayarlanmıştır. İlk iki tam bağlı katmanın her biri 4096, son tam bağlı katman 1000 kanal içermektedir. Modelin eğitim verilerindeki örnekleri ezberlemesini önlemek için ilk iki tam bağlı katmanda %50’lik dropout (drop değeri 0,5) uygulanmaktadır (Krizhevsky *et al.* 2012).

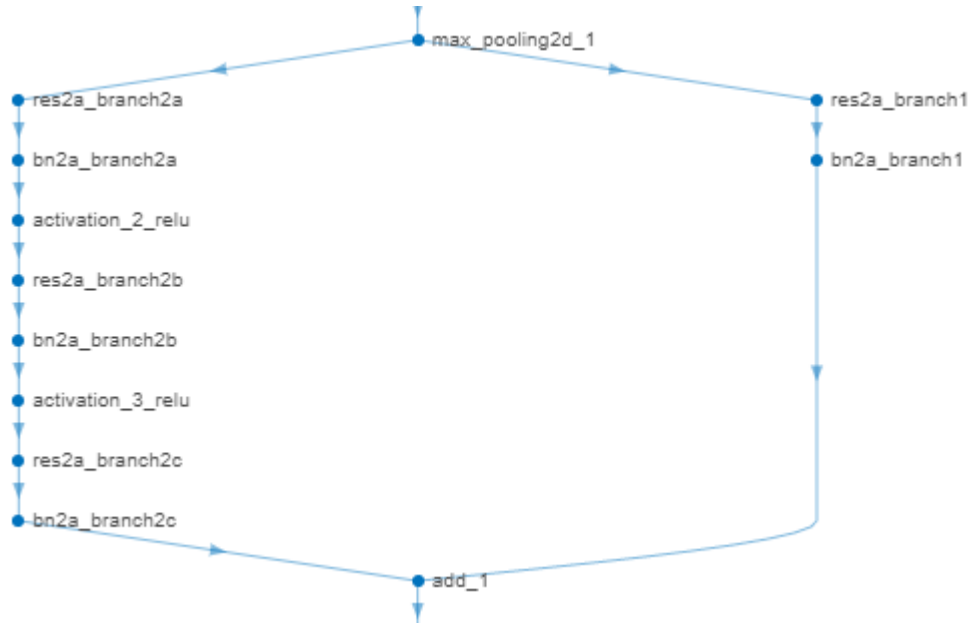


Şekil 8. AlexNet ağ yapısı.

ResNet-50 ve ResNet-101

ResNet, He *et al.* (2016) tarafından ILSVRC 2015 sınıflandırma yarışmasında ilk kez ortaya atılmıştır ve yarışmada birinci olmuştur. ImageNet test setinde %3,57 hata oranına sahiptir. Evrişimli sinir ağının geliştirilmiş versiyonudur.

ResNet modeli ESA'larının performans düşümü problemini çözmeyi hedeflemiştir. Ağ derinliği arttıkça bir bozulma sorunu ortaya çıkmaktadır: doğruluk doygun hale gelmekte ve ardından hızla düşmektedir. Bu tür bir bozulma aşırı uyumdan kaynaklanmamaktadır, parametre güncellemeleri sırasında bilgi kaybı oluşmasından kaynaklanmaktadır. ResNet bu problemi çözmek için katmanlar arasına ara katmanları birbirine bağlayan ekstra bağlantılar eklemiştir. Bu ekstra bağlantılar bir veya daha fazla katmanı atlayabilen bağlantılardır ve kısa yol bağlantıları olarak adlandırılır. Şekil 9'da gösterildiği gibi kısa yol bağlantıları kimlik eşleme gerçekleştirir ve bunların çıktıları yığılanmış katmanların çıktılarına eklenir. Bu bağlantılar ağa ne fazladan parametre ne de hesaplama karmaşıklığı eklenmesine neden olur (He *et al.* 2016).



Şekil 9. ResNet kısa yol bağlantıları.

Bu çalışmada ResNet-50 ve ResNet-101 modelleri kullanarak deneyler gerçekleştirilmiştir. Tablo 2’de ResNet katman yapısı gösterilmiştir.

50 katmanlı ResNet: Tablo 2’de görüldüğü gibi 34 katmanlı ağdaki her 2 katmanlı bloğu 3 katmanlı dar boğaz bloğuyla değiştirilerek 50 katmanlı ResNet elde edilmiştir. Bu modelde 3,8 milyar FLOP mevcuttur.

101 Katmanlı ResNet: Tablo 2’de görüldüğü gibi ResNet-50’ye göre daha fazla 3 katmanlı blok kullanarak 101 katmanlı ResNet oluşturulmuştur. Bu modelde 7,6 milyar FLOP mevcuttur.

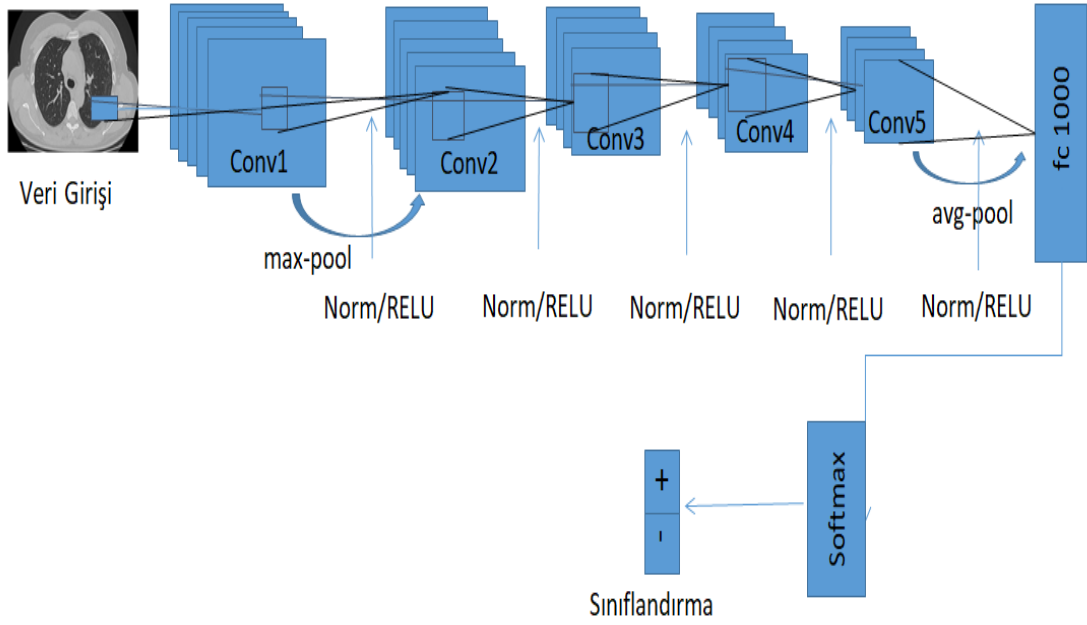
Modelin görüntü giriş boyutu 224x224’dür. Tam bağlantılı katmanlardaki nöronlar, önceki katmandaki tüm nöronlara bağlıdır.

Şekil 10’da ResNet ağ yapısı gösterilmektedir. Şekil 10’da görüldüğü gibi normalleştirme, birinci evrişim katmanı hariç tüm evrişim katmanlarından sonra uygulanmaktadır. Maksimum havuzlama birinci evrişim katmanından sonra uygulanmaktadır. Maksimum havuzlama, 3x3 piksel pencereler üzerinde gerçekleştirilmektedir. Maksimum havuzlama işlemi için adım sayısı (stride) [2 2], piksel ekleme (padding) [1 1 1 1] olarak ayarlanmıştır. 7x7 boyutunda ortalama havuzlama işlemi beşinci evrişim katmanından sonra yapılmaktadır. RELU, aktivasyon fonksiyonu olarak kullanılmaktadır. RELU, her evrişim katmanından sonra uygulanmaktadır. Bu eğitim süresini kısaltılmayı sağlamaktadır. İlk evrişim katmanı, 7x7 boyutunda 64 çekirdekli evrişim filtreler içermektedir. İlk evrişim katmanında filtreler için adım sayısı [2 2], piksel ekleme [3 3 3 3] olarak ayarlanmıştır. İkinci evrişim katmanı, girdi olarak ilk evrişim katmanından sonra RELU ve havuzlama işlemlerinin

çıkışını almaktadır. 1×1 ve 3×3 boyutlarında 64 çekirdekli ve 1×1 boyutunda 256 çekirdekli evrişim filtreleri içermektedir. İkinci evrişim katmanında filtreler için adım sayısı [1 1], piksel ekleme 1×1 boyutunda filtreler için [0 0 0 0] olarak ve 3×3 boyutunda filtreler için ‘same’ olarak ayarlanmıştır. Üçüncü evrişim katmanı, girdi olarak ikinci evrişim katmanından sonra RELU ve normalleştirilme işlemlerinin çıkışını almaktadır. 1×1 ve 3×3 boyutlarında 128 çekirdekli ve 1×1 boyutunda 512 çekirdekli evrişim filtreleri içermektedir. Üçüncü evrişim katmanında filtreler için adım sayısı [1 1], piksel ekleme 1×1 boyutunda filtreler için [0 0 0 0] olarak ve 3×3 boyutunda filtreler için ‘same’ olarak ayarlanmıştır. Dördüncü evrişim katmanı, girdi olarak üçüncü evrişim katmanından sonra RELU ve normalleştirilme işlemlerinin çıkışını almaktadır. 1×1 ve 3×3 boyutlarında 256 çekirdekli ve 1×1 boyutunda 1024 çekirdekli evrişim filtreleri içermektedir. Dördüncü evrişim katmanında filtreler için adım sayısı [1 1], piksel ekleme 1×1 boyutunda filtreler için [0 0 0 0] olarak ve 3×3 boyutunda filtreler için ‘same’ olarak ayarlanmıştır. Beşinci evrişim katmanı, girdi olarak dördüncü evrişim katmanından sonra RELU ve normalleştirilme işlemlerinin çıkışını almaktadır. 1×1 ve 3×3 boyutlarında 512 çekirdekli ve 1×1 boyutunda 2048 çekirdekli evrişim filtreleri içermektedir. Beşinci evrişim katmanında filtreler için adım sayısı [1 1], piksel ekleme 1×1 boyutunda filtreler için [0 0 0 0] olarak ve 3×3 boyutunda filtreler için ‘same’ olarak ayarlanmıştır.

Tablo 2. ResNet Katmanları (He *et al.* 2016).

Katman adı	Çıkış ölçüsü	34 katmanlı	50 katmanlı	101 katmanlı
Conv1	112x112	7x7,64		
3x3 max pool				
Conv2_x	56x56	$\begin{bmatrix} 3 \times 3 & 64 \\ 3 \times 3 & 64 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1 & 64 \\ 3 \times 3 & 64 \\ 1 \times 1 & 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1 & 64 \\ 3 \times 3 & 64 \\ 1 \times 1 & 256 \end{bmatrix} \times 3$
Conv3_x	28x28	$\begin{bmatrix} 3 \times 3 & 128 \\ 3 \times 3 & 128 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1 & 128 \\ 3 \times 3 & 128 \\ 1 \times 1 & 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1 & 128 \\ 3 \times 3 & 128 \\ 1 \times 1 & 512 \end{bmatrix} \times 4$
Conv4_x	14x14	$\begin{bmatrix} 3 \times 3 & 256 \\ 3 \times 3 & 256 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 & 256 \\ 3 \times 3 & 256 \\ 1 \times 1 & 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 & 256 \\ 3 \times 3 & 256 \\ 1 \times 1 & 1024 \end{bmatrix} \times 23$
Conv5_x	7x7	$\begin{bmatrix} 3 \times 3 & 512 \\ 3 \times 3 & 512 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1 & 512 \\ 3 \times 3 & 512 \\ 1 \times 1 & 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1 & 512 \\ 3 \times 3 & 512 \\ 1 \times 1 & 2048 \end{bmatrix} \times 3$
	1x1	Avg pool, fc1000, softmax, sınıflandırma		
FLOPs		3,6x10 ⁹	3,8x10 ⁹	7,6x10 ⁹



Şekil 10. ResNet ağ yapısı.

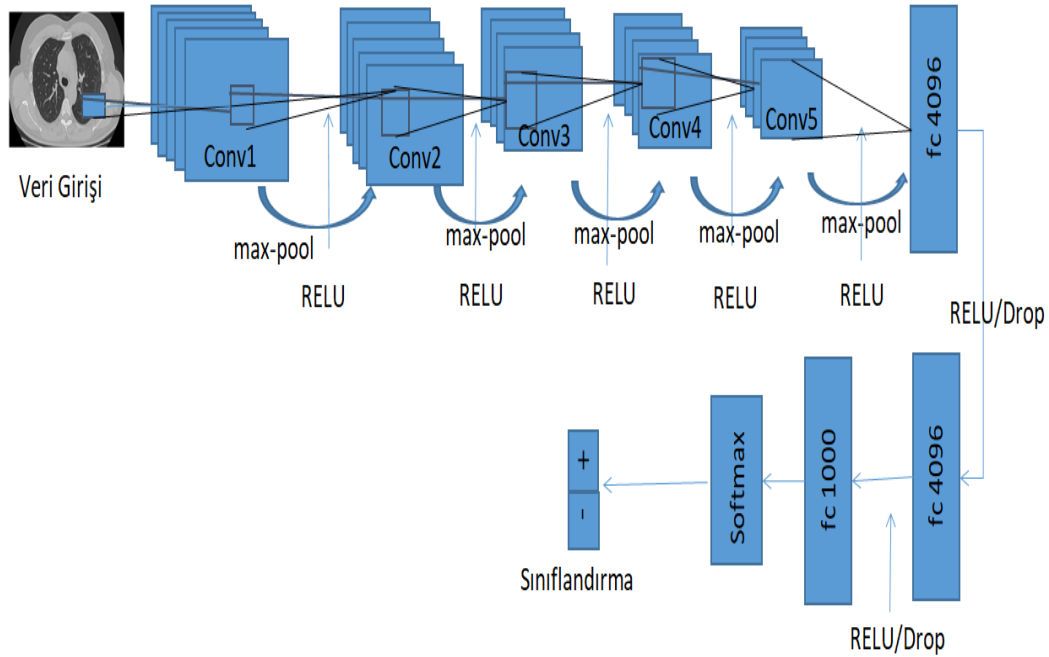
Vgg-16 ve Vgg-19

VGG, ILSVRC 2014 sınıflandırma görevinde Visual Geometry Group'tan (VGG) Simonyan ve Zisserman tarafından ortaya atılmıştır (Zhao *et al.* 2020). Önceden eğitilmiş ESA'nın bir türüdür (Simonyan and Zisserman 2014). VGG, üç set; 1,3 milyon eğitim görüntüsü, 100.000 test görüntüsü ve 50.000 doğrulama görüntüsüne bölünmüş 1000 sınıfın görüntüsünü içeren ImageNet ILSVRC veri kümesi konusunda eğitilmiştir (Gulli and Pal 2017). Model, ImageNet'te %92,7 doğruluk oranı elde etmiştir (Simonyan and Zisserman 2014).

VGG, ESA'ların derinliğini artırarak sınıflandırma doğruluğunu artırmak amacıyla geliştirilmiştir. Nesne tanıma için sırasıyla 16 katmanlı Vgg-16 ve 19 katmanlı Vgg-19 kullanılmıştır. VGG, 224×224 boyutunda görüntü girişini almaktadır. Bunları sabit filtre boyutu 3×3 ve 1 adım olan bir evrişim katman yığınınından geçirir (Zhao *et al.* 2020). Boyut küçültmek için evrişim katmanlar arasına yerleştirilmiş beş maksimum havuz filtresi vardır. Maksimum havuzlama, 2x2 piksel pencereler üzerinde gerçekleştirilmektedir. Maksimum havuzlama işlemi için adım sayısı(stride) [2 2], piksel ekleme (padding) [0 0 0 0] olarak ayarlanmıştır. Dropout katmanında çıktılar %50 (drop değeri 0,5) oranında düşürülmektedir.

Şekil 11'de VGG ağ yapısı gösterilmektedir. Şekilde görüldüğü gibi evrişim katmanlarından sonra sırasıyla 4096, 4096 ve 1000 kanala sahip 3 tam bağlı katmanlar gelmektedir (Zhao *et al.* 2020). Son katman softmax katmanıdır (Simonyan and Zisserman

2014). Evrişim katmanlarında filtreler için adım sayısı [1 1], piksel ekleme [1 1 1 1] olarak ayarlanmıştır. İlk evrişim katmanı, 3×3 boyutunda 64 çekirdekli evrişim filtreleri içermektedir. İkinci evrişim katmanı, girdi olarak ilk evrişim katmanından sonra RELU ve havuzlama işlemlerinin çıktısını almaktadır. 3x3 boyutunda 128 çekirdekli evrişim filtreleri içermektedir. Üçüncü evrişim katmanı, girdi olarak ikinci evrişim katmanından sonra RELU ve havuzlama işlemlerinin çıktısını almaktadır. 3x3 boyutunda 256 çekirdekli evrişim filtreleri içermektedir. Dördüncü evrişim katmanı, girdi olarak üçüncü evrişim katmanından sonra RELU ve havuzlama işlemlerinin çıktısını almaktadır. 3x3 boyutunda 512 çekirdekli evrişim filtreleri içermektedir. Beşinci evrişim katmanı, girdi olarak dördüncü evrişim katmanından sonra RELU ve havuzlama işlemlerinin çıktısını almaktadır. 3x3 boyutunda 512 çekirdekli evrişim filtreleri içermektedir. Beşinci evrişim katmanından sonra son havuzlama işlemi gerçekleştirilmektedir. Çıktısı tam bağlı katmana verilmektedir. Tam bağlı katmanlar arasında RELU ve dropout işlemleri uygulanmaktadır. Çalışmamızda hem Vgg-16 hem de Vgg-19 kullanılmıştır.



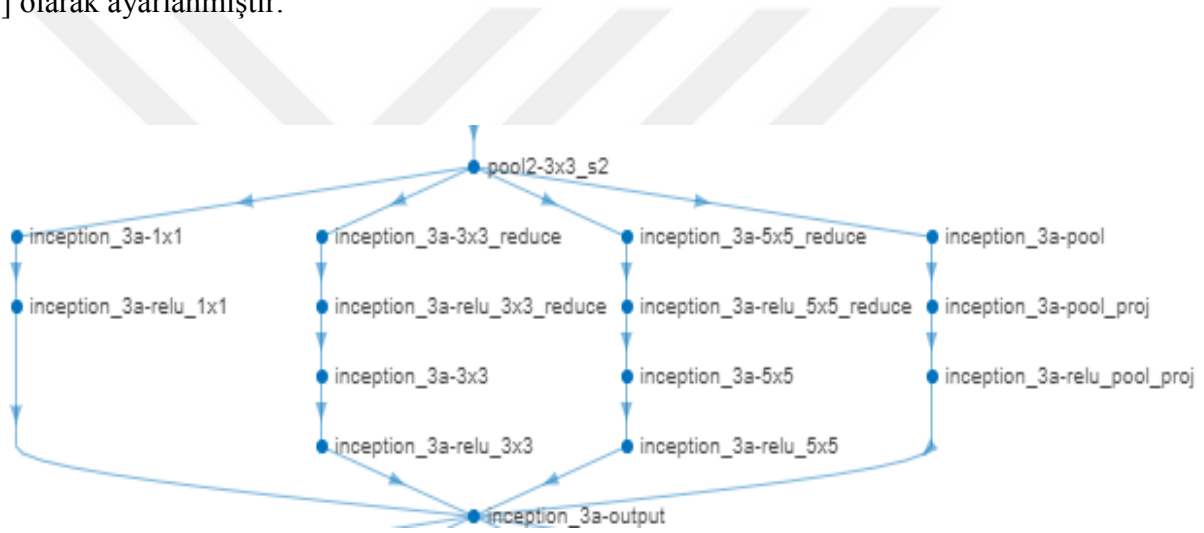
Şekil 11. Vgg ağ yapısı.

GoogleNet

GoogLeNet, kod adı Inception olan ve adını "we need to go deeper" ile bağlantılı olan Lin *et al.* (2013) ağ çalışmasından almıştır. 2014 yılında Szegedy *et al.* (2015) tarafından önerilen bu model ILSVRC yarışmasında birinci olmuştur. %94,3 ile yüksek bir başarı oranı elde etmiştir. Bu ağ, yalnızca parametrelere sahip katmanları sayarken 22 katman derinliğindedir. Ağın inşası için toplam 144 katman kullanılmıştır.

GoogLeNet derin öğrenme modelinden önce ortaya atılan modellerden en belirgin farkı Inception modülüdür. Inception modülünün temel mantığı hesaplama gereksinimlerinin çok fazla artacağı yerlerde boyutu makul bir şekilde küçültmektir. Şekil 12’de Inception modülü gösterilmektedir. Şekilde görüldüğü gibi bu modül içerisinde yer alan 1x1, 3x3, 5x5 gibi farklı boyutlardaki filtre elemanları ile boyut azaltma işlemi gerçekleştirilmektedir. AlexNet mimarisinden 12 kat daha az parametre kullanılmıştır (Szegedy *et al.* 2015).

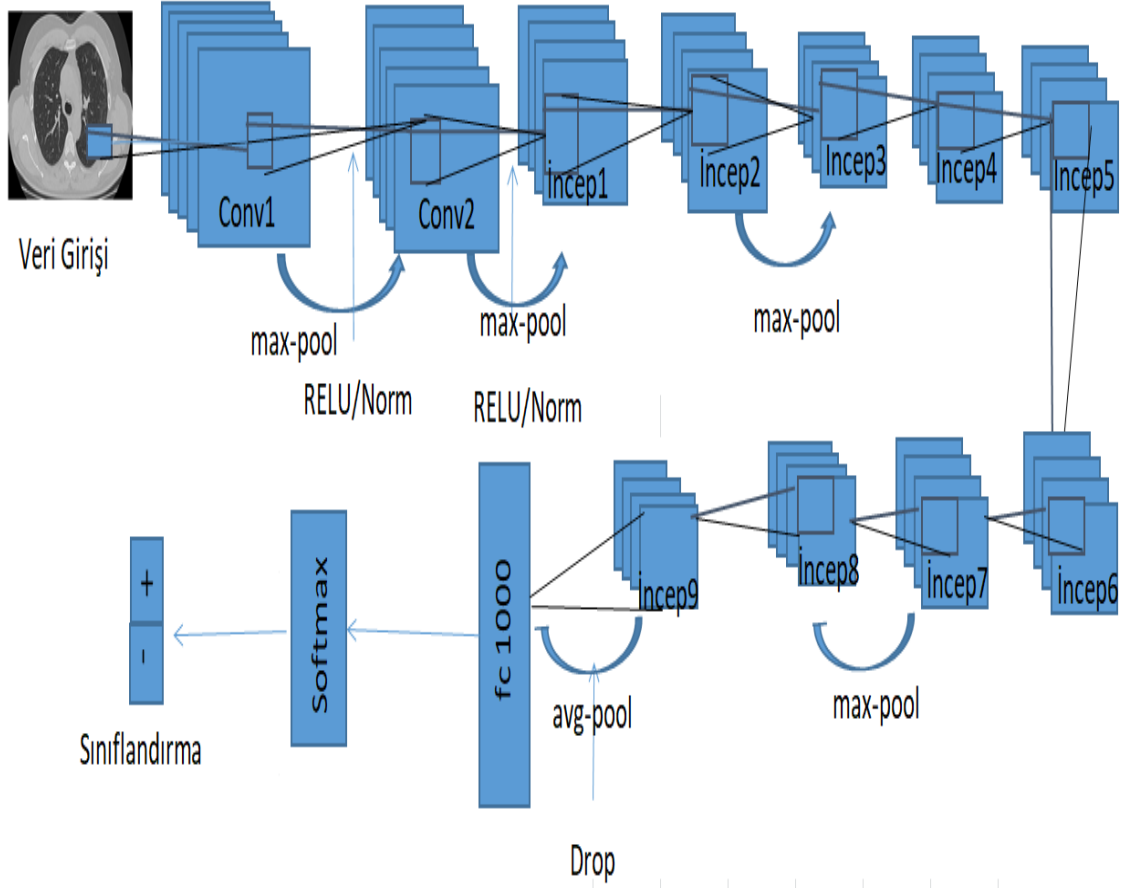
GoogLeNet’in görüntü giriş boyutu 224 x 224’dür. Dropout katmanında çıktılar %40 oranında düşürülmektedir. Her aşamada alternatif bir paralel maksimum havuzlama yolu eklenerek havuzlama işlemi toplam beş havuzlama katmanında gerçekleştirilmektedir (Szegedy *et al.* 2015). Maksimum havuzlama, 3x3 piksel pencereler üzerinde gerçekleştirilmektedir. Maksimum havuzlama işlemleri için adım sayısı (stride) [2 2], piksek ekleme (padding) [0 1 0 1] olarak ayarlanmıştır.



Şekil 12. Inception modülü.

GoogLeNet ağ yapısı Şekil 13’de gösterilmektedir. Şekil 13’de görüldüğü gibi model iki evrişim katmanı içermektedir. İlk evrişim katmanı, 7×7 boyutunda 64 çekirdekli evrişim filtreleri içermektedir. İlk evrişim katmanında filtreler için adım sayısı (stride) [2 2], piksek ekleme (padding) [3 3 3 3] olarak ayarlanmıştır. İkinci evrişim katmanı, girdi olarak ilk evrişim katmanından sonra RELU, maksimum havuzlama ve normalizasyon işlemlerinin çıktısını almaktadır. 1x1 boyutunda 64 çekirdekli ve 3x3 boyutunda 192 çekirdekli evrişim filtreleri içermektedir. İkinci evrişim katmanında 1x1 boyutunda filtreler için adım sayısı (stride) [1 1], piksek ekleme (padding) [0 0 0 0] olarak ve 3x3 boyutunda filtreler için adım sayısı (stride) [1 1], piksek ekleme (padding) [1 1 1 1] olarak ayarlanmıştır. İkinci evrişim katmanından sonra RELU, maksimum havuzlama ve normalizasyon işlemlerinin çıktısını inception modülü girdi olarak almaktadır. İkinci evrişim katmanından sonra 9 tane inception modülü gelmektedir. Inception modüllerinde adım sayısı [1 1] olarak, piksel ekleme 1x1

boyutunda filtreler için $[0\ 0\ 0\ 0]$, 3×3 boyutunda filtreler için $[1\ 1\ 1\ 1]$, 5×5 boyutunda filtreler için $[2\ 2\ 2\ 2]$ olarak ayarlanmıştır. İkinci inception modülünün çıkışına ve yedinci inception modülünün çıkışına maksimum havuzlama işlemi uygulanmaktadır. Tam bağlı katman girdi olarak sonuncu inception modülünden sonra 7×7 boyutunda ortalama havuzlama ve dropout (0,4 drop değeri) işlemlerinin çıkışını almaktadır.



Şekil 13. GoogLeNet ağ yapısı.

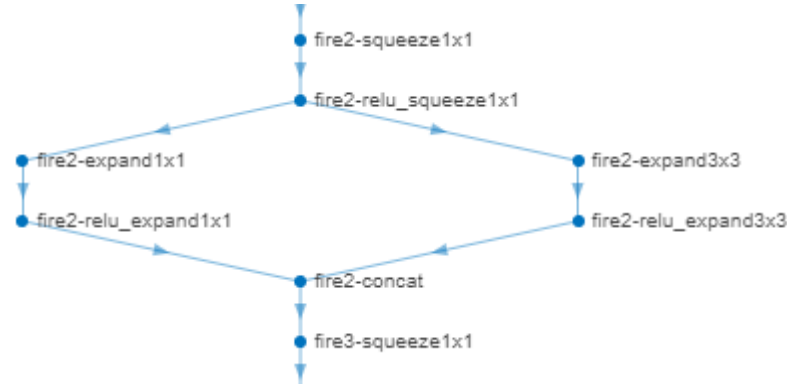
SqueezeNet

SqueezeNet, ImageNet veritabanından bir milyondan fazla görüntü üzerinde eğitilmiş 18 katman derinliğinde evrişimli bir sinir ağıdır (Anonymous 2021g).

Model üç stratejiye dayanmaktadır.

1. 3×3 filtreleri yerine 1×1 filtrelerle değiştirerek 9 kat parametre sayısını azaltmak.
2. Giriş kanallarının sayısını 3×3 filtreye düşürerek parametre sayısını azaltmak.
3. Ağda geç örnekleme yaparak evrişim katmanları, büyük aktivasyon haritalarına sahip olmaktadır. Böylece sınırlı sayıda parametre ile doğruluk oranını artırmak (Iandola *et al.* 2016).

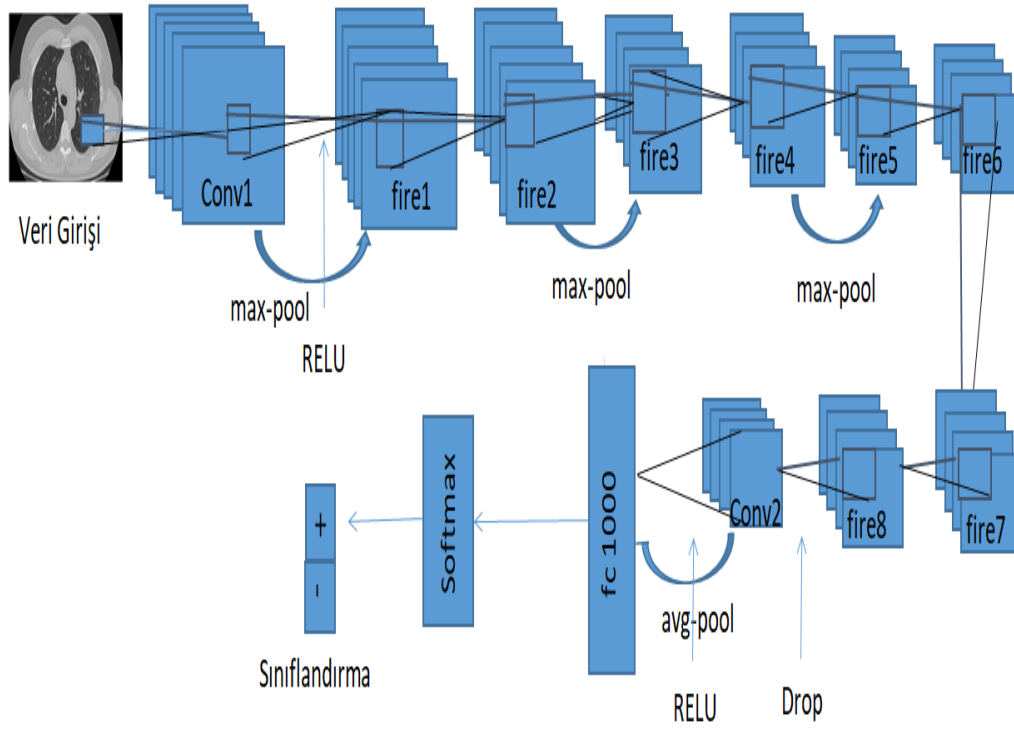
Diğer derin öğrenme modellerinden farklı tarafı yangın modülleridir. Yangın modülünün yapısını Şekil 14’de gösterilmiştir. Şekil 14’de görüldüğü gibi bu modül 1x1 ve 3x3 evrişim filtrelerinin karışımından oluşan bir genişleme katmanından ve yalnızca 1x1 evrişim filtreleri ile sıkıştırılmalı evrişim katmanından oluşmaktadır (Iandola *et al.* 2016).



Şekil 14. Yangın modülü.

Ağ görüntü giriş boyutu 227x227’dir. (Anonymous 2021g). Şekil 15’de SqueezeNet ağ yapısı gösterilmektedir. Şekil 15’de görüldüğü gibi SqueezeNet mimarisi bağımsız evrişim katmanı ile başlamaktadır. Ardından 3x3 boyutunda maksimum havuzlama işlemi gerçekleştirilmektedir. Maksimum havuzlama işlemi için adım sayısı (stride) [2 2], piksel ekleme (padding) [1 1 1 1] olarak ayarlanmıştır. Daha sonra 8 yangın modülü ve evrişim katmanı ile devam etmektedir. Yangın modülleri arasında iki maksimum havuzlama işlemi daha gerçekleştirilmektedir. Yangın modülleri arasındaki maksimum havuzlama işlemi için adım sayısı [2 2], piksel ekleme [0 1 0 1] olarak ayarlanmıştır. Son olarak ortalama havuzlama işlemi ile sonlanmaktadır. Dropout katmanında çıktılar %50 (0,5 drop değeri) oranında düşürülmektedir. Ağın başından sonuna kadar yangın modülü başına filtre sayısını kademeli olarak artırılmaktadır (Askar *et al.* 2021).

İlk evrişim katmanı, 3×3 boyutunda 64 çekirdekli evrişim filtreler içermektedir. İlk evrişim katmanındaki filtreler için adım sayısı [2 2], piksel ekleme [0 0 0 0] olarak ayarlanmıştır. Yangın modüllerinde filtreler için adım sayısı [1 1], piksel ekleme 1x1 boyutunda filtreler için [0 0 0 0] ve 3x3 boyutundaki filtreler için [1 1 1 1] olarak ayarlanmıştır. İkinci evrişim katmanı, girdi olarak 8 yangın modülün sonra dropout işlemi çıktısını almaktadır. 1x1 boyutunda 1000 çekirdekli evrişim filtreleri içermektedir. İkinci evrişim katmanında filtreler için adım sayısı [1 1], piksel ekleme [0 0 0 0] olarak ayarlanmıştır. Ortalama havuzlama katmanı, girdi olarak ikinci evrişim katmanı sonrası RELU işleminin çıktısını almaktadır.



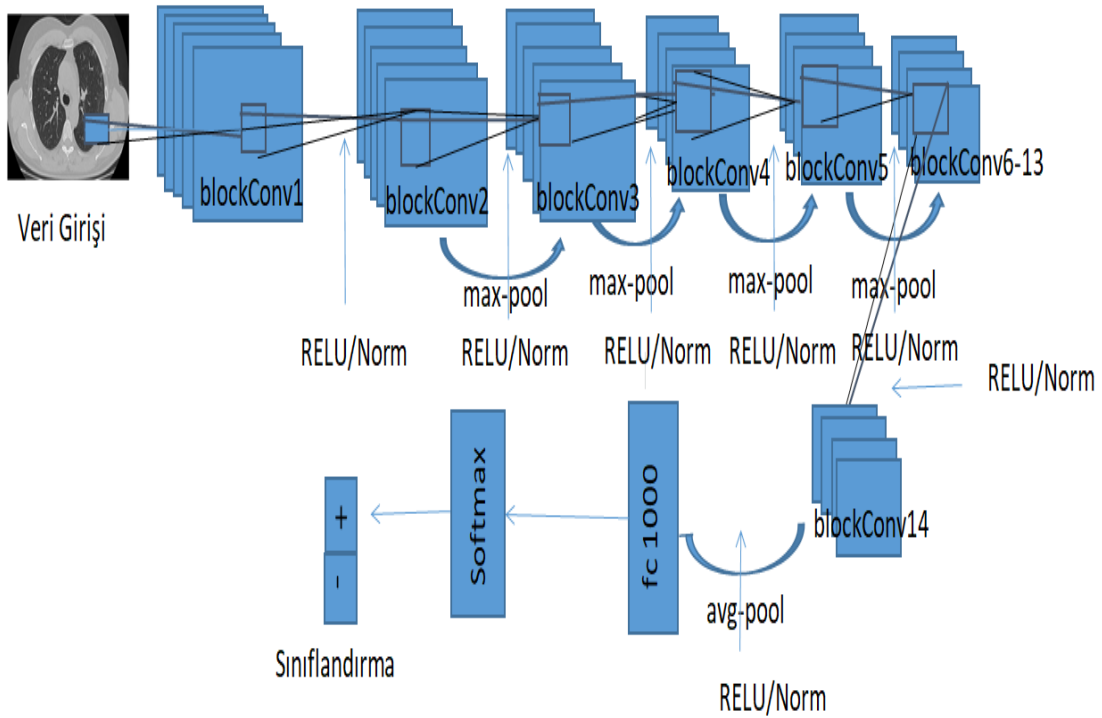
Şekil 15. SqueezeNet ağ yapısı.

Xception

Xception, ImageNet veritabanından bir milyondan fazla görüntü üzerinde eğitilmiş 71 katman derinliğinde evrişimli bir sinir ağıdır (Anonymous 2021h). Xception mimarisi, ağıın özellik çıkarma tabanını oluşturan 36 evrişim katmanı içermektedir. Inception mimarisinin altında yatan düşüncenin daha güçlü bir versiyonu olduğu için, "Extreme Inception" anlamına gelen Xception olarak adlandırılmıştır. Ağıın görüntü giriş boyutu 229 x 229'dur. (Chollet 2017).

Şekil 16'da Xception ağ yapısı gösterilmektedir. Şekil 16'da görüldüğü gibi model, 14 blok evrişim katmanı içermektedir. Maksimum havuzlama işlemi 3x3 piksel pencereler üzerinde gerçekleştirilmektedir. Maksimum havuzlama işlemleri için adım sayısı (stride) [2 2], piksel ekleme (padding) 'same' olarak ayarlanmıştır. İlk blok evrişim katmanı, 3x3 boyutunda 32 çekirdekli ve 3x3 boyutunda 64 çekirdekli evrişim filtreler içermektedir. İlk blok evrişim katmanında 32 çekirdekli filtreler için adım sayısı [2 2], 64 çekirdekli filtreler için adım sayısı [1 1] ve piksel ekleme [0 0 0 0] olarak ayarlanmıştır. İkinci blok evrişim katmanı, girdi olarak ilk evrişim katmanından sonra RELU ve normaleştirme işlemlerinin çıktısını almaktadır. 3x3 boyutunda 64 çekirdekli evrişim filtreleri ve 1x1, 3x3 boyutunda 128 çekirdekli evrişim filtreleri içermektedir. İkinci blok evrişim katmanında 3x3 boyutundaki filtreler için adım sayısı [1 1], piksel ekleme 'same' olarak, 1x1 boyutunda filtrelerde adım sayısı

[1 1], piksel ekleme [0 0 0 0] olarak ayarlanmıştır. Üçüncü blok evrişim katmanı, girdi olarak ikinci blok evrişim katmanından sonra RELU, normalleştirilme ve havuzlama işlemlerinin çıktısını almaktadır. 3x3 boyutunda 128 çekirdekli evrişim filtreleri ve 1x1, 3x3 boyutlarında 256 çekirdekli evrişim filtreleri içermektedir. Üçüncü blok evrişim katmanında 3x3 boyutundaki filtreler için adım sayısı [1 1], piksel ekleme 'same' olarak, 1x1 boyutunda filtrelerde adım sayısı [1 1], piksel ekleme [0 0 0 0] olarak ayarlanmıştır. Dördüncü blok evrişim katmanı, girdi olarak üçüncü blok evrişim katmanından sonra RELU, normalleştirilme ve havuzlama işlemlerinin çıktısını almaktadır. 3x3 boyutunda 256 çekirdekli evrişim filtreleri ve 1x1, 3x3 boyutlarında 728 çekirdekli evrişim filtreleri içermektedir. Dördüncü blok evrişim katmanında 3x3 boyutundaki filtreler için adım sayısı [1 1], piksel ekleme 'same' olarak, 1x1 boyutunda filtrelerde adım sayısı [1 1], piksel ekleme [0 0 0 0] olarak ayarlanmıştır. Beşinci blok evrişim katmanı, girdi olarak dördüncü blok evrişim katmanından sonra RELU, normalleştirilme ve havuzlama işlemlerinin çıktısını almaktadır. 1x1 ve 3x3 boyutlarında 728 çekirdekli evrişim filtreleri içermektedir. Beşinci blok evrişim katmanında 3x3 boyutundaki filtreler için adım sayısı [1 1], piksel ekleme 'same' olarak, 1x1 boyutunda filtrelerde adım sayısı [1 1], piksel ekleme [0 0 0 0] olarak ayarlanmıştır. Altıncı blok evrişim katmanı, girdi olarak beşinci blok evrişim katmanından sonra RELU ve normalleştirilme işlemlerinin çıktısını almaktadır. 1x1 ve 3x3 boyutlarında 728 çekirdekli evrişim filtreleri içermektedir. Altıncı blok evrişim katmanında 3x3 boyutundaki filtreler için adım sayısı [1 1], piksel ekleme 'same' olarak, 1x1 boyutunda filtrelerde adım sayısı [1 1], piksel ekleme [0 0 0 0] olarak ayarlanmıştır. On üçüncü blok evrişim katmanına kadar diğer blok evrişim katmanları aynı işlemler gerçekleşmekte ve aynı boyutta evrişim filtreleri içermektedir. On üçüncü blok evrişim katmanı, girdi olarak on ikinci blok evrişim katmanından sonra RELU ve normalleştirilme işlemlerinin çıktısını almaktadır. 3x3 boyutunda 728 çekirdekli ve 1x1, 3x3 boyutlarında 1024 çekirdekli evrişim filtreleri içermektedir. On üçüncü blok evrişim katmanında 3x3 boyutundaki filtreler için adım sayısı [1 1], piksel ekleme 'same' olarak, 1x1 boyutunda filtrelerde adım sayısı [1 1], piksel ekleme [0 0 0 0] olarak ayarlanmıştır. On dördüncü blok evrişim katmanı, girdi olarak on üçüncü blok evrişim katmanından sonra RELU, normalleştirilme ve havuzlama işlemlerinin çıktısını almaktadır. 1x1, 3x3 boyutlarında 1536 çekirdekli ve 1x1 boyutunda 2048 evrişim filtreleri içermektedir. On dördüncü blok evrişim katmanında 3x3 boyutundaki filtreler için adım sayısı [1 1], piksel ekleme 'same' olarak, 1x1 boyutunda filtrelerde adım sayısı [1 1], piksel ekleme [0 0 0 0] olarak ayarlanmıştır.



Şekil 16. Xception ağ yapısı.

Sınıflandırma yöntemleri

k en yakın komşu sınıflandırma yöntemi

k en yakın komşu (kNN) yöntemi eğitim setinden elde edilen özelliklere göre test veri setini sınıflandırmak için yaygın olarak kullanılan bir makine öğrenme metodudur (Lubaib and Muneer 2016).

Regresyon ve sınıflandırma için kullanılabilir, ancak daha çok sınıflandırma problemleri için kullanılmaktadır. kNN algoritması eğitim verileri üzerinde varsayımında bulunmamaktadır. Bu algoritma tembel öğrenci algoritması olarak da adlandırılmaktadır. Çünkü eğitim setinden hemen öğrenme gerçekleşmemektedir, bunun yerine veri setini depolamakta ve sınıflandırma sırasında veri setinde bir eylem gerçekleştirmektedir. Eğitim aşamasındaki kNN algoritması sadece veri setini depolar ve yeni veri aldığında bu verileri mevcut verilere çok benzeyen bir kategori altında sınıflandırmaktadır.

$i = 1, 2, \dots, n$ eğitim veri noktaları olsun. x , etiketin bilinmediği bir nokta olarak ele alınmaktadır ve etiket sınıfını bulmak için k en yakın komşu algoritması adımları aşağıdaki gibidir.

- Adım-1: k komşu sayısını seç.
- Adım-2: 1'den n'e kadar tüm komşu noktalarının x noktasına göre Öklid mesafelerini hesapla.

- Adım-3: Hesaplanan n tane mesafeyi artan düzende sırala.
- Adım-4: Bu sıralanmış listeden ilk k adet mesafeyi al.
- Adım-5: Bu k komşu arasında, her kategorideki veri noktalarının sayısını say (Anonymous 2021i).
- Adım-6: Yeni veri noktasını, komşu sayısının maksimum olduğu kategoriye ata.

Algoritmaya göre k komşu sayısı belirlenmektedir. Komşu sayısı sabit değildir. En iyi k seçimi verilere göre değişmektedir. İkili sınıflandırmalarda, kararların bağlanmasını önlemek için k değerini tek sayı olarak seçmek daha doğrudur. Çalışmamızdaki veri setinde k değeri 1’den 10’a kadar bütün değerler için test edilmiştir ve en iyi sonuç “k = 1” olarak belirlenmiştir. Eğitim setindeki örnekler ile test verisi arasında uzaklık fonksiyonu hesaplanmaktadır. Veriler arasındaki mesafeyi hesaplamak için Öklid yöntemi kullanılmaktadır (Anonymous 2021i). Denklem 2’de uzaklık fonksiyonu verilmiştir. Daha sonra elde edilen uzaklıklar sıralanır ve eğitim setinde test verisine en yakın k adet örnek seçilmektedir. Bu örnekler arasında en çok karşılaşılan sınıf, test verisinin sınıfı olarak kabul edilmektedir.

$$d(j, i) = \sqrt{\sum_{k=1}^n (x_{ik} - x_{jk})^2} \quad (2)$$

Burada n ise özelliklerin sayısını vermektedir.

Destek vektör makinesi sınıflandırma yöntemi

Destek vektör makinesi (DVM) yöntemi son zamanlarda yaygın olarak kullanılmaktadır. Cortes and Vapnik(1995) tarafından ortaya atılan bu yöntem, istatistiksel denetimli öğrenme algoritmasıdır.

DVM algoritmasının amacı, yeni veri noktasını kolayca doğru kategoriye koyabilmemiz için n boyutlu alanı sınıflara ayırabilen en iyi çizgi veya karar sınırını oluşturmaktır. Bu en iyi karar sınırı hiper düzlem olarak adlandırılmaktadır (Anonymous 2021j). Hiper düzlemin boyutları, veri kümesinde bulunan özelliklere bağlıdır, yani 2 özellik varsa (Şekil 16’daki gibi), o zaman hiper düzlem düz bir çizgi olmaktadır. Ve 3 özellik varsa, hiper düzlem 2 boyutlu bir düzlem olmaktadır. DVM, hiper düzlemin oluşturulmasına yardımcı olan uç noktaları (vektörleri) seçmektedir. Hiper düzleme en yakın olan ve hiper düzlemin konumunu etkileyen veri noktaları, Destek Vektörü olarak adlandırılmaktadır.

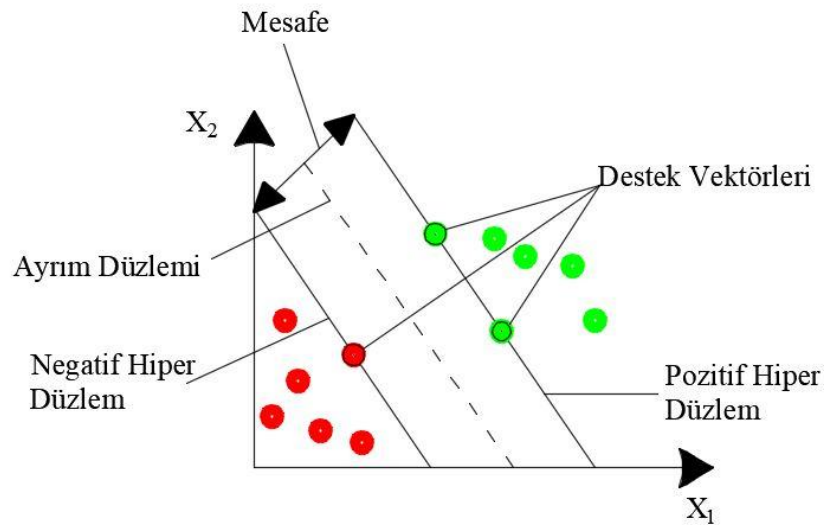
DVM yönteminin iki farklı türü olabilmektedir:

- Doğrusal DVM: Doğrusal DVM, doğrusal bir hiper düzlem ile ayrılabilen veriler için kullanılmaktadır. Yani, bir veri setindeki iki farklı sınıfı tek bir düz çizgi ile

ayrılabilirliği anlamına gelmektedir. Bu durumda bu tür verilere doğrusal olarak ayrılabilen veriler ve yöntem, Doğrusal DVM yöntemi olarak adlandırılmaktadır.

- Doğrusal olmayan DVM: Doğrusal olmayan DVM, doğrusal olarak ayrılamamış veriler için kullanılmaktadır; bu, bir veri kümesinin düz bir çizgi kullanılarak sınıflandırılmaması anlamına gelmektedir, bu durumda bu tür veriler doğrusal olmayan veri olarak adlandırılmakta ve kullanılan sınıflandırıcı, Doğrusal Olmayan DVM sınıflandırıcı olarak adlandırılmaktadır (Garg and Dhiman 2020).

Şekil 17’de iki etikete (yeşil ve kırmızı) sahip bir veri setinde (x_1, x_2) koordinat çiftini yeşil veya kırmızı olarak sınıflandırabilen bir Doğrusal DVM sınıflandırıcı gösterilmektedir. 2 boyutlu uzay olduğundan, sadece düz bir çizgi, bu iki sınıfı birbirinden ayırabilmektedir. Ancak bu sınıfları ayırabilecek birden fazla çizgi kullanılabilir. DVM algoritmasıyla eğitim veri setiyle elde edilmiş iki sınıfa ait negatif hiper düzlem ve pozitif hiper düzlem olmak üzere iki hiper düzlem hesaplanmaktadır. Elde edilen bu iki hiper düzleme eşit uzaklıkta olan iki sınıf arasında en iyi ayrımı yapabilen uygun hiper düzlem elde edilmektedir. Daha sonra (x_1, x_2) koordinat çifti elde edilen düzlem ile sınıflandırılır (Lubaib and Muneer 2016).

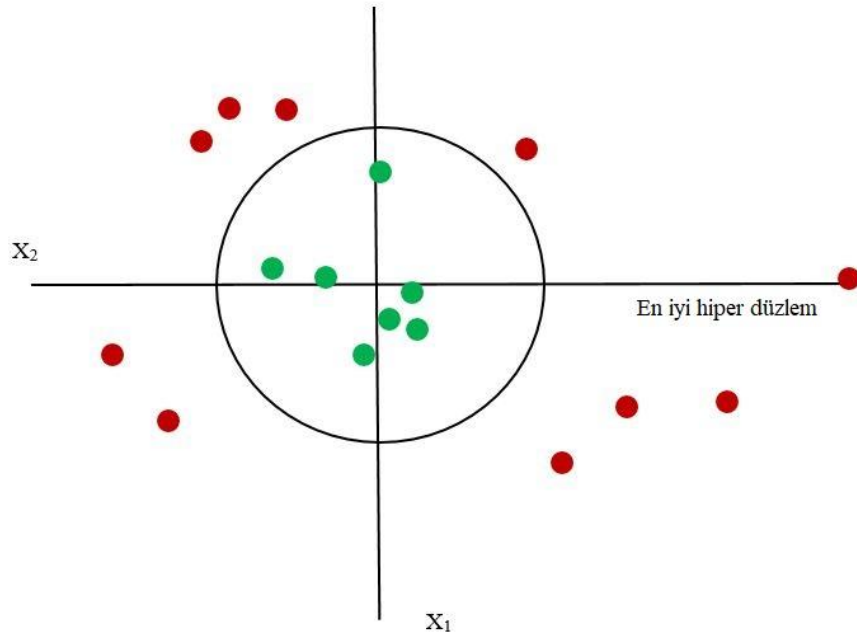


Şekil 17. Destek vektörleri ve ayırım düzlemi (Byun and Lee).

Veriler doğrusal olarak düzenlenmişse, düz bir çizgi ayırabilmektedir, ancak doğrusal olmayan veriler için tek bir düz çizgi çizilememektedir. Şekil 18’de Doğrusal Olmayan DVM sınıflandırıcısı gösterilmektedir. Doğrusal olmayan veriler için üçüncü bir boyut z eklenmektedir. z , Denklem 3 ile hesaplanmaktadır.

$$z = x_1^2 + x_2^2 \quad (3)$$

2 boyutlu uzayda $z = 1$ ile dönüştürülerek Şekil 18’deki gibi bir görüntü oluşmaktadır.



Şekil 18. Doğrusal olmayan DVM

Karar ağaçları sınıflandırma yöntemi

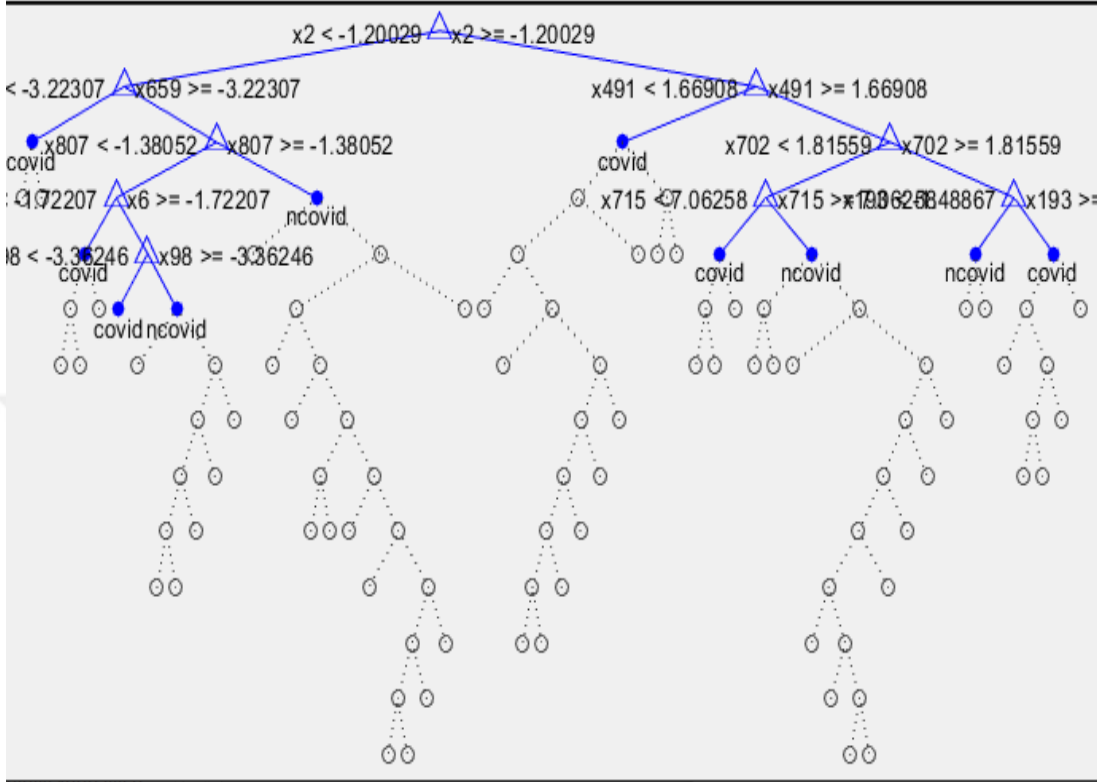
Karar Ağaçları (KA) yöntemi, sınıflandırma bakımından yapay sinir ağlarının aksine kolay ve yaygın yöntemdir (Nisbet *et al.* 2009). Bu yöntem, regresyon ve sınıflandırma problemlerinde kullanılan denetimli öğrenme modelidir.

Karar Ağacı kullanmanın amacı, eğitim verilerinden çıkarılan basit karar kurallarını öğrenerek hedef örneğin sınıfını veya değerini tahmin etmek için kullanılabilecek bir eğitim modeli oluşturmaktır.

Şekil 19'da çalışmamızda kullanılan karar ağacı görüntüsü verilmiştir. Şekilde görüldüğü gibi KA, bir örneğin sınıf etiketini tahmin etmek için ağacın kökünden başlamaktadır. Kök Düğüm: Tüm popülasyonu veya numuneyi temsil etmektedir ve ayrıca iki veya daha fazla homojen kümeye ayrılmaktadır. Başlangıçta tüm eğitim seti kök olarak kabul edilmektedir. Kök özellik değerlerini, örneğin özellikleri ile karşılaştırılmaktadır. Yani, bu değere karşılık gelen dalı izlemek ve bir sonraki düğüme atlamaktır (Anonymous 2021k).

KA algoritması iki aşamada gerçekleşir. İlk olarak eğitim veri setinden elde edilen özellikler kullanılarak ağacın bir düğümü oluşturulmaktadır. Daha sonra bu özelliğe dayalı olarak bu verileri iki yeni düğüm şeklinde bölme işlemi gerçekleştirilmektedir. Bölme: Bir düğümü iki veya daha fazla alt düğüme bölme işlemidir. Bu düğümlerden, ağacın yaprakları oluşmaktadır. Her bir dal sadece bir sınıfla etiketli oluncaya kadar bölünme işlemi devam etmektedir.

İkinci aşamada ağacın gereksiz dalları budanmaktadır. Budama: Karar düğümünün alt düğümlerini kesmeye budama denir. Bölme işleminin tersidir. Test verileri ile Karar Ağacının en üstünde olan kök düğümünden başlayarak, dalın bütün olası çıktıları aracılığıyla karar düğümleri test edilmektedir. (Bramer 2007; Nisbet *et al.* 2009).



Şekil 19. Karar ağacı.

Örnekler, özellik değerlerine göre özyinelemeli olarak dağıtılmaktadır. Özellikler ağaca yerleştirilirken kök veya iç düğümü olarak belirlenmesi, bazı istatistiksel yaklaşımlar kullanılarak yapılmaktadır (Anonymous 2021k). Bu yaklaşımlar;

Entropi: İşlenmekte olan bilgidaki rastgeleliğin bir ölçüsüdür. Entropi ne kadar yüksekse, bu bilgiden herhangi bir sonuç çıkarmak o kadar zor olmaktadır (Anonymous 2021k). Olasılık 0 veya 1 olduğunda sonuç kesin olarak bellidir, bir rastgelelik olmadığı için entropi değeri sıfırdır. Olasılık 0,5 olduğunda entropi maksimumdur. Çünkü rastgele bir sonuç üreterek sonucu kesin bir şekilde belirleyemez. Tek özellik için entropi hesabı Denklem 4'deki gibi hesaplanmaktadır.

$$E(S) = \sum_{i=1}^c -p_i \log_2 p_i \quad (4)$$

S mevcut örnek, c sınıf sayısı ve p_i örneğin i sınıfına ait olma olasılığı olarak verilmektedir.

Çoklu özellikler için entropi hesabı Denklem 5'deki gibi hesaplanmaktadır.

$$E(T, X) = \sum_{c \in X} P(c)E(c) \quad (5)$$

T mevcut özellik ve X seçilen özellik olarak verilmektedir.

Bilgi kazanımı: Belirli bir özelliğin hedef sınıflandırmalarına göre ne kadar iyi ayırdığının bir ölçütüdür. Bir KA oluşturmak, en yüksek bilgi kazancını ve en küçük entropiye sahip bir özelliği bulmakla ilgilidir (Anonymous 2021k). Bilgi kazancı ve entropi ters orantılıdır. Bilgi kazanımı; bölünmeden önceki entropi ile veri kümesinin bölünmesinden sonraki ortalama entropi arasındaki farkı, verilen özellik değerlerine göre hesaplanmaktadır. Bilgi kazanımı Denklem 6'daki gibi hesaplanmaktadır.

$$IG(T, X) = E(T) - E(T, X) \quad (6)$$

Bu yaklaşımlar, her özellik için değerleri hesaplamaktadır. Değerler sıralanır ve özellikler sıraya göre ağaca yerleştirilmektedir (Anonymous 2021k). Yani en yüksek bilgi kazanımı ve en küçük entropiye sahip özellik köke yerleştirilmektedir.

Rastgele orman sınıflandırma yöntemi

Rastgele Orman (RO), regresyon ve sınıflandırma problemleri için Breiman (2001) tarafından geliştirilen denetimli bir öğrenme modelidir. Bu model, her biri aynı düğümlere sahip, ancak farklı yapraklara yol açan farklı verileri kullanan farklı karar ağaçlarından oluşmaktadır. Tüm bu KA'ların ortalamasını temsil eden bir cevap bulmak için çoklu KA'ların kararlarını birleştirmektedir (Anonymous 2021l).

RO yöntemi, toplu öğrenme olarak kabul edilmektedir. Yani birden fazla KA'nın sonuçlarını kullanarak ortalama bulduğu için daha doğru sonuçlar elde edilmektedir.

Sınıflandırıcıyı eğitmek için, yer değiştirmeli orijinal eğitim veri setinden birden fazla ön yükleme eğitim veri seti oluşturulmaktadır. Her ön yükleme eğitim veri seti için bir ağaç oluşturulmaktadır. Mevcut ağaç için orijinal eğitim seti değiştirilmekte ve örneklenirken, vakaların yaklaşık üçte biri örneklemden çıkarılmaktadır. Bu torba dışı veriler, ağaçlar ormana eklendikçe sınıflandırma hatasının bir tahminini elde etmek için kullanılmaktadır. Yanlış sınıflandırılan örneklerin ağırlığı bir sonraki iterasyonda daha önemli olacak şekilde artırılmaktadır (Breiman and Cutler 2004).

Sınıflandırma verilerine dayalı olarak Rastgele Ormanlar gerçekleştirirken, genellikle Gini indeksi hesaplanmaktadır. Karar Ağacı dalındaki düğümlerin nasıl düğümlendiğine karar vermek için kullanılmaktadır (Anonymous 2021l). Denklem 7'de Gini indeksi formülü verilmektedir.

$$Gini = 1 - \sum_{i=1}^c (p_i)^2 \quad (7)$$

Bu denklemde bir düğümdeki her dalın Gini'sini belirlemek için sınıf ve olasılığı kullanılmaktadır. Hangi dalların oluşma olasılığının daha yüksek olduğu belirlenmektedir (Anonymous 2021). Burada örneğin i sınıfına ait olma olasılığını, c ise sınıf sayısını temsil etmektedir.

Ayrıca, düğümlerin bir Karar Ağacı'nda nasıl dallanması gerektiğine karar vermek için Denklem 4'deki entropi kullanılmaktadır.

Naive Bayes sınıflandırma yöntemi

Naive Bayes (NB) yöntemi, bağımsız varsayıma dayanan basit istatistiksel yaklaşımlı bir sınıflandırma yöntemidir. Bu yöntem, bir sınıfın belirli bir özelliğinin olmasının herhangi bir başka özelliğın olmasına bağılı olmadığını kabul etmektedir. NB yönteminin avantajı, sınıflandırma için az sayıda eğitim verisi gerektirmesidir (Murty and Devi 2011).

Naive Bayes sınıflandırıcıları, Bayes Teoremine dayalı bir sınıflandırma algoritmaları toplamından oluşmaktadır (Anonymous 2021). Bayes Teoremi, daha önce meydana gelmiş başka bir olayın olasılığı göz önüne alındığında, bir olayın meydana gelme olasılığını bulmaktadır. Denklem 8'de Bayes Teoreminin matematiksel ifadesi verilmektedir.

$$P(A | B) = \frac{P(B | A)P(A)}{P(B)} \quad (8)$$

$P(A | B)$, B'nin doğru olması durumunda A olayının gerçekleşme olasılığını temsil etmektedir (Jahanbani Fard 2015).

$P(B | A)$, A'nın doğru olduğu göz önüne alındığında, B olayının gerçekleşme olasılığını temsil etmektedir.

$P(A)$ ve $P(B)$, verilen koşullar olmadan sırasıyla A ve B'yi gözlemleme olasılıklarını temsil etmektedir.

NB sınıflandırmasında sisteme belirli bir oranda öğretilmiş veri sunulmaktadır. Öğretilmiş veriler üzerinde olasılık işlemleri uygulanmaktadır. Test verileri elde edilmiş olasılık değerlerine göre değerlendirilerek bu verilerin hangi sınıfa ait olduğu tespit edilmeye çalışılmaktadır.

NB yönteminde $X = \{x_1, x_2, \dots, x_n\}$ hangi sınıfa ait olduğu bilinmeyen veri örneğidir. $C_1, C_2, \dots, C_m$ olası sınıf değerleri olsun. X'in özelliklerinin sınıflara göre olasılıkları Denklem 9'daki gibi hesaplanmaktadır.

$$P(X|C_i) = \prod_{k=1}^n P(x_k | C_i) \quad (9)$$

X'in her sınıfa ait olma olasılığı Denklem 10'daki gibi hesaplanmaktadır. Bu durumda en büyük sonuca göre X'in hangi sınıfta ait olduğu karar verilmektedir.

$$P(C_i|X) = \frac{P(X|C_i) P(C_i)}{P(X)} \quad (10)$$

Naive Bayes yönteminin popüler çeşitleri;

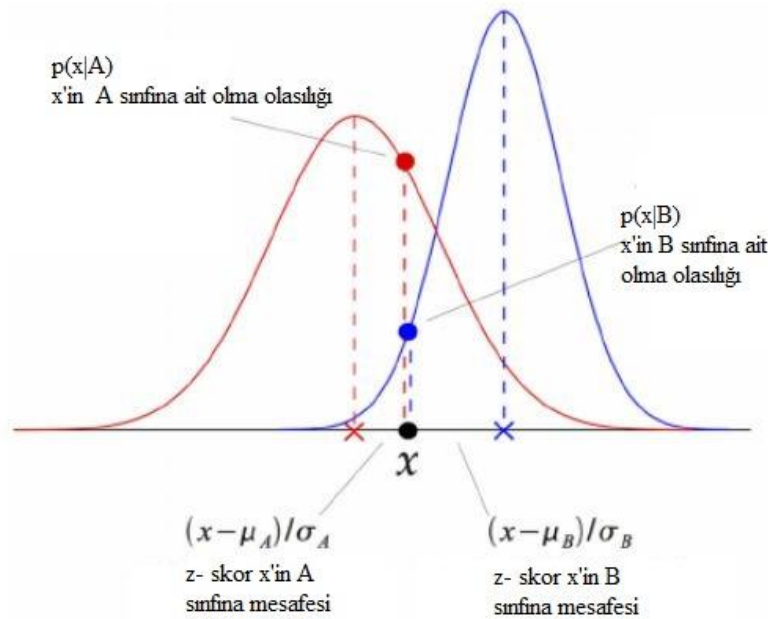
Gaussian Naive Bayes: Bu yöntemde, her bir özellik ile ilişkili sürekli değerlerin bir Gauss dağılımına göre dağıtıldığı varsayılmaktadır (Padmanaban 2014). Şekil 20'de Gauss dağılımı verilmektedir. Özelliklerin olasılığı Denklem 11'deki gibi hesaplanmaktadır.

$$P(x_i | y) = \frac{1}{\sqrt{2\pi\sigma^2_y}} \exp\left(-\frac{(x_i - \mu_y)^2}{2\sigma^2_y}\right) \quad (11)$$

y sınıf etiketini, μ ortalamayı ve σ standart sapmayı temsil etmektedir.

Çok terimli Naive Bayes: Özellik vektörleri, çok terimli bir dağılım tarafından belirli olayların üretildiği frekansları temsil etmektedir. Bu yöntem, genel olarak belge sınıflandırması için kullanılmaktadır.

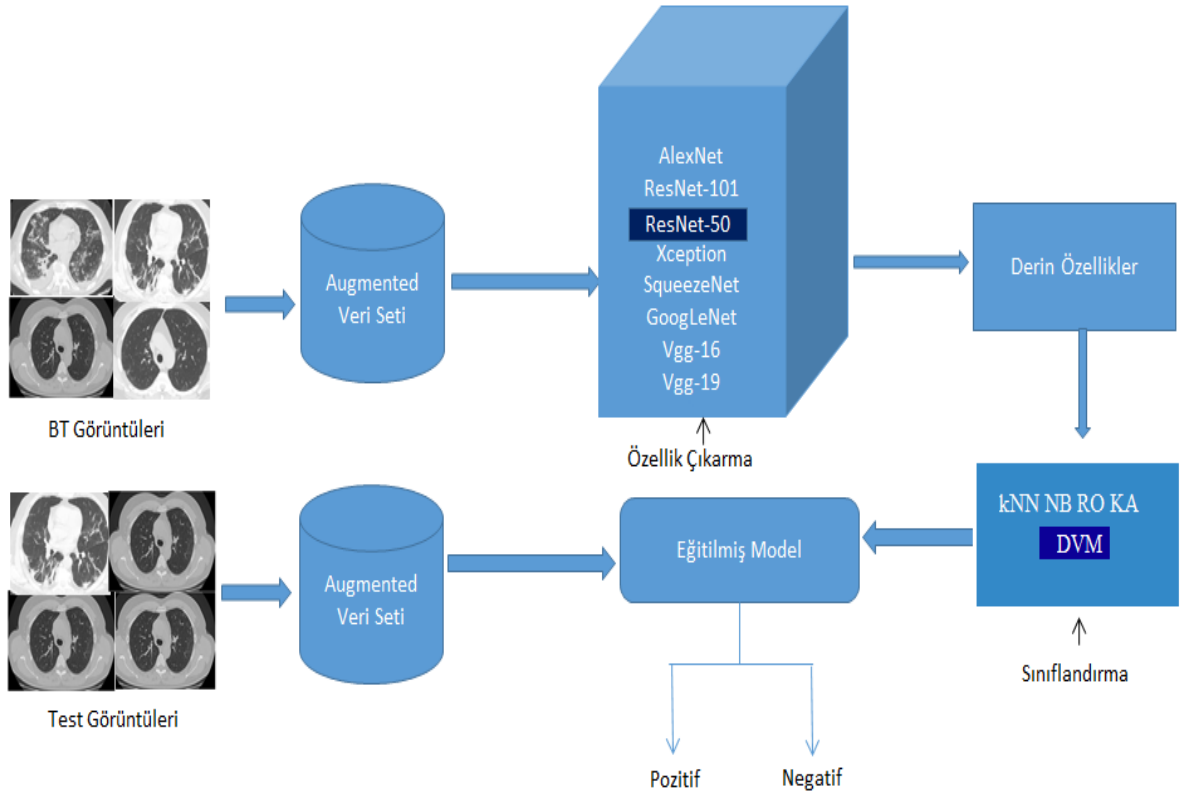
Bernoulli Naive Bayes: Çok değişkenli Bernoulli modelinde özellikler, girdileri tanımlayan bağımsız ikili değişkenlerdir. Çok terimli model gibi bu model de, terim sıklıkları (yani belgedeki bir sözcüğün sıklığı) yerine ikili terim oluşumu (yani, bir belgede bir sözcüğün geçmesi ya da olmaması) özelliklerinin kullanıldığı belge sınıflandırma görevleri için popülerdir (Anonymous 2021m).



Şekil 20. Gaussian NB yöntemi (Anonymous 2021m).

Bu çalışmanın blok diyagramı Şekil 21’de gösterilmiştir. Siirt Araştırma ve Eğitim Hastanesine Mayıs 2020 ve Haziran 2020 arasında COVID-19 şüphesiyle gelen hastalar katılımcı olarak kabul edilmektedir. COVID-19 teşhisi için BT görüntüleme tekniği uygulanmaktadır. Böylece veri toplama işlemi gerçekleşmektedir. Veri setinden eğitim için ayrılan görüntüler kullanılarak farklı derin öğrenme modellerinden (ResNet-50, ResNet-101, AlexNet, Vgg-16, Vgg-19, GoogLeNet, SqueezeNet ve Xception) ayrı ayrı derin özellikler çıkarılmaktadır. Çıkarılan bu derin özellikler çeşitli sınıflandırma yöntemlerinde (DVM, kNN, RO, KA ve NB) kullanılarak en yüksek performansa sahip derin öğrenme modeli ve sınıflandırma yöntemi ikilisi bulunmaya çalışılmaktadır. Görüntü verileri ImageDatastore nesnesi olarak depolanır. ImageDatastore, belleğe sığmayacak kadar büyük verileri toplu olarak içe aktarır. Ancak veri depolarındaki görüntülerin farklı boyutları vardır ve derin öğrenme modelinin giriş boyutlarına uygun görüntü verisi gerekmektedir. Bu nedenle elde edilen veriler, derin öğrenme modellerinin girdi katmanında yeniden boyutlandırma (OutputSizeMode), renk kanalı ön işleme (ColorPreprocessing) ve veri artırma (DataAugmentation) gibi teknikler uygulanarak uygun formata dönüştürülmektedir. Verilerin %80’i eğitim için ve %20’si test için ayrılmaktadır. Eğitim verileri, modelleri geliştirmek için kullanılmaktadır. Geliştirilen modellerin performansı test verileriyle değerlendirilmektedir.

Derin öğrenme modelinin girdi katmanında görüntüler modele uygun hale getirildikten sonra kullanılan modelin evrişim katmanlarına iletilir. Evrişim katmanında modele ait evrişim filtreleri ile özellik haritaları oluşturulur. İlk evrişim katmanında girdinin kenarları hesaplanmaktadır. Kenarlar girdinin yüksek frekanslı bölgelerini göstermektedir. Bu hesaplamalar yapıldıktan sonra çıktı boyutu giriş boyutundan farklı olacağı için piksel ekleme (Padding) yöntemi uygulanır. Böylece çıktı ve giriş boyutları eşitlenmiş olur. Daha derin katmanlar, önceki katmanların alt düzey özellikleri kullanılarak daha yüksek düzeyli özellikler oluştururlar. Derin özellikler modelin tam bağlı katmanından çıkarılmaktadır. Elde edilen derin özellikler ve eğitim verilerinin sınıf etiketleri, farklı sınıflandırma yöntemlerinde kullanılarak öğrenme gerçekleşmektedir. Daha sonra veri setinden test için ayrılan görüntülerin derin özellikleri, derin öğrenme modelleri ile çıkarılmaktadır. Çıkarılan derin özellikler eğitilmiş sınıflandırma yöntemlerinde kullanılarak tahmin işlemi gerçekleşmektedir. Son olarak test verilerinin sınıf etiketleri ile tahmin edilen sınıf etiketleri karşılaştırılarak derin öğrenme modelleri ve sınıflandırma yöntemleri doğruluk, duyarlılık, özgüllük, F1 skor, çalışma zamanı ve AUC gibi performans metrikleriyle değerlendirilmektedir.



Şekil 21. Çalışmanın blok diyagramı.

ARAŞTIRMA BULGULARI VE TARTIŞMA

Bu çalışmada doktorların COVID-19 olası vakalarına PCR testi yapılıp yapılmamasına karar vermelerine yardımcı olmak için ResNet-50, ResNet-101, AlexNet, Vgg-16, Vgg-19, GoogLeNet, SqueezeNet, Xception gibi derin öğrenme modelleri ve sınıflandırma yöntemleri uygulanarak sonuçlar karşılaştırılmıştır. Yapılan çalışma Intel(R) Core (TM) i5-3210M CPU @ 2,50 GHz. 4 GB RAM özellikli bir bilgisayarda MATLAB R2020b üzerinde gerçekleştirilmiştir. Raporlanan deney sonuçlarında kullanılan veri seti 1345 adet toraks BT görüntüsünden oluşmaktadır. Bu veri setinin %80'i eğitim için ve %20'si test için kullanılmıştır.

Performans Metrikleri

Karışıklık matrisi eğitilmiş ağın özetidir, ağın performansını göstermektedir (Yağanoğlu and Köse 2018; Çınar and Yıldırım 2020). Genel bir karışıklık matrisi Tablo 3'de gösterilmiştir.

Tablo 3. Karışıklık Matrisi

	Tahmini COVID-19	Tahmini Normal
Gerçekte COVID-19	DP	YN
Gerçekte Normal	YP	DN

DP: Görüntünün COVID-19 olduğu tahmin edilir ve bu görüntü gerçekten COVID-19'dur.

DN: Görüntünün normal olduğu tahmin edilir ve bu görüntü normaldir.

YP: Görüntünün COVID-19 olduğu tahmin edilir, ancak bu görüntü normaldir.

YN: Görüntünün normal olduğu tahmin edilir, ancak bu görüntü COVID-19'dur.

Doğruluk: Doğruluk değeri, doğru tahmin edilen görüntü sayısının, test işleminde kullanılan toplam görüntü sayısına oranıdır (Çınar and Yıldırım 2020).

Duyarlılık: Doğru tahmin edilen pozitif değerlerin toplam pozitif değer sayısına bölünmesiyle elde edilir.

Özgüllük: Doğru tahmin edilen negatif değerlerin toplam negatif değer sayısına bölünmesiyle elde edilir.

Hassasiyet: Doğru tahmin edilen pozitif değerlerin pozitif tahmin edilenlerin toplamına oranıdır.

F1-skor: Hassasiyet ve duyarlılığın harmonik ortalamasıdır.

Doğruluk, duyarlılık, özgüllük, hassasiyet ve F1-skor denklem 12, 13, 14, 15, 16 (Bozkurt *et al.* 2018) ile hesaplanır.

$$\text{Doğruluk} = \frac{DP+DN}{DP+DN+YP+YN} \quad (12)$$

$$\text{Duyarlılık} = \frac{DP}{DP+YN} \quad (13)$$

$$\text{Özgüllük} = \frac{DN}{DN+YP} \quad (14)$$

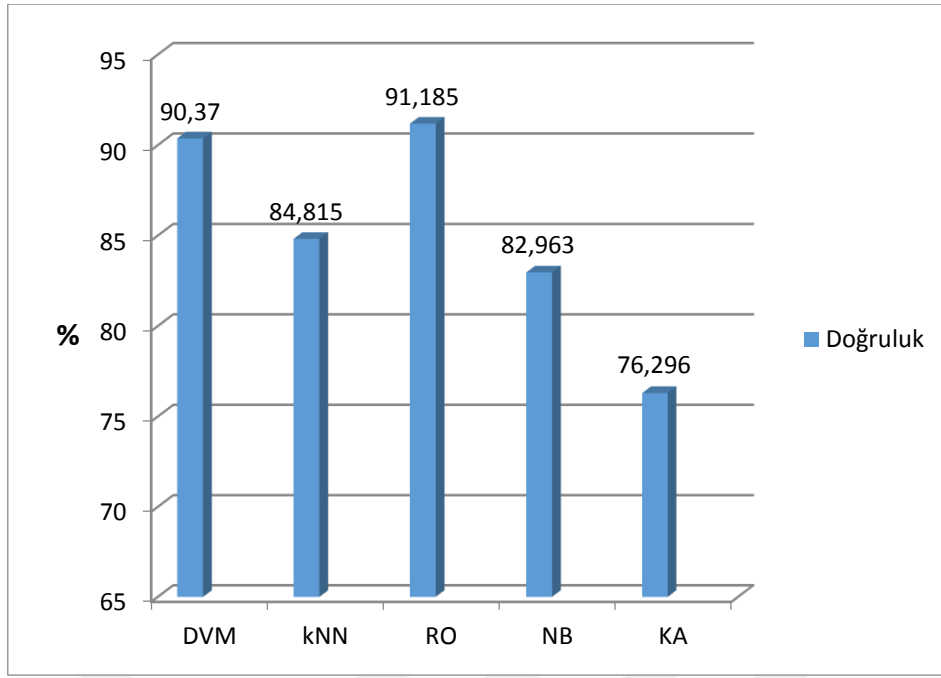
$$\text{Hassasiyet} = \frac{DP}{DP+YP} \quad (15)$$

$$F1 - \text{skor} = 2 * \frac{\text{Duyarlılık} * \text{Hassasiyet}}{\text{Duyarlılık} + \text{Hassasiyet}} \quad (16)$$

ROC eğrisi, çeşitli eşik ayarlarında yanlış pozitif oranına (1-Özgüllük) karşı doğru pozitif oranı (Duyarlılık) çizerek oluşturulmaktadır. ROC eğrisinin altında kalan alan AUC değerini vermektedir. AUC, ayrılabilirliğin derecesini veya ölçüsünü temsil etmektedir. Modelin sınıflar arasında ne kadar ayırım yapabildiğini söylemektedir. AUC ne kadar yüksekse, model negatifleri negatif ve pozitifleri pozitif olarak tahmin etmede o kadar iyidir.

ResNet-101 deney sonuçları

ResNet-101 derin öğrenme modeli ile elde edilen derin özellikler DVM, kNN, RO, KA ve NB gibi sınıflandırma yöntemlerinde kullanılarak sonuçlar karşılaştırılmıştır. Şekil 22’de görüldüğü gibi en yüksek performansı (%91,185) RO yöntemi gösterirken KA en düşük performansı (% 76,296) göstermiştir.



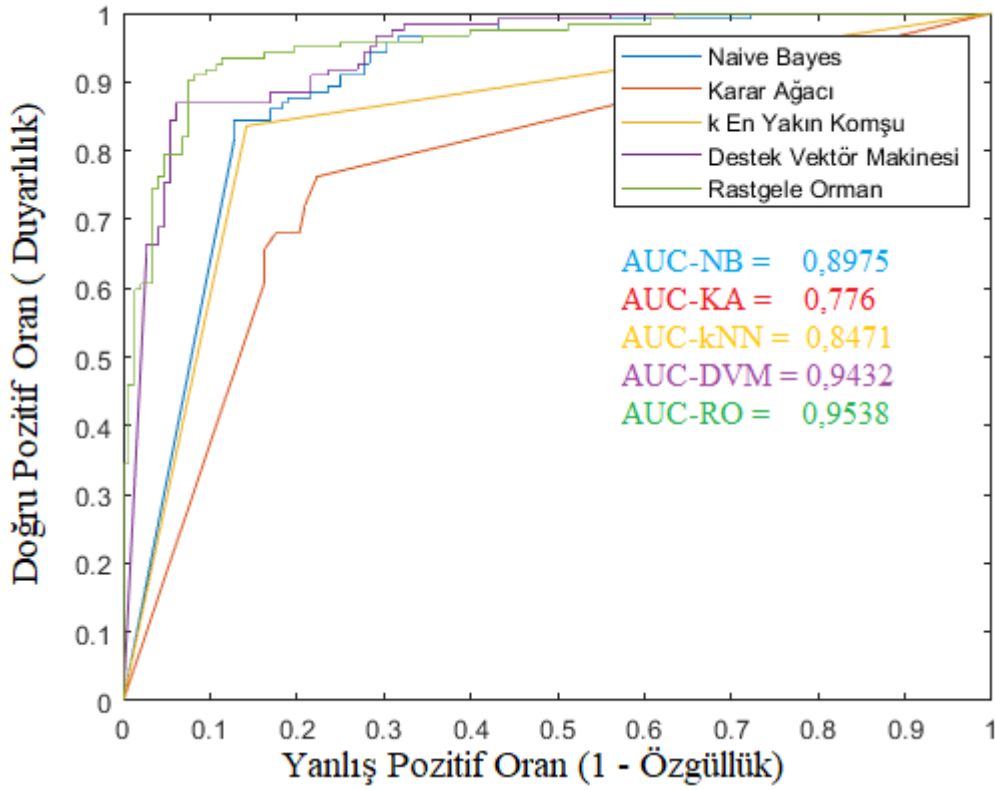
Şekil 22. ResNet-101 ile sınıflandırma yöntemlerinin doğruluk oranı.

Tablo 4’de detaylı olarak sınıflandırma yöntemlerinin sonuçları raporlanmıştır. Buna göre özgüllük değerine bakarak DVM ve RO yöntemlerinde COVID-19 negatif görüntülerini tespit etmekte diğer yöntemlerden daha başarılı oldukları açıktır. Fakat pozitif görüntülerde DVM yönteminin aynı başarıyı gösterememesi doğruluk değerini düşürmüştür. Ayrıca NB dışındaki diğer sınıflandırma yöntemlerinde özgüllük değeri duyarlılık değerinden daha yüksek olduğundan negatif görüntü tespiti pozitif görüntü tespitine göre daha başarılı olduğu söylenebilir.

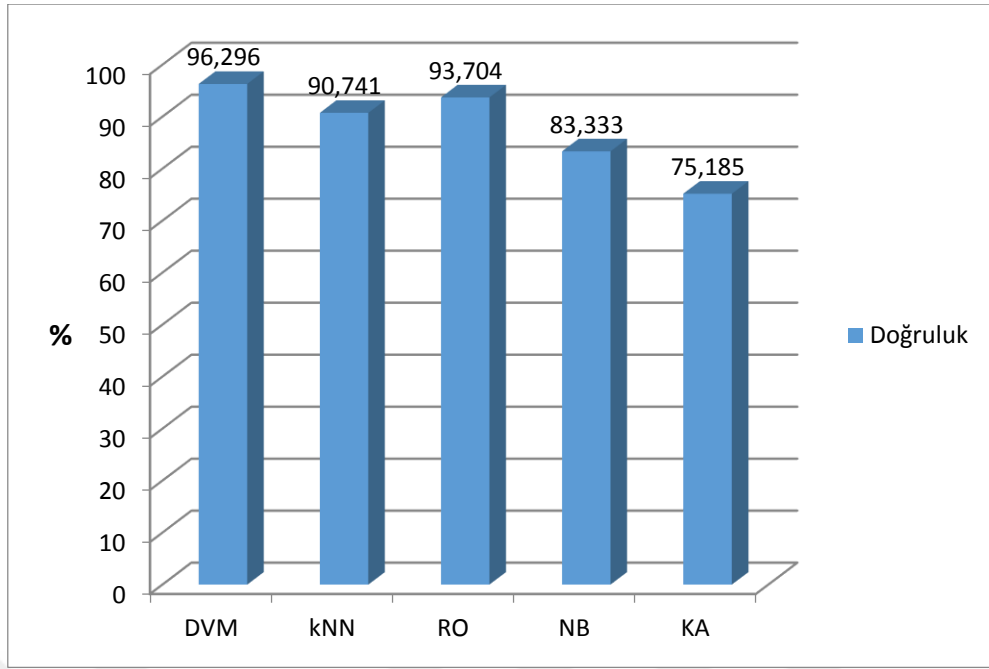
Şekil 23’de ResNet-101 ROC-eğrisi verilmiştir. ROC eğrisinin altında kalan alan AUC değerini vermektedir. AUC, ayrılabilirliğin derecesini veya ölçüsünü temsil etmektedir. Modelin sınıflar arasında ne kadar ayırım yapabildiğini söylemektedir. Şekil 23’de görüldüğü gibi AUC değeri NB yönteminde 0,8975, KA yönteminde 0,776, kNN yönteminde 0,8471, DVM yönteminde 0,9432 ve RO yönteminde 0,9538 olarak tespit edilmiştir. Şekilde iki sınıf ayırımını en iyi yapan sınıflandırma yönteminin Rastgele Orman yöntemi olduğu görülmektedir. F1-skoruna, doğruluk değerine ve ROC-eğrisine bakıldığında üç başarı kriterinde de en yüksek değerleri Rastgele Orman yöntemi göstermiştir.

Tablo 4. Resnet-101 Deney Sonuçları

	Doğruluk (%)	Duyarlılık (%)	Özgüllük (%)	F1-skor (%)	Hassasiyet (%)
NB	82,963	83,607	82,432	81,6	79,688
DVM	90,37	86,066	93,919	88,983	92,105
KNN	84,815	83,607	85,811	83,266	82,927
KA	76,296	70,492	81,081	75,439	72,882
RO	91,185	90,164	93,243	90,909	91,667

**Şekil 23.** ResNet-101 ROC-eğrisi**ResNet-50 Deney Sonuçları**

ResNet-50 derin öğrenme modeli ile elde edilen derin özellikler DVM, kNN, RO, KA ve NB gibi sınıflandırma yöntemlerinde kullanılarak elde edilen sonuçlar incelenmiştir. Şekil 24’de görüldüğü gibi en yüksek performansı (%96,296) DVM yöntemi gösterirken KA en düşük performansı (% 75,185) göstermiştir.



Şekil 24. ResNet-50 ile sınıflandırma yöntemlerinin doğruluk oranı.

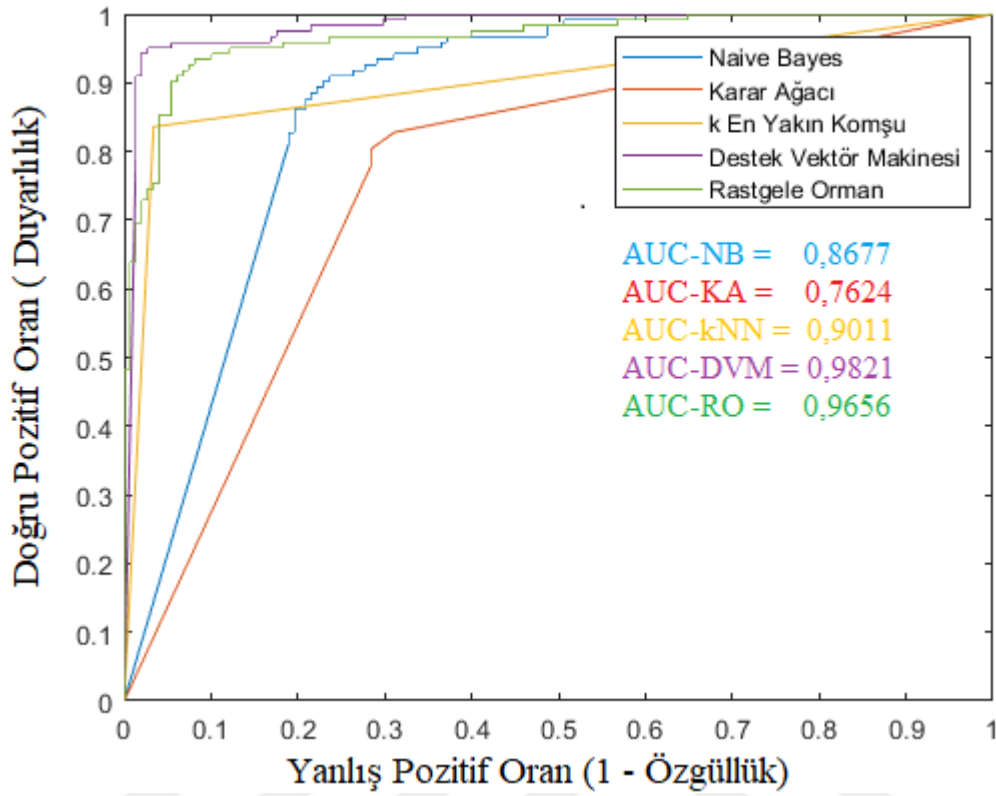
Tablo 5’de detaylı olarak sınıflandırma yöntemlerinin sonuçları verilmiştir. Buna göre özgüllük değerine bakarak DVM, RO ve kNN yöntemlerinde COVID-19 negatif görüntülerini tespit etmekte diğer yöntemlerden daha başarılı oldukları görülmektedir. Fakat pozitif görüntülerde kNN yönteminin aynı başarıyı gösterememesi doğruluk değerini düşürmüştür. DVM yöntemi hem negatif hem pozitif görüntülerin tespitinde çok yüksek başarı gösterdiği için tüm sınıflandırma yöntemlerinden daha yüksek doğruluk değeri elde etmiştir. Diğer yöntemlerin aksine NB ve KA yöntemlerinin duyarlılık değeri özgüllük değerinden yüksek çıkmıştır. Yani bu yöntemlerde COVID-19 pozitif görüntüler negatif görüntülere kıyasla daha iyi tespit edilmiştir.

Tablo 5. ResNet-50 Deney Sonuçları

	Doğruluk (%)	Duyarlılık (%)	Özgüllük (%)	F1-skor (%)	Hassasiyet (%)
NB	83,333	84,426	82,432	82,072	79,845
DVM	96,296	95,082	97,297	95,868	96,667
KNN	90,741	83,607	96,622	89,083	95,327
KA	75,185	80,328	70,946	74,525	69,504
RO	93,704	91,803	95,27	92,946	94,118

Şekil 25’de ResNet-50 ROC-eğrisi verilmiştir. ROC eğrisinin altında kalan alan AUC değerini vermektedir. AUC, ayrılabilirliğin derecesini veya ölçüsünü temsil etmektedir. Modelin sınıflar arasında ne kadar ayırım yapabildiğini söylemektedir. Şekil 25’de görüldüğü gibi AUC değeri NB yönteminde 0,8677, KA yönteminde 0,7624, kNN yönteminde 0,9011, DVM yönteminde 0,9821 ve RO yönteminde 0,9656 olarak tespit edilmiştir. Şekilde iki sınıf ayırımını en iyi yapan sınıflandırma yönteminin DVM yöntemi olduğu görülmektedir. F1-

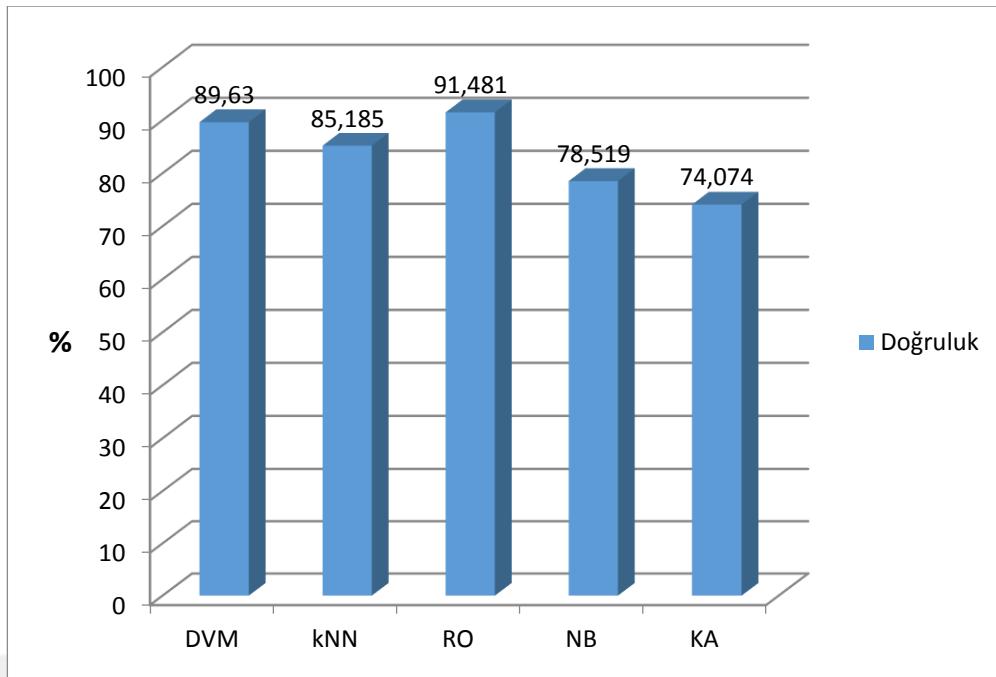
skoruna, doğruluk değerine ve ROC-eğrisine bakıldığında üç başarı kriterinde de en yüksek değerleri DVM yöntemi göstermiştir.



Şekil 25. ResNet-50 ROC-eğrisi

AlexNet Deney Sonuçları

AlexNet derin öğrenme modeli ile elde edilen derin özellikler DVM, kNN, RO, KA ve NB gibi sınıflandırma yöntemlerinde kullanılarak elde edilen sonuçlar birbiriyle karşılaştırılmıştır. Şekil 26'da görüldüğü gibi en yüksek performansı (% 91,481) RO yöntemi gösterirken KA en düşük performansı (% 74,074) göstermiştir.



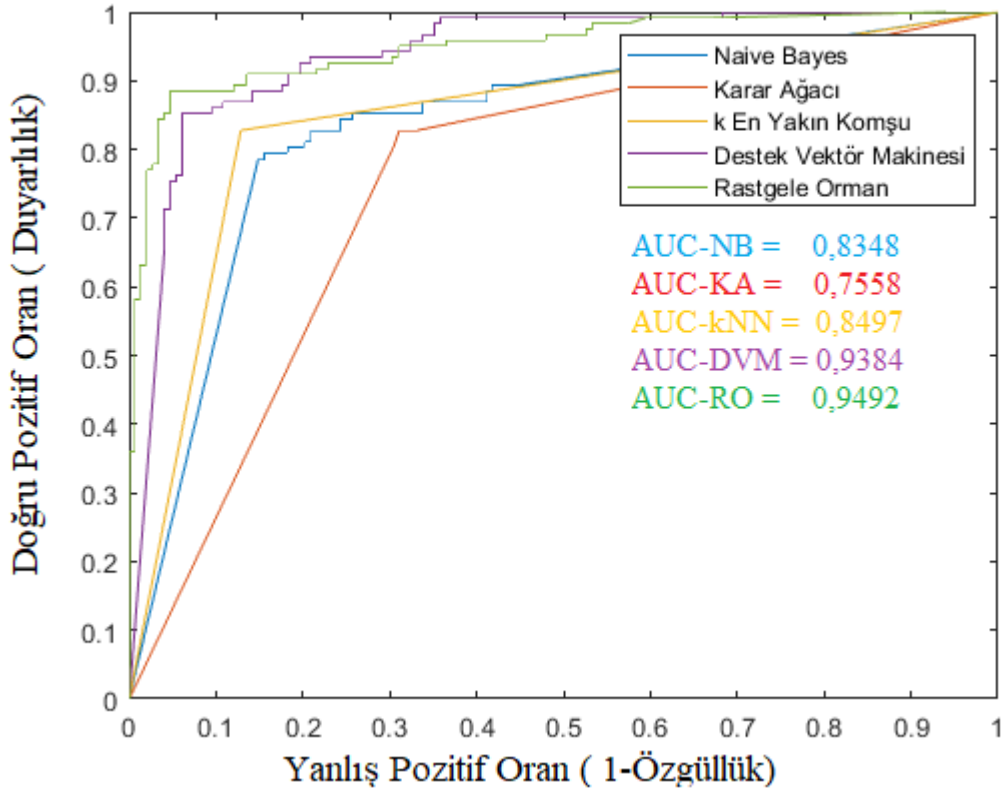
Şekil 26. AlexNet ile sınıflandırma yöntemlerinin doğruluk oranı.

Tablo 6’da detaylı olarak sınıflandırma yöntemlerinin sonuçları raporlanmıştır. Buna göre özgüllük değerine bakarak DVM ve RO yöntemlerinde COVID-19 negatif görüntülerini tespit etmekte diğer yöntemlerden daha başarılı oldukları açıktır. Fakat duyarlılık değerine bakıldığında pozitif görüntülerde DVM yönteminin aynı başarıyı gösterememesi doğruluk değerini düşürmüştür. NB ve KA yöntemlerinde duyarlılık değerinin özgüllük değerine kıyasla yüksek olması pozitif görüntü tespitinin negatif görüntü tespitine göre daha başarılı olduğunu göstermiştir.

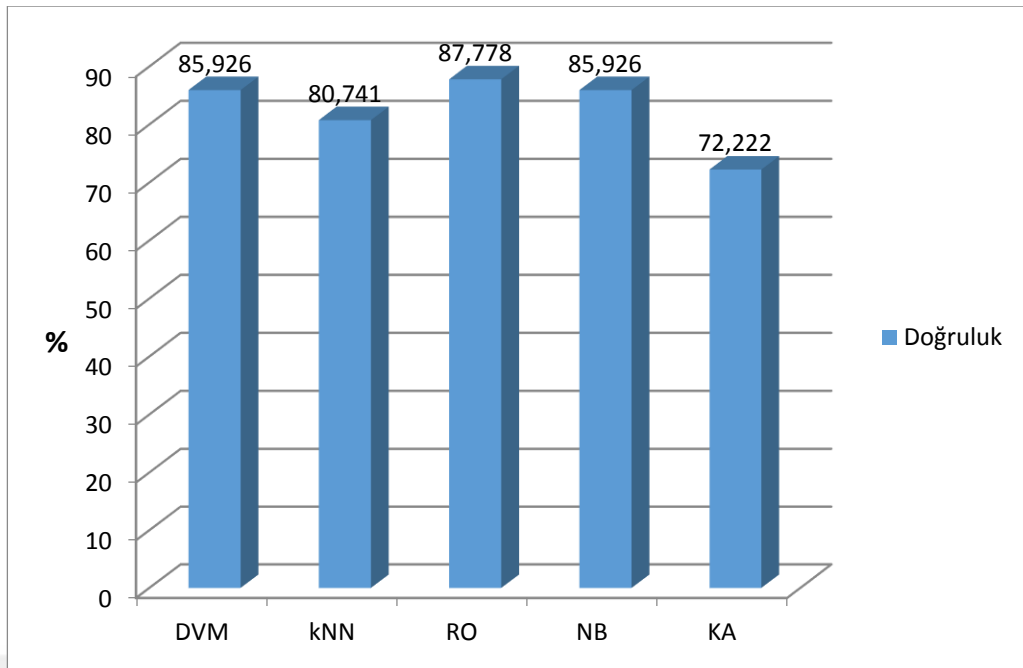
Şekil 27’de AlexNet ROC-eğrisi verilmiştir. ROC eğrisinin altında kalan alan AUC değerini vermektedir. AUC, ayrılabilirliğin derecesini veya ölçüsünü temsil etmektedir. Modelin sınıflar arasında ne kadar ayırım yapabildiğini söylemektedir. Şekil 27’de görüldüğü gibi AUC değeri NB yönteminde 0,8348, KA yönteminde 0,7558, kNN yönteminde 0,8497, DVM yönteminde 0,9384 ve RO yönteminde 0,9492 olarak tespit edilmiştir. Şekilde iki sınıf ayırımını en iyi yapan sınıflandırma yönteminin Rastgele Orman yöntemi olduğu görülmektedir. F1-skoruna, doğruluk değerine ve ROC-eğrisine bakıldığında üç başarı kriterinde de en yüksek değerleri Rastgele Orman yöntemi göstermiştir.

Tablo 6. AlexNet Deney Sonuçları

	Doğruluk (%)	Duyarlılık (%)	Özgüllük (%)	F1-skor (%)	Hassasiyet (%)
NB	78,519	79,508	77,703	76,984	74,615
DVM	89,63	85,246	93,243	88,136	91,228
KNN	85,185	82,787	87,162	83,471	84,167
KA	74,074	81,967	67,568	74,074	67,568
RO	91,481	87,705	94,595	90,295	93,043

**Şekil 27.** AlexNet ROC-eğrisi**Vgg-19 Deney Sonuçları**

Vgg-19 derin öğrenme modeli ile elde edilen derin özellikler DVM, kNN, RO, KA ve NB gibi sınıflandırma yöntemlerinde kullanılarak sonuçlar karşılaştırılmıştır. Şekil 28’de görüldüğü gibi en yüksek performansı (%87,778) RO yöntemi gösterirken KA en düşük performansı (% 72,222) göstermiştir. NB ve DVM yöntemlerinin aynı performansa (% 85,926) sahip oldukları görülmektedir.



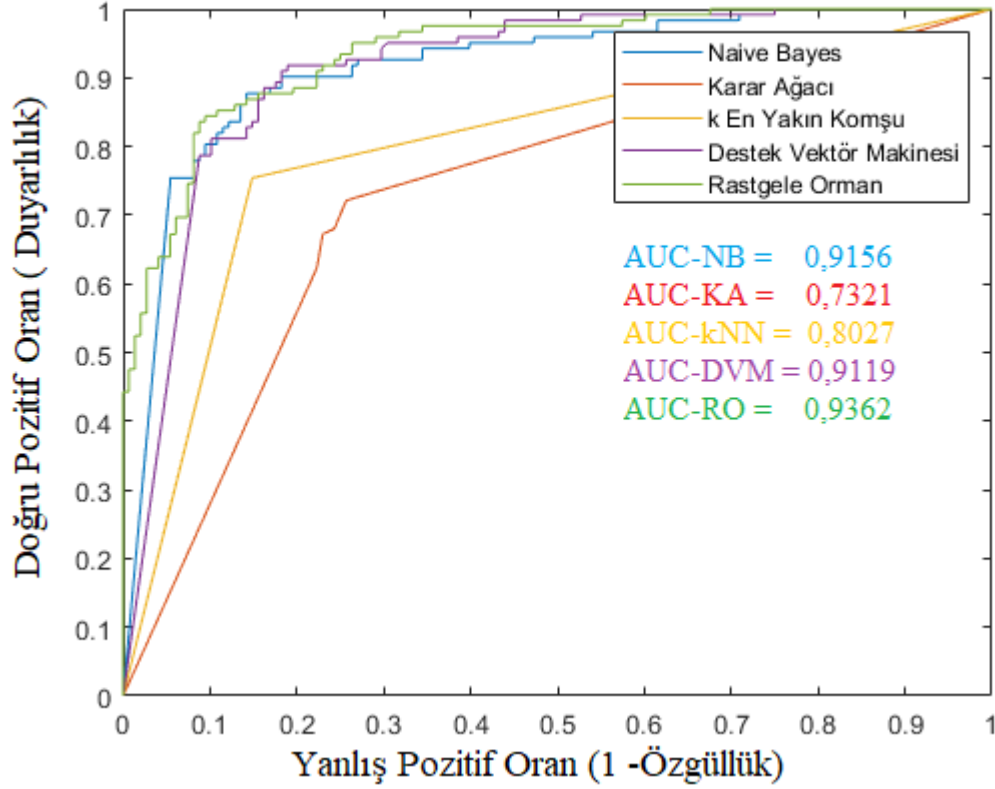
Şekil 28. Vgg-19 ile sınıflandırma yöntemlerinin doğruluk oranı.

Tablo 7’de detaylı olarak sınıflandırma yöntemlerinin sonuçları kaydedilmiştir. Buna göre özgüllük değerine bakarak NB ve RO yöntemlerinde COVID-19 negatif görüntülerini tespit etmekte diğer yöntemlerden daha başarılı oldukları görülmektedir. Duyarlılık değerine bakıldığında pozitif görüntülerin tespitinde en başarı sonuçları DVM ve RO yöntemleri elde etmiştir. RO yöntemi hem duyarlılık değerinde hem de özgüllük değerinde en yüksek sonuçları elde ettiğinden doğruluk değeri RO yönteminde diğer yöntemlerden daha yüksek çıkmıştır. Ayrıca DVM hariç diğer sınıflandırma yöntemlerinde özgüllük değeri duyarlılık değerinden daha yüksek olduğundan negatif görüntü tespiti pozitif görüntü tespitine göre daha başarılı olduğu söylenebilir.

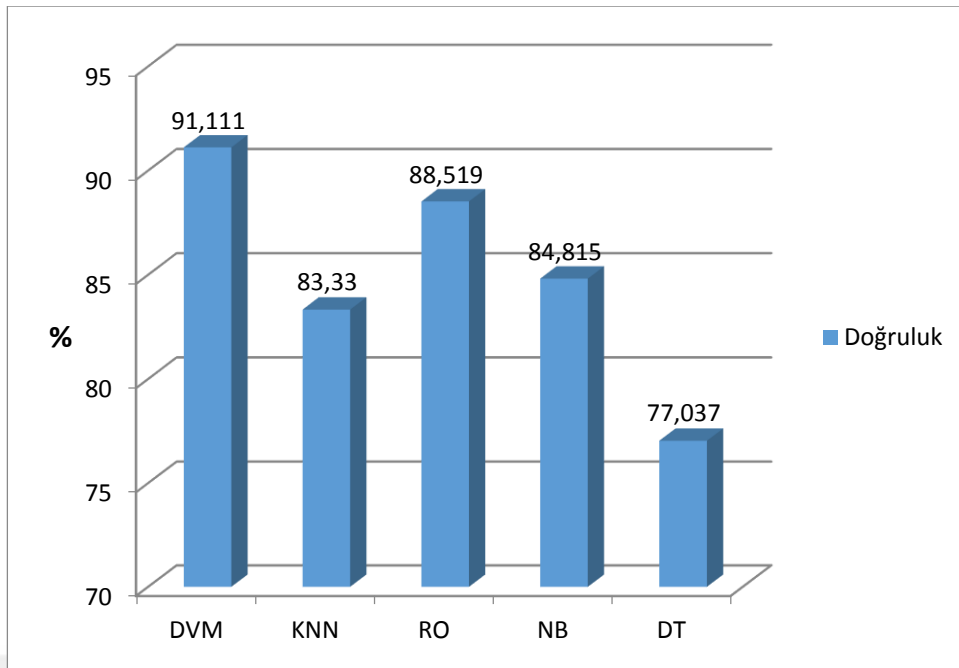
Şekil 29’da AlexNet ROC-eğrisi verilmiştir. ROC eğrisinin altında kalan alan AUC değerini vermektedir. AUC, ayrılabilirliğin derecesini veya ölçüsünü temsil etmektedir. Modelin sınıflar arasında ne kadar ayırım yapabildiğini söylemektedir. Şekil 29’da görüldüğü gibi AUC değeri NB yönteminde 0,9156, KA yönteminde 0,7321, kNN yönteminde 0,8027, DVM yönteminde 0,9119 ve RO yönteminde 0,9362 olarak tespit edilmiştir. Şekilde iki sınıf ayırımını en iyi yapan sınıflandırma yönteminin Rastgele Orman yöntemi olduğu görülmektedir. F1-skoruna, doğruluk değerine ve ROC-eğrisine bakıldığında üç başarı kriterinde de en yüksek değerleri Rastgele Orman yöntemi göstermiştir.

Tablo 7. Vgg-19 Deney Sonuçları

	Doğruluk (%)	Duyarlılık (%)	Özgüllük (%)	F1-skor (%)	Hassasiyet (%)
NB	85,926	81,967	89,189	84,034	86,207
DVM	85,926	88,525	83,784	85,039	81,818
KNN	80,741	75,41	85,135	77,966	80,702
KA	72,222	67,213	76,351	68,619	70,085
RO	87,778	86,066	89,189	86,42	86,777

**Şekil 29.** Vgg-19 ROC-eğrisi**Vgg-16 Deney Sonuçları**

Vgg-16 derin öğrenme modeli ile elde edilen derin özellikler DVM, kNN, RO, KA ve NB gibi sınıflandırma yöntemlerinde kullanılarak elde edilen sonuçlar incelenmiştir. Şekil 30'da görüldüğü gibi en yüksek performansı (% 91,111) DVM yöntemi gösterirken KA en düşük performansı (% 77,037) göstermiştir.



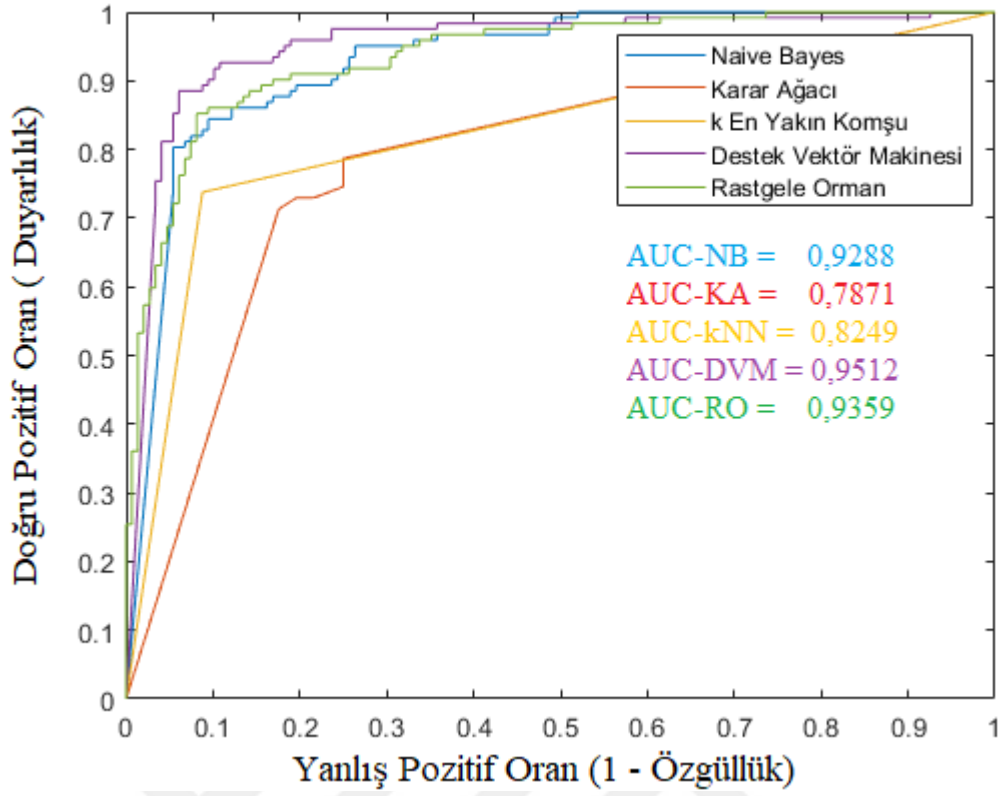
Şekil 30. Vgg-16 ile sınıflandırma yöntemlerinin doğruluk oranı.

Tablo 8’de detaylı olarak sınıflandırma yöntemlerinin sonuçları raporlanmıştır. Buna göre özgüllük ve duyarlılık değerine bakarak DVM yöntemi hem COVID-19 negatif görüntülerini tespit etmekte hem de pozitif görüntüleri tespit etmekte diğer yöntemlerden daha başarılı olduğu açıktır. Ayrıca tüm sınıflandırma yöntemlerinde özgüllük değeri duyarlılık değerinden daha yüksek sonuçlar elde ettiği için negatif görüntü tespiti pozitif görüntü tespitine göre daha başarılı olduğu söylenebilir.

Şekil 31’de Vgg-16 ROC-eğrisi verilmiştir. ROC eğrisinin altında kalan alan AUC değerini vermektedir. AUC, ayrılabilirliğin derecesini veya ölçüsünü temsil etmektedir. Modelin sınıflar arasında ne kadar ayırım yapabildiğini söylemektedir. Şekil 31’de görüldüğü gibi AUC değeri NB yönteminde 0,9288, KA yönteminde 0,7871, kNN yönteminde 0,8249, DVM yönteminde 0,9512 ve RO yönteminde 0,9359 olarak tespit edilmiştir. Şekilde iki sınıf ayırımını en iyi yapan sınıflandırma yönteminin DVM yöntemi olduğu görülmektedir. F1-skoruna, doğruluk değerine ve ROC-eğrisine bakıldığında üç başarı kriterinde de en yüksek değerleri DVM yöntemi göstermiştir.

Tablo 8. Vgg-16 Deney Sonuçları

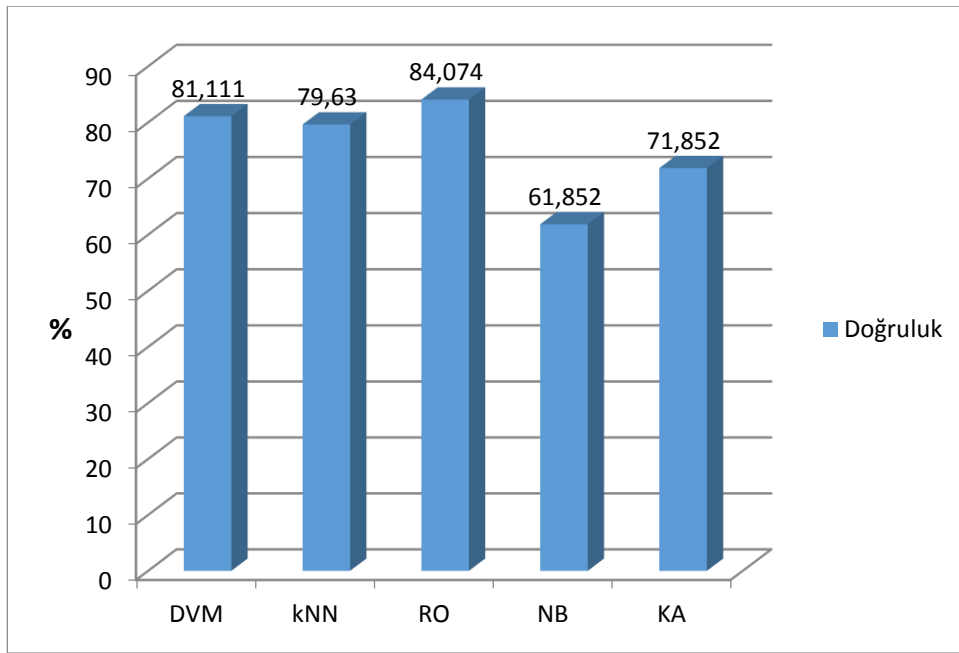
	Doğruluk (%)	Duyarlılık (%)	Özgüllük (%)	F1-skor (%)	Hassasiyet (%)
NB	84,815	79,508	89,189	82,553	85,841
DVM	91,111	88,525	93,243	90	91,525
KNN	83,33	73,77	91,216	79,999	87,379
KA	77,037	74,459	79,054	74,59	74,59
RO	88,519	85,246	91,216	88,889	87,029



Şekil 31. Vgg-16 ROC-eğrisi

GoogLeNet Deney Sonuçları

GoogLeNet derin öğrenme modeli ile elde edilen derin özellikler DVM, kNN, RO, KA ve NB gibi sınıflandırma yöntemlerinde kullanılarak sonuçlar karşılaştırılmıştır. Şekil 32’de görüldüğü gibi en yüksek performansı (% 84,074) RO yöntemi gösterirken NB en düşük performansı (%61,852) göstermiştir.



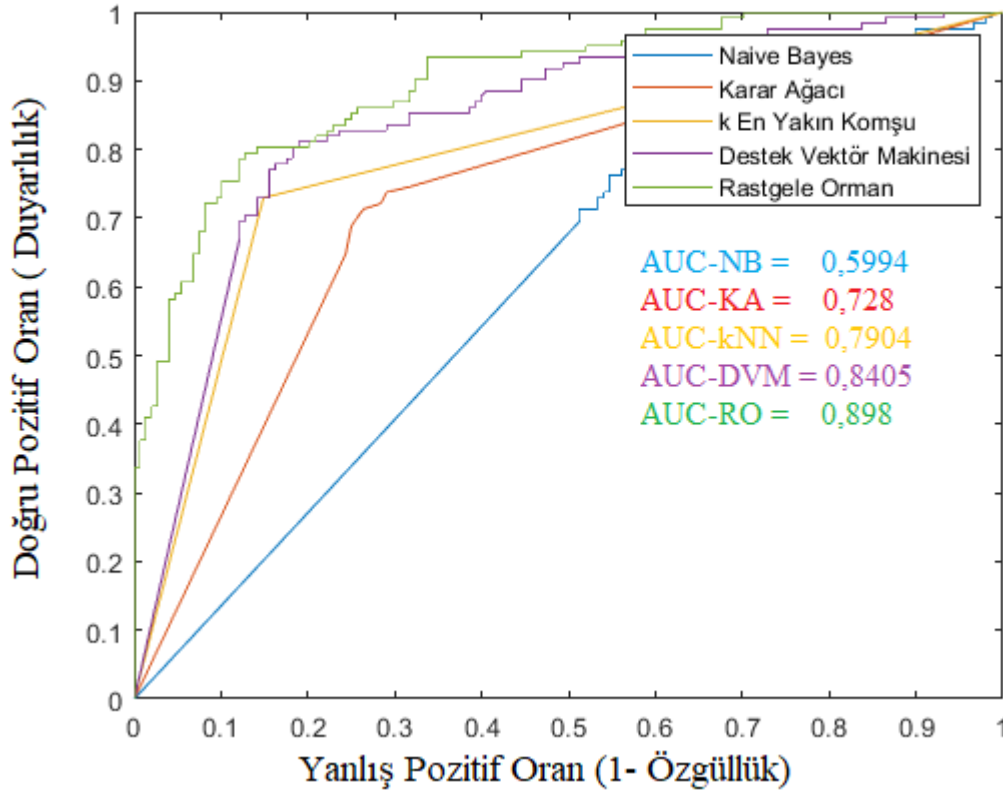
Şekil 32. GoogLeNet ile sınıflandırma yöntemlerinin doğruluk oranı.

Tablo 9’da detaylı olarak sınıflandırma yöntemlerinin sonuçları verilmiştir. Buna göre özgüllük değerine bakarak kNN ve RO yöntemlerinde COVID-19 negatif görüntülerini tespit etmekte diğer yöntemlerden daha başarılı oldukları açıktır. Fakat pozitif görüntülerde kNN yönteminin aynı başarıyı gösterememesi doğruluk değerini düşürmüştür. NB ve KA yöntemlerinin duyarlılık değeri özgüllük değerinden yüksek çıkmıştır. Yani bu yöntemlerde COVID-19 pozitif görüntüler negatif görüntülere kıyasla daha iyi tespit edilmiştir.

Şekil 33’de GoogLeNet ROC-eğrisi verilmiştir. ROC eğrisinin altında kalan alan AUC değerini vermektedir. AUC, ayrılabilirliğin derecesini veya ölçüsünü temsil etmektedir. Modelin sınıflar arasında ne kadar ayırım yapabildiğini söylemektedir. Şekil 33’de görüldüğü gibi AUC değeri NB yönteminde 0,5994, KA yönteminde 0,728, kNN yönteminde 0,7904, DVM yönteminde 0,8405 ve RO yönteminde 0,898 olarak tespit edilmiştir. Şekilde iki sınıf ayırımını en iyi yapan sınıflandırma yönteminin Rastgele Orman yöntemi olduğu görülmektedir. F1-skoruna, doğruluk değerine ve ROC-eğrisine bakıldığında üç başarı kriterinde de en yüksek değerleri Rastgele Orman yöntemi göstermiştir.

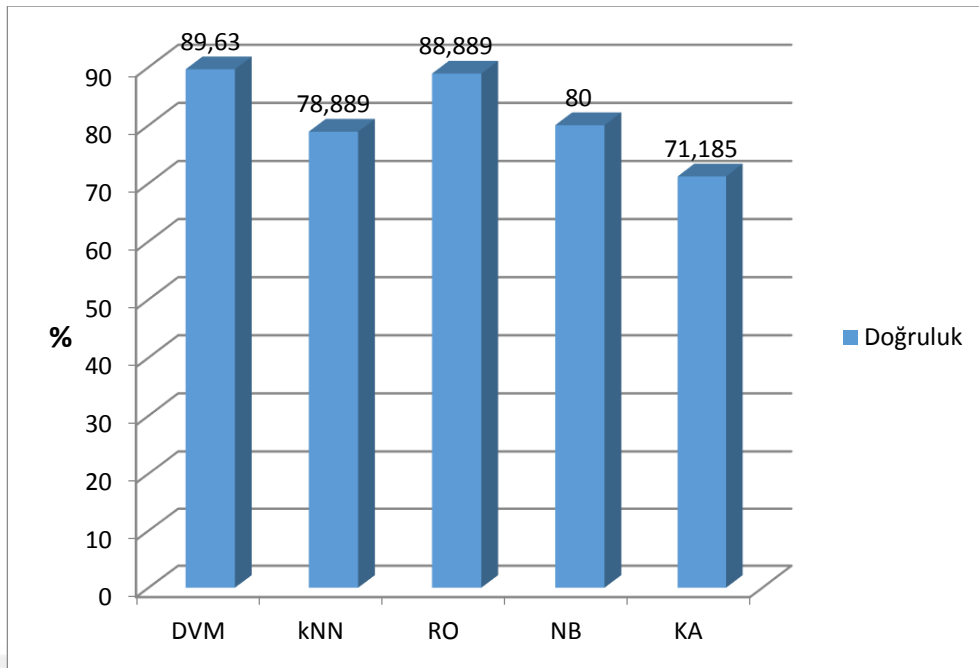
Tablo 9. GoogLeNet Deney Sonuçları

	Doğruluk (%)	Duyarlılık (%)	Özgüllük (%)	F1-skor (%)	Hassasiyet (%)
NB	61,852	77,869	48,649	64,847	55,556
DVM	81,111	81,148	81,081	79,518	77,953
KNN	79,63	72,951	85,135	76,395	80,18
KA	71,852	72,131	71,622	69,841	67,692
RO	84,074	80,328	87,162	82,009	83,761

**Şekil 33.** GoogLeNet ROC-eğrisi

SqueezeNet Deney Sonuçları

SqueezeNet derin öğrenme modeli ile elde edilen derin özellikler DVM, kNN, RO, KA ve NB gibi sınıflandırma yöntemlerinde kullanılarak sonuçlar incelenmiştir. Şekil 34’de görüldüğü gibi en yüksek performansı (% 89,63) DVM yöntemi gösterirken KA en düşük performansı (% 71,185) göstermiştir.



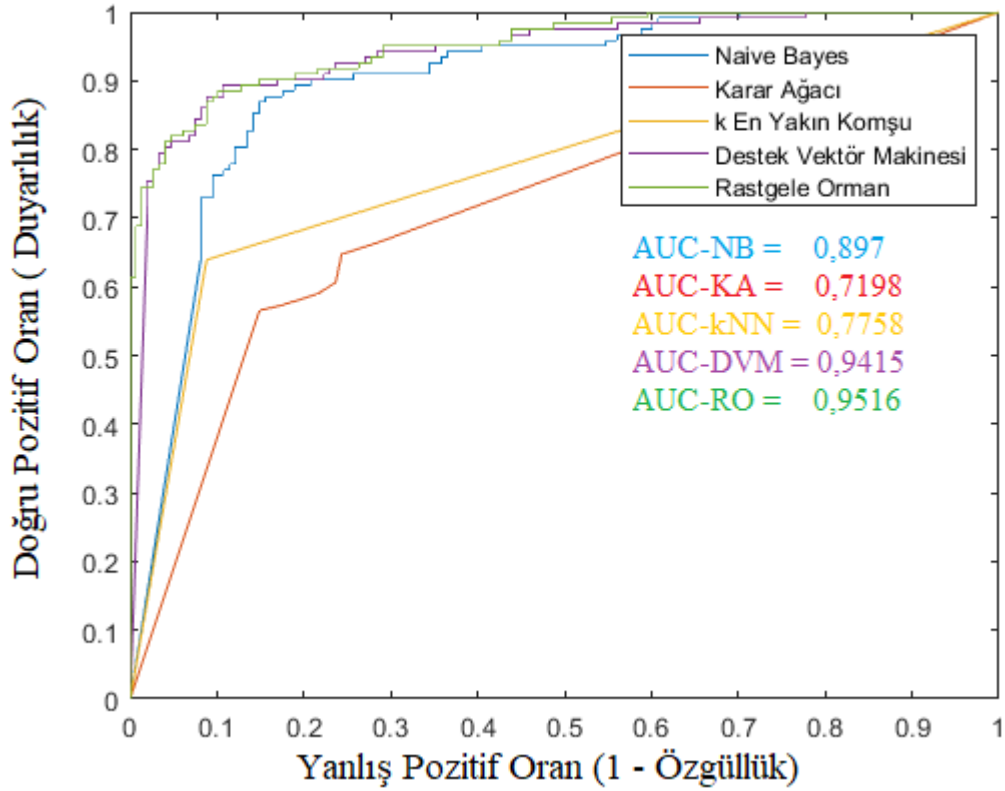
Şekil 34. SqueezeNet ile sınıflandırma yöntemlerinin doğruluk oranı.

Tablo 10’da detaylı olarak sınıflandırma yöntemlerinin sonuçları raporlanmıştır. Buna göre özgüllük değerine bakarak RO, DVM ve kNN yöntemlerinin COVID-19 negatif görüntülerini tespit etmekte başarılı oldukları açıktır. Fakat pozitif görüntülerde kNN yönteminin aynı başarıyı gösterememesi doğruluk değerini düşürmüştür. Duyarlılık değerine bakıldığında DVM yöntemi COVID-19 pozitif görüntülerini tespit etmekte diğer yöntemlere göre daha başarılı olduğu görülmüştür. Ayrıca tüm sınıflandırma yöntemlerinde negatif görüntü tespiti pozitif görüntü tespitine göre daha başarılı olmuştur.

Şekil 35’de SqueezeNet ROC-eğrisi verilmiştir. ROC eğrisinin altında kalan alan AUC değerini vermektedir. AUC, ayrılabilirliğin derecesini veya ölçüsünü temsil etmektedir. Modelin sınıflar arasında ne kadar ayırım yapabildiğini söylemektedir. Şekil 35’de görüldüğü gibi AUC değeri NB yönteminde 0,897, KA yönteminde 0,7198, kNN yönteminde 0,7758, DVM yönteminde 0,9415 ve RO yönteminde 0,9516 olarak tespit edilmiştir. Şekilde iki sınıf ayırımını en iyi yapan sınıflandırma yönteminin Rastgele Orman yöntemi olduğu görülmektedir. F1-skora ve doğruluk değerine bakıldığında en yüksek değerleri DVM yöntemi gösterirken ROC-eğrisi en yüksek değeri Rastgele Orman yönteminde göstermiştir.

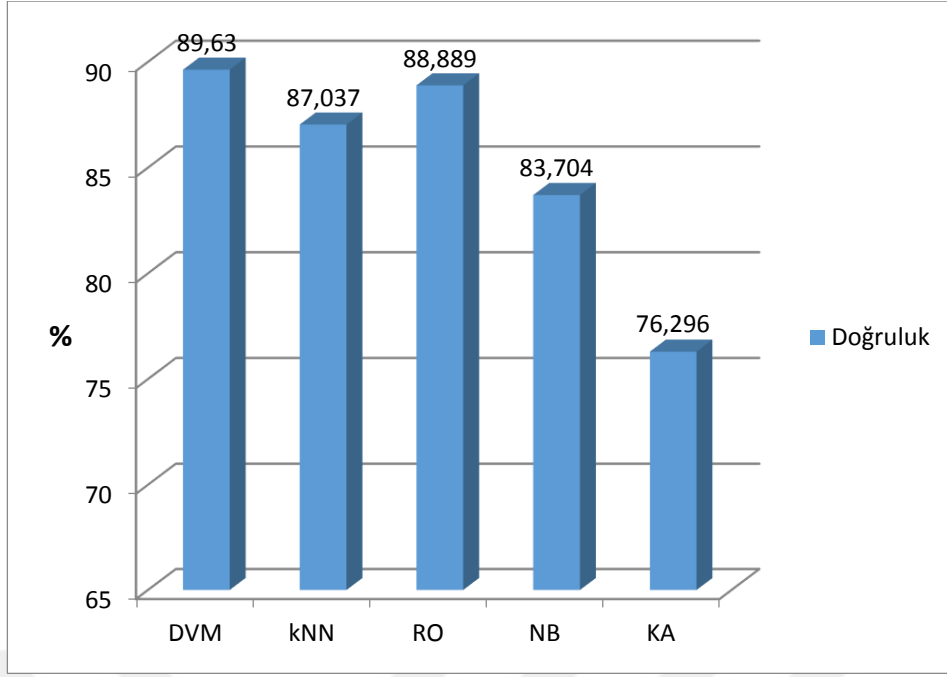
Tablo 10. SqueezeNet Deney Sonuçları

	Doğruluk (%)	Duyarlılık (%)	Özgüllük (%)	F1-skor (%)	Hassasiyet (%)
NB	80	67,213	90,541	75,229	85,417
DVM	89,63	87,705	91,216	88,43	89,167
KNN	78,889	63,934	91,216	73,239	85,714
KA	71,185	65,574	77,027	67,797	70,175
RO	88,889	80,328	95,946	86,726	94,231

**Şekil 35.** SqueezeNet ROC-eğrisi

Xception Deney Sonuçları

Xception derin öğrenme modeli ile elde edilen derin özellikler DVM, kNN, RO, KA ve NB gibi sınıflandırma yöntemlerinde kullanılarak sonuçlar karşılaştırılmıştır. Şekil 36'da görüldüğü gibi en yüksek performansı (% 89,63) DVM yöntemi gösterirken KA en düşük performansı (% 76,296) göstermiştir.



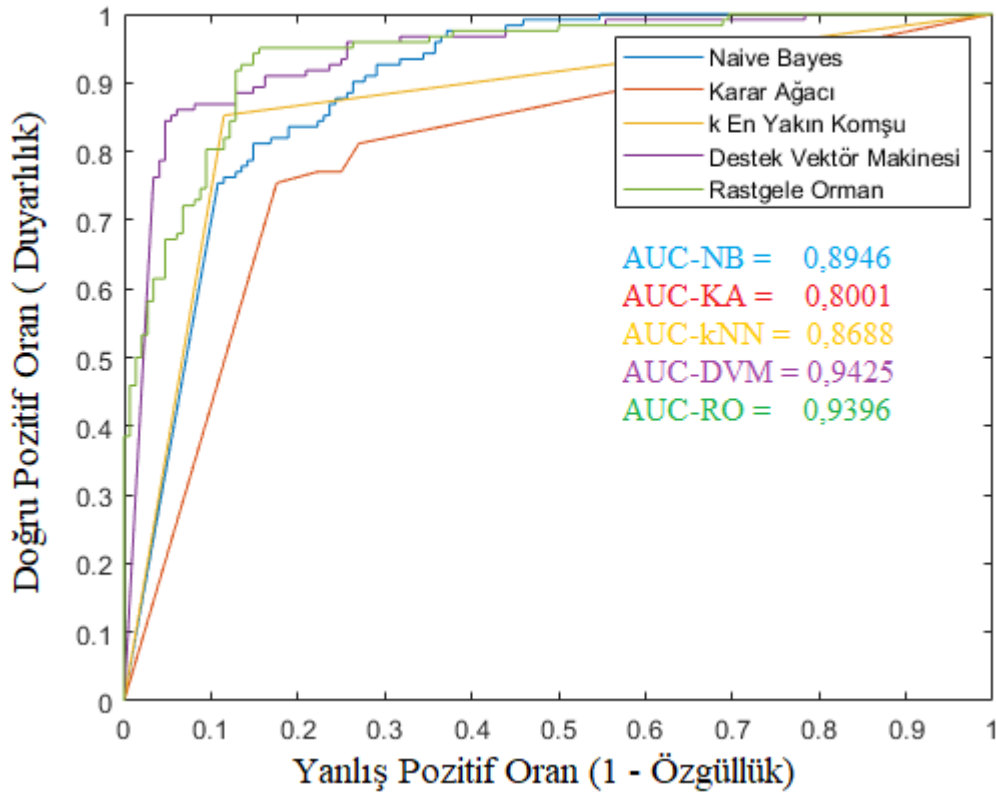
Şekil 36. Xception ile sınıflandırma yöntemlerinin doğruluk oranı.

Tablo 11’de detaylı olarak sınıflandırma yöntemlerinin sonuçları raporlanmıştır. Buna göre özgüllük değerine bakarak DVM yöntemi COVID-19 negatif görüntülerini tespit etmekte diğer yöntemlerden daha başarılı olduğu açıktır. Duyarlılık değerine bakıldığında RO yöntemi COVID-19 pozitif görüntülerini tespit etmekte diğer yöntemlere göre daha başarılı olduğu görülmektedir. Ayrıca KA ve RO yöntemleri hariç diğer sınıflandırma yöntemlerinde özgüllük değeri duyarlılık değerinden daha yüksek olduğundan negatif görüntü tespiti pozitif görüntü tespitine göre daha başarılı olduğu söylenebilir.

Şekil 37’de Xception ROC-eğrisi verilmiştir. ROC eğrisinin altında kalan alan AUC değerini vermektedir. AUC, ayrılabilirliğin derecesini veya ölçüsünü temsil etmektedir. Modelin sınıflar arasında ne kadar ayırım yapabildiğini söylemektedir. Şekil 37’de görüldüğü gibi AUC değeri NB yönteminde 0,8946, KA yönteminde 0,8001, kNN yönteminde 0,8688, DVM yönteminde 0,9425 ve RO yönteminde 0,9396 olarak tespit edilmiştir. Şekilde iki sınıf ayırımını en iyi yapan sınıflandırma yönteminin DVM yöntemi olduğu görülmektedir. F1-skoruna, doğruluk değerine ve ROC-eğrisine bakıldığında üç başarı kriterinde de en yüksek değerleri DVM yöntemi göstermiştir.

Tablo 11. Xception Deney Sonuçları

	Doğruluk (%)	Duyarlılık (%)	Özgüllük (%)	F1-skor (%)	Hassasiyet (%)
NB	83,704	83,607	83,784	82,258	80,952
DVM	89,63	86,885	91,892	88,333	89,831
KNN	87,037	85,246	88,514	85,597	85,95
KA	76,296	77,049	75,676	74,603	72,308
RO	88,889	91,803	86,486	88,189	84,848

**Şekil 37.** Xception ROC-eğrisi

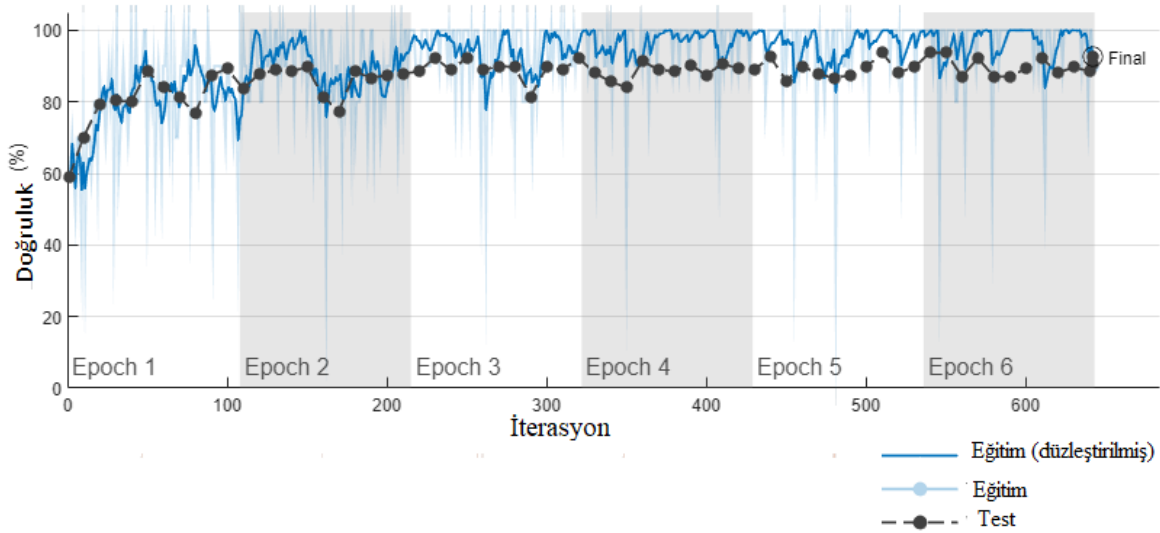
Genel olarak bakıldığında Tablo 12’de görüldüğü gibi en yüksek performans ResNet-50 ve DVM yöntemi ile elde edilmiştir. Doğruluk değeri % 96,296, F1-skor değeri 95,858 ve AUC değeri 0,9821 olarak görülmektedir. Çalışma zamanına (ÇZ) bakıldığında ResNet-50 diğer 4 modelden (ResNet-101, Xception, Vgg19 ve Vgg-16) daha hızlı çalıştığı görülmektedir. En düşük performans GoogLeNet ve NB yöntemi ile elde edilmiştir. Doğruluk değeri % 61,852, F1-skor değeri 64,847, AUC değeri 0,5994 ve çalışma süresi 81,303 sn olarak tespit edilmiştir. Tablo 12’de ÇZ değerine bakıldığında en hızlı çalışan (22,58 sn) AlexNet ve kNN yönteminin doğruluk değeri %85,249’dur. En yavaş çalışan (528,84 sn) Vgg-19 ve RO yönteminin doğruluk değeri %87,778. AlexNet, SqueezeNet ve GoogLeNet modellerinin diğer modellerden daha hızlı olduğu açıktır. Fakat SqueezeNet ve GoogLeNet modellerinin doğruluk değerleri yüksek değildir. Vgg-19 hem yavaş hemde doğruluk değeri diğer modellerden daha düşüktür. Bu durumda en iyi seçimin ResNet-50 olduğu söylenebilir. Modellerin katman sayısının az olmasına ve 1x1 filtreleme boyutlarına sahip olmasına göre

çalışma süresinin azaldığı söylenebilir. 1x1 filtreleme derinlikte boyut azaltmaktadır. Havuzlama katmanı girdide yükseklik ve genişlikte bir azaltma gerçekleştirirken kanal sayısını artırmaktadır yani derinlik artmış olur. Derin artıkçada çalışma zamanı artmaktadır. Ayrıca derin öğrenme modelleri ve sınıflandırma yöntemleri genelinde COVID-19 negatif görüntülerin daha iyi tespit edildiği söylenebilir. Bu da negatif görüntülerin pozitif görüntülerden daha fazla olduğu için negatif görüntülerde daha iyi öğrenme gerçekleştiği düşünülebilir. GoogLeNet modeli hariç diğer modellerde en düşük performans KA yöntemi ile elde edilmiştir.

Tablo 12. Deney Sonuçları

		Doğruluk (%)	Duyarlılık (%)	Özgüllük (%)	F1-skor (%)	Hassasiyet (%)	AUC	ÇZ (sn)
ALEXNET	NB	78,519	79,508	77,703	76,984	79,508	0,8348	22,672
	DVM	89,63	85,246	93,243	88,136	85,246	0,9384	49,124
	KNN	85,185	82,787	87,162	83,471	82,787	0,8497	22,582
	KA	74,074	81,967	67,568	74,074	81,967	0,7558	23,181
	RO	91,481	87,705	94,595	90,295	87,705	0,9492	71,914
RESNET50	NB	83,333	84,426	82,432	82,072	84,426	0,8677	138,44
	DVM	96,296	95,082	97,297	95,868	95,082	0,9821	159,39
	KNN	90,741	83,607	96,622	89,083	83,607	0,9011	122,76
	KA	75,185	80,328	70,946	74,525	80,328	0,7624	133,78
	RO	93,704	91,803	95,27	92,946	91,803	0,9656	145,85
RESNET101	NB	82,963	83,607	82,432	81,6	83,607	0,8975	229,34
	DVM	90,37	86,066	93,919	88,983	86,066	0,9432	260,63
	KNN	84,815	83,607	85,811	83,266	83,607	0,8471	220,09
	KA	76,296	70,492	81,081	75,439	70,492	0,776	220,51
	RO	91,185	90,164	93,243	90,909	90,164	0,9538	247,92
VGG16	NB	84,815	79,508	89,189	82,553	79,508	0,9288	362,52
	DVM	91,111	88,525	93,243	90	88,525	0,9512	414,68
	KNN	83,33	73,77	91,216	79,999	73,77	0,8249	419,68
	KA	77,037	74,459	79,054	74,59	74,459	0,7871	380,75
	RO	88,519	85,246	91,216	88,889	85,246	0,9359	387,55
VGG19	NB	85,926	81,967	89,189	84,034	81,967	0,9156	429,28
	DVM	85,926	88,525	83,784	85,039	88,525	0,9119	518,31
	KNN	80,741	75,41	85,135	77,966	75,41	0,8027	482,86
	KA	72,222	67,213	76,351	68,619	67,213	0,7321	451,48
	RO	87,778	86,066	89,189	86,42	86,066	0,9362	528,84
SQUEEZENET	NB	80	67,213	90,541	75,229	67,213	0,897	41,049
	DVM	89,63	87,705	91,216	88,43	87,705	0,9415	56,241
	KNN	78,889	63,934	91,216	73,239	63,934	0,7758	56,330
	KA	71,185	65,574	77,027	67,797	65,574	0,7198	42,535
	RO	88,889	80,328	95,946	86,726	80,328	0,9516	51,164
XCEPTION	NB	83,704	83,607	83,784	82,258	83,607	0,8946	339,08
	DVM	89,63	86,885	91,892	88,333	86,885	0,9425	416,96
	KNN	87,037	85,246	88,514	85,597	85,246	0,8688	337,14
	KA	76,296	77,049	75,676	74,603	77,049	0,8001	356,01
	RO	88,889	91,803	86,486	88,189	91,803	0,9396	373,29
GOOGLENET	NB	61,852	77,869	48,649	64,847	77,869	0,5994	81,303
	DVM	81,111	81,148	81,081	79,518	81,148	0,8405	86,217
	KNN	79,63	72,951	85,135	76,395	72,951	0,7904	83,822
	KA	71,852	72,131	71,622	69,841	72,131	0,728	232,09
	RO	84,074	80,328	87,162	82,009	80,328	0,898	98,620

Şekil 38’de ResNet-50 modelinin eğitim ve test süreci gösterilmektedir. Şekilde görüldüğü gibi maksimum 6 dönem için eğitim yapılmıştır. Bir dönem, tüm eğitim veri setinde tam bir eğitim döngüsüdür. Tam bağlı katman, softmax katmanı ve sınıflandırma çıktı katmanı değiştirilerek yeni sınıflandırma görevine aktarılmaktadır. Bu katmanlarda öğrenme oranı 20 olarak ayarlanarak son katmanlarda öğrenme hızlandırılmıştır. Transfer katmanlarda başlangıç öğrenme oranı 0,0001 olarak belirlenmiştir. Böylece bu katmanlarda öğrenme yavaşlatılmıştır. Doğrulama frekansı 10 iterasyon olarak ayarlanmıştır.



Şekil 38. ResNet-50 eğitim ve test süreci

Tablo 13’de görüldüğü gibi yalnız derin öğrenme modeli (ResNet-50) kullanarak sınıflandırma yapmak yerine, derin öğrenme modellerini sınıflandırma yöntemleriyle birlikte kullanmak çalışma zamanını azaltmayı sağlamaktadır. ResNet-50 ile sınıflandırma yaparken 7458 sn çalışma zamanı hesaplanırken, ResNet-50 ve DVM yöntemleri birlikte sınıflandırma yaptığında 159,39 sn çalışma zamanı hesaplanmıştır. Ayrıca ResNet-50 ile sınıflandırma yaparken COVID-19 görüntülerini %94,262 oranında doğru tespit ederken, ResNet-50 ve DVM yöntemi birlikte kullanıldığında COVID-19 görüntülerini %95,082 oranında doğru tespit ettiği saptanmıştır.

Tablo 13. ResNet-50 ile Sınıflandırma Deney Sonuçları

Doğruluk (%)	Duyarlılık (%)	Özgüllük (%)	F1-skor (%)	Hassasiyet (%)	ÇZ (sn)
92,59	94,262	91,216	91,999	89,844	7458

SONUÇLAR VE ÖNERİLER

Bu çalışmada toraks BT görüntülerinden olası COVID-19 vakası tespiti için çeşitli sınıflandırma yöntemlerinin performansları derin öğrenme teknikleri kullanılarak değerlendirilmiştir. Elde edilen deneysel sonuçlardan, özgüllük değerlerinin her bir sınıflandırma yöntemindeki duyarlılık değerlerinden daha yüksek olduğunu görülmektedir. Veri kümesindeki daha fazla COVID-19 negatif görüntüsü, modelin COVID-19 negatif sınıf türünü daha iyi öğrenmesini sağlamıştır. Gelecekte yetersiz COVID-19 pozitif görüntü sayısının artırılarak başarı oranının artıracağı düşünülmektedir.

COVID-19 pandemisi yeni bir hastalık sorunu olup, bu konu ile ilgili literatürde kesin ve yeterli veri bulunmamaktadır. Wu *et al.* (2021) ve Saha *et al.* (2021) çalışmalarının dışında literatürdeki tüm çalışmalar bu tez çalışmasındaki kadar COVID-19 hasta görüntüsüne ulaşamamıştır. Saha *et al.* (2021) çalışmalarında kullanılan veri seti açık kaynaklı veri tabanından oluşturulmuştur. Fakat bu çalışmada kullanılan veri seti hastane ortamından elde edilmiş gerçek görüntüler oluşmaktadır. Ayrıca bu çalışmada COVID-19 pozitif olduğu düşünülen görüntüler bir radyoloji uzmanı tarafından doğruluğu teyit edilmiştir. Literatürdeki çalışmaların çoğunda kullanılan yöntemlerin çalışma zamanlarından bahsedilmemiştir. Bu tez çalışmasında doğruluk, duyarlılık, özgüllük, hassasiyet, F1-skor, AUC ve çalışma zamanlarına göre yöntemler değerlendirilmiştir. Literatürdeki çalışmalara (Minaee *et al.* 2020; Apostolopoulos and Mpesiana 2020; Sethy and Behara 2020; Horry *et al.* 2020; Toğaçar *et al.* 2020; Ozcan *et al.* 2020; Narin *et al.* 2020; Ucar and Korkmaz 2020; Rahimzadeh and Attar 2020; Abbas *et al.* 2020; Ozturk *et al.* 2020; Ismael and Şengür 2021) kıyasla bu çalışmada derin öğrenme modelleri ile birlikte çeşitli sınıflandırma yöntemleri kullanarak geniş bir değerlendirme yapılmıştır. Minaee *et al.* (2020) çalışmalarında ResNet18, ResNet50, SqueezeNet ve DenseNet-121 gibi derin öğrenme modelleri incelenmiştir. Apostolopoulos and Mpesiana (2020) yaptıkları çalışmada Vgg19, MobileNetV2, Inception, Xception ve InceptionResNetv2 gibi derin öğrenme modelleri incelenmiştir. Sethy and Behara (2020) çalışmalarında AlexNet, DenseNet201, GoogleNet, Inceptionv3, ResNet18, ResNet50, ResNet101, VGG16, VGG19, XceptionNet, Inceptionresnetv2 gibi modellerle derin özellikler çıkarılmış ve DVM yönteminde kullanılarak veriler sınıflandırılmıştır. Horry *et al.* (2020) yaptıkları çalışmada Vgg-16, Vgg-19, ResNet50, Inceptionv3 ve Xception gibi derin öğrenme modelleri kıyaslanmıştır. Ozcan *et al.* (2020) çalışmalarında ResNet-18, GoogleNet ve ResNet50 modelleri incelenmiştir. Toğaçar *et al.* (2020) yaptıkları çalışmada MobileNetV2 ve

SqueezeNet derin öğrenme modelleri ile derin özellikler çıkarılmış ve DVM yönteminde kullanılarak derin öğrenme modellerinin performansları kıyaslanmıştır. Narin *et al.* (2020) çalışmalarında ResNet50, ResNet101, ResNet152, InceptionV3 ve Inception-ResNetV2 gibi derin öğrenme modelleri incelenmiştir. Ucar and Korkmaz (2020) bu çalışmada Bayes ve SqueezeNet ikilisi kullanılmıştır. Rahimzadeh and Attar (2020) çalışmalarında Birleştirilmiş ESA yöntemi uygulanmıştır. Ismael and Şengür (2021) yaptıkları bu çalışmada Derin özellikler çıkarmak için ResNet18, ResNet50, ResNet101, VGG16 ve VGG19 gibi modeller kullanmıştır. Bu derin özellikler, DVM yönteminde performansları değerlendirilmiştir. Abbas *et al.* (2020) çalışmalarında DeTrac, Resnet18 modellerini incelerken, Ozturk *et al.* (2020) çalışmalarında DarkNet modeli değerlendirilmiştir. Bu tez çalışmasında ResNet-50, ResNet-101, AlexNet, SqueezeNet, Xception, Vgg-19 ve Vgg-16 derin öğrenme modelleri ile derin özellikler çıkarılmıştır. Elde edilen derin özellikler DVM, RO, kNN, NB ve KA sınıflandırma yöntemlerinde kullanılarak performansları değerlendirilmiştir. Literatürdeki çalışmalarda bu kadar geniş bir inceleme yapılmamıştır. Bu tez çalışmasında önerilen modelin BT görüntüleri kullanan son çalışmalarla karşılaştırmalı performans değerlendirmesi Tablo 14’de verilmiştir. COVID-19 vakalarının doğru tespit edilmesi daha önemli olduğu için Tablo 14’de duyarlılık değerleri dikkate alınmıştır. Bu çalışmada önerilen modelden elde edilen duyarlılık (%95,082), BT görüntülerine dayalı literatürdeki çalışmalardaki tüm modellerden daha iyi performans göstermiştir.

Tablo 14. Önerilen Modelin BT Görüntüleri Kullanan Çalışmalarla Karşılaştırılması

	Yöntem	Görüntü Sayısı	Veri Kaynağı	Performans
Wu <i>et al.</i> (2020)	COVNet	4356(1296 covid, 1735 pnömoni, 1325 normal)	Çoklu hastane ortamı	%90 duyarlılık
Zheng <i>et al.</i> (2020)	DeCOVNet	603	3 farklı hastane ortamı	%90,7 duyarlılık
Singh <i>et al.</i> (2020)	MODE-ESA	150(75 covid, 75 normal)	Halka açık veri tabanı	%90,7 duyarlılık
Wu <i>et al.</i> (2021)	COVIDAL	962(304 covid, 316 pnömoni, 342 normal)	Halka açık veri tabanı	%86,6 doğruluk
Bu çalışma	ResNet50-DVM	1345(607 covid, 738 normal)	Hastane ortamı	%95,082 duyarlılık

Makine öğrenimi odaklı akıllı algoritmalar, dünya çapında kliniklerde yaygın olarak kullanılarak çeşitli modellerin geliştirilmesine katkıda bulunurken, COVID-19 salgınına da etkili yanıtlar ve geliştirmeler sağlayabilir. Derin öğrenme teknikleri ile COVID-19 vakalarında yanlış tanının, tanı süresinin ve tetkik araçlarının (COVID-19 test kitleri) gereksiz kullanımının azalmasına yardımcı olacaktır. Bu çalışmada incelenen ve performansı yüksek bulunan derin öğrenme modeli ve sınıflandırma metodu COVID-19 hastalığı için gereksiz testleri önleyerek yardımcı karar destek sistemi olarak kullanılabilir. Geliştirilen sistemin amacı bir uzmanın yerini almak değil, sadece teşhis aşamasında konu ile ilgili çalışan uzmanlardan elde edilen bilgiler ışığında alınacak kararın desteklenmesine yardımcı olmaktır.



KAYNAKLAR

- Abbas, A., Abdelsamea, M. M. and Gaber, M. M., 2020. Classification of COVID-19 in chest X-ray images using DeTraC deep convolutional neural network. *Applied Intelligence*, 1-11.
- Anonymous 2021a <https://www.worldometers.info/coronavirus/> (10.04.2021).
- Anonymous 2021b <https://teknoloji.org/derin-ogrenme-nedir-yapay-sinir-aglari-ne-ise-yarar/> (28.03.2021).
- Anonymous 2021c https://en.wikipedia.org/wiki/Nervous_system (28.03.2021).
- Anonymous 2021d <https://machinelearningknowledge.ai/keras-normalization-layers-explained-for-beginners-batch-normalization-vs-layer-normalization/> (28.03.2021).
- Anonymous 2021e <https://tr.wikipedia.org/wiki/COVID-19> (09.06.2021)
- Anonymous 2021f <https://www.mathworks.com/products/deep-learning.html> (01.04.2021).
- Anonymous 2021g <https://www.mathworks.com/help/deeplearning/ref/squeezenet.html> (02.04.2021)
- Anonymous 2021h <https://www.mathworks.com/help/deeplearning/ref/xception.html> (02.04.2021)
- Anonymous 2021i <http://www.datascience lovers.com/machine-learning/k-nearest-neighbors-knn-theory/> (02.04.2021)
- Anonymous 2021j <https://www.coursehero.com/file/40397178/Support-Vector-Machine-Algorithmpptx/> (02.04.2021)
- Anonymous 2021k <https://www.kdnuggets.com/2020/01/decision-tree-algorithm-explained.html> (02.04.2021)
- Anonymous 2021l <https://medium.com/@Synced/how-random-forest-algorithm-works-in-machine-learning-3c0fe15b6674> (02.04.2021)
- Anonymous 2021m <https://www.geeksforgeeks.org/naive-bayes-classifiers/> (02.04.2021)
- Apostolopoulos, I. D. and Mpesiana, T. A., 2020. COVID-19: automatic detection from x-ray images utilizing transfer learning with convolutional neural networks. *Physical and Engineering Sciences in Medicine*, 43(2), 635-640.
- Askar, H., Krois, J., Rohrer, C., Mertens, S., Elhennawy, K., Ottolenghi, L. and Schwendicke, F., 2021. Detecting white spot lesions on dental photography using deep learning: A pilot study. *Journal of Dentistry*, 107, 103615.
- Benco, M., Kamencay, P., Hudec, R., Radilova, M. and Sykora, P., 2020. Image Texture Analysis based on Color Level Methods using Neural Network. In 2020 ELEKTRO, IEEE, 1-5.
- Bengio, Y., Courville, A. and Vincent, P., 2013. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8), 1798-1828.
- Bhattacharya, S., Maddikunta, P. K. R., Pham, Q. V., Gadekallu, T. R., Chowdhary, C. L., Alazab, M. and Piran, M. J., 2021. Deep learning and medical image processing for coronavirus (COVID-19) pandemic: A survey. *Sustainable cities and society*, 65, 102589.

- Bozkurt, F., 2021. Derin Öğrenme Tekniklerini Kullanarak Akciğer X-Ray Görüntülerinden COVID-19 Tespiti. *Avrupa Bilim ve Teknoloji Dergisi*, 24, pp. 149-156.
- Bozkurt, F., Köse C. and Sarı, A., 2018. An inverse approach for automatic segmentation of carotid and vertebral arteries in CTA. *Expert Systems with Applications*, 93, pp. 358-375.
- Bramer, M., 2007. *Max Principles of Data Mining*. Springer, 526, London.
- Breiman, L. and Cutler, A., 2004. Random Forests, URL: http://www.stat.berkeley.edu/users/breiman.RandomForests/cc_papers.htm.
- Breiman, L., 2001. Random forests. *Machine learning*, 45(1), 5-32.
- Byun, H. and Lee, S. W., 2003. A survey on pattern recognition applications of support vector machines. *International Journal of Pattern Recognition and Artificial Intelligence*, 17(03), 459-486.
- Chollet, F., 2017. Xception: Deep learning with depthwise separable convolutions. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1251-1258.
- Chonacky, N. and Winch, D., 2005. Reviews of Maple, Mathematica, and Matlab: Coming Soon to a Publication Near You. *Computing in Science and Engineering*, 7(2), 9-10.
- Çınar, A. And Yıldırım, M., 2020. Detection of tumors on brain MRI images using the hybrid convolutional neural network architecture. *Medical hypotheses*, 139, 109684.
- Ciresan, D. C., Meier, U., Gambardella, L. M., & Schmidhuber, J., 2011. Convolutional neural network committees for handwritten character classification. 2011 International Conference on Document Analysis and Recognition, IEEE, Beijing, China.
- Cortes, C. and Vapnik, V., 1995. Support-vector networks. *Machine learning*, 20(3), 273-297.
- Dağ, Z. T., 2012. Avuç içi terlemesi ile kandaki glikoz ve HbA1C parametreleri arasındaki ilişkinin yapay sinir ağları kullanılarak belirlenmesi. *Doktora Tezi, Fen Bilimleri Enstitüsü, Sakarya Üniversitesi*
- Das, N. N., Kumar, N., Kaur, M., Kumar, V. and Singh, D., 2020. Automated deep transfer learning-based approach for detection of COVID-19 infection in chest X-rays. *Irbm*.
- Du, X., Cai, Y., Wang, S. and Zhang, L., 2016. Overview of deep learning. In 2016 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC), IEEE, 159-164.
- Ezzat, D., Hassanien, A. E. and Ella, H. A., 2020. An optimized deep learning architecture for the diagnosis of COVID-19 disease based on gravitational search optimization. *Applied Soft Computing*, 106742.
- Fang, Y., Zhang, H., Xie, J., Lin, M., Ying, L., Pang, P. and Ji, W., 2020. Sensitivity of chest CT for COVID-19: comparison to RT-PCR. *Radiology*, 296(2), E115-E117.
- Garg, M. and Dhiman, G., 2020. A novel content-based image retrieval approach for classification using GLCM features and texture fused LBP variants. *Neural Computing and Applications*, 1-18.
- Gulli, A. and Pal, S., 2017. *Deep learning with Keras*. Packt Publishing Ltd, 318.
- Gupta, A., Gupta, S. and Katarya, R., 2021. InstaCovNet-19: A deep learning classification model for the detection of COVID-19 patients using Chest X-ray. *Applied Soft Computing*, 99, 106859.

- Haenlein, M. and Kaplan, A., 2019. A brief history of artificial intelligence: On the past, present, and future of artificial intelligence. *California management review*, 61(4), 5-14.
- Haigh, T., 2008. Cleve Moler: Mathematical software pioneer and creator of Matlab. *IEEE Annals of the History of Computing*, 30(1), 87-91.
- He, K., Zhang, X., Ren, S. and Sun, J., 2016. Deep residual learning for image recognition. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770-778.
- Hemdan, E. E. D., Shouman, M. A. and 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*.
- Hinton, G. Bengio, Y. and LeCun, Y., 2015. Deep learning. *nature*, 521(7553), 436-444.
- Hinton, G. E. and Salakhutdinov, R. R., 2006. Reducing the dimensionality of data with neural networks. *science*, 313(5786), 504-507.
- Horry, M. J., Paul, M., Ulhaq, A., Pradhan, B., Saha, M. and Shukla, N., 2020. X-Ray image based COVID-19 detection using pre-trained deep learning models.
- Iandola, F. N., Han, S., Moskewicz, M. W., Ashraf, K., Dally, W. J. and Keutzer, K., 2016. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and < 0.5 MB model size. *arXiv preprint arXiv:1602.07360*.
- Islam, M. M., Karray, F., Alhajj, R. and Zeng, J., 2021. A review on deep learning techniques for the diagnosis of novel coronavirus (covid-19). *IEEE Access*, 9, 30551-30572.
- Ismael, A. M. and Şengür, A., 2021. Deep learning approaches for COVID-19 detection based on chest X-ray images. *Expert Systems with Applications*, 164, 114054.
- Jahanbani Fard, M., 2015. Bayesian Approach for Early Stage Event Prediction in Survival Data. MS Thesis, Wayne State University, Michigan, ABD.
- Kang, Z., Li, X. and Zhou, S., 2020. Recommendation of low-dose CT in the detection and management of COVID-2019.
- Karakanis, S. and Leontidis, G., 2021. Lightweight deep learning models for detecting COVID-19 from chest X-ray images. *Computers in Biology and Medicine*, 130, 104181.
- Kim, P. 2017. Matlab deep learning: With machine learning, neural networks and artificial intelligence. Apress, 151p, Seoul, Korea.
- Krizhevsky, A., Sutskever, I. and Hinton, G.E., "Imagenet classification with deep convolutional neural net works," *Adv. Neural Inf. Process. Syst.*, pp. 1097–1105, 2012.
- Kutsurelis, J. E., 1998. Forecasting financial markets using neural networks: An analysis of methods and accuracy. Naval Postgraduate School Monterey CA.
- Lin, M., Chen, Q. and Yan, S., 2013. Network in network. *arXiv preprint arXiv:1312.4400*.
- Lubaib, P. and Muneer, K. A., 2016. The heart defect analysis based on PCG signals using pattern recognition techniques. *Procedia Technology*, 24, 1024-1031.
- M. Yağanoğlu and C. Köse "Real-Time Detection of Important Sounds with a Wearable Vibration Based Device for Hearing-Impaired People," *Electronics*, 7(4), 50, 2018.
- Mahase, E., 2020. Coronavirus: global stocks of protective gear are depleted, with demand at 100 times normal level. British Medical Journal Publishing Group, WHO warns.
- Mahmud, T., Rahman, M. A. and Fattah, S. A., 2020. CovXNet: A multi-dilation convolutional neural network for automatic COVID-19 and other pneumonia detection

- from chest X-ray images with transferable multi-receptive feature optimization. *Computers in biology and medicine*, 122, 103869.
- Minaee, S., Kafieh, R., Sonka, M., Yazdani, S. and Soufi, G. J., 2020. Deep-COVID: Predicting COVID-19 from chest x-ray images using deep transfer learning. *Medical image analysis*, 65, 101794.
- Murty, M. N. and Devi, V. S., 2011. *Pattern recognition: An algorithmic approach*. Springer Science and Business Media, 196, Newyork.
- Narin, A., Kaya, C. and Pamuk, Z., 2020. Automatic detection of coronavirus disease (COVID-19) using x-ray images and deep convolutional neural networks. *arXiv preprint arXiv:2003.10849*.
- Nayak, J., Naik, B., Dinesh, P., Vakula, K., Dash, P. B. and Pelusi, D., 2021. Significance of deep learning for Covid-19: state-of-the-art review. *Research on Biomedical Engineering*, 1-24.
- Nisbet, R., Elder, J. and Miner, G., 2009. *Handbook of statistical analysis and data mining applications*. Academic Press, 824, USA.
- Oh, Y., Park, S. and Ye, J. C., 2020. Deep learning COVID-19 features on cxr using limited training data sets. *IEEE Transactions on Medical Imaging*, 39(8), 2688-2700.
- Ouchicha, C., Ammor, O. and Meknassi, M., 2020. CVDNet: A novel deep learning architecture for detection of coronavirus (COVID-19) from chest x-ray images. *Chaos, Solitons and Fractals*, 140, 110245.
- Ozturk, T., Talo, M., Yildirim, E. A., Baloglu, U. B., Yildirim, O. and Acharya, U. R., 2020. Automated detection of COVID-19 cases using deep neural networks with X-ray images. *Computers in biology and medicine*, 121, 103792.
- Padmanaban, H., 2014. Comparative analysis of Naive Bayes and tree augmented naïve Bayes models. MS Thesis, San Jose State University, California, ABD.
- Panahi, A. H., Rafiei, A. and Rezaee, A., 2021. FCOD: Fast COVID-19 Detector based on deep learning techniques. *Informatics in medicine unlocked*, 22, 100506.
- Pandit, M. K., Banday, S. A., Naaz, R. and Chishti, M. A., 2020. Automatic detection of COVID-19 from chest radiographs using deep learning. *Radiography*.
- Panwar, H., Gupta, P. K., Siddiqui, M. K., Morales-Menendez, R. and Singh, V., 2020b. Application of deep learning for fast detection of COVID-19 in X-Rays using nCOVnet. *Chaos, Solitons and Fractals*, 138, 109944.
- Panwar, H., Gupta, P. K., Siddiqui, M. K., Morales-Menendez, R., Bhardwaj, P. and Singh, V., 2020a. A deep learning and grad-CAM based color visualization approach for fast detection of COVID-19 cases using chest X-ray and CT-Scan images. *Chaos, Solitons & Fractals*, 140, 110190.
- Rahimzadeh, M. and Attar, A., 2020. A modified deep convolutional neural network for detecting COVID-19 and pneumonia from chest X-ray images based on the concatenation of Xception and ResNet50V2. *Informatics in Medicine Unlocked*, 19, 100360.
- Saha, P., Sadi, M. S. and Islam, M. M., 2021. EMCNet: Automated COVID-19 diagnosis from X-ray images using convolutional neural network and ensemble of machine learning classifiers. *Informatics in medicine unlocked*, 22, 100505.
- Samuel, A. L., 1959. Some studies in machine learning using the game of checkers. *IBM Journal of research and development*, 3(3), 210-229.

- Schmidhuber, J., 2015. Deep learning in neural networks: An overview. *Neural networks*, 61, 85-117.
- Sethy, P. K., Behera, S. K., Ratha, P. K. and Biswas, P., 2020. Detection of coronavirus disease (COVID-19) based on deep features and support vector machine. *Preprints*.
- Shukla, P. K., Shukla, P. K., Sharma, P., Rawat, P., Samar, J., Moriwal, R. and Kaur, M., 2020. Efficient prediction of drug–drug interaction using deep learning models. *IET Systems Biology*, 14(4), 211-216.
- Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G. and Hassabis, D., 2016. Mastering the game of Go with deep neural networks and tree search. *nature*, 529(7587), 484-489.
- Simard, P. Y., Steinkraus, D. and Platt, J. C., 2003. Best practices for convolutional neural networks applied to visual document analysis. In *Icdar*, 3.
- Simonyan, K. and Zisserman, A., 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.
- Singh, D., Kumar, V. and Kaur, M., 2020. Classification of COVID-19 patients from chest CT images using multi-objective differential evolution–based convolutional neural networks. *European Journal of Clinical Microbiology & Infectious Diseases*, 39, 7, pp. 1379-1389.
- Staub, S., Karaman, E., Kaya, S., Karapınar, H. and Güven, E., 2015. Artificial neural network and agility. *Procedia-Social and Behavioral Sciences*, 195, 1477-1485.
- Sujath, R., Chatterjee, J. M. and Hassanien, A. E., 2020. A machine learning forecasting model for COVID-19 pandemic in India. *Stochastic, Environmental Research and Risk Assessment*, 34, 959-972.
- Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D. and Rabinovich, A., 2015. Going deeper with convolutions. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1-9.
- Taşkın, A. G. D. Ç. and Emel, G. G., 2010. Veri Madenciliğinde Kümeleme Yaklaşımları Ve Kohonen Ağları İle Perakendecilik Sektöründe Bir Uygulama. *Süleyman Demirel Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi*, 15(3), 395-409.
- Toğaçar, M., Ergen, B. and Cömert, Z., 2020. COVID-19 detection using deep learning models to exploit Social Mimic Optimization and structured chest X-ray images using fuzzy color and stacking approaches. *Computers in biology and medicine*, 121, 103805.
- Tsiknakis, N., Trivizakis, E., Vassalou, E. E., Papadakis, G. Z., Spandidos, D. A., Tsatsakis, A. and Marias, K., 2020. Interpretable artificial intelligence framework for COVID-19 screening on chest X-rays. *Experimental and Therapeutic Medicine*, 20(2), 727-735.
- Turing, A. M., 1950. The word problem in semi-groups with cancellation. *Annals of Mathematics*, 491-505.
- Ucar, F. and Korkmaz, D., 2020. COVIDiagnosis-Net: Deep Bayes-SqueezeNet based diagnosis of the coronavirus disease 2019 (COVID-19) from X-ray images. *Medical Hypotheses*, 140, 109761.
- Wu, X., Chen, C., Zhong, M., Wang, J. and Shi, J., 2021. COVID-AL: The diagnosis of COVID-19 with deep active learning. *Medical Image Analysis*, 68, 101913.
- Wu, X., Hui, H., Niu, M., Li, L., Wang, L., He, B. and Zha, Y., 2020. Deep learning-based multi-view fusion model for screening 2019 novel coronavirus pneumonia: a multicentre study. *European Journal of Radiology*, 128, 109041.

- Zhao, H., Lai, Z., Leung, H. and Zhang, X., 2020. Neural-Network-Based Feature Learning: Convolutional Neural Network. In Feature Learning and Understanding, Springer, Cham, 219-251.
- Zheng, C., Deng, X., Fu, Q., Zhou, Q., Feng, J., Ma, H. and Wang, X., 2020. Deep learning-based detection for COVID-19 from chest CT using weak label. MedRxiv.



ÖZGEÇMİŞ

Kişisel Bilgiler	
Adı Soyadı:	Çinare OĞUZ
Doğum tarihi:	
Doğum Yeri:	
Uyruğu:	
Adres:	
Tel:	
E-mail:	
Eğitim	
Lise:	Erzurum Anadolu Lisesi
Lisans:	Çanakkale On Sekiz Mart Üniversitesi, Mühendislik Fakültesi
Yüksek lisans:	Atatürk Üniversitesi, Fen Bilimleri Ens Bilgisayar Mühendisliği Anabilim Dalı
Yabancı Dil Bilgisi	
İngilizce:	İyi
Almanca:	Başlangıç
Rusça:	Orta
Üye Olunan Mesleki Kuruluşlar	
Tezden Üretilmiş Yayınlar	
1. Determination of Covid-19 Possible Cases by Using Deep Learning Techniques. <i>Sakarya University Journal of Science</i> , 25(1), 1-11.	