



**T.C.
GAZİ ÜNİVERSİTESİ
BİLİŞİM ENSTİTÜSÜ**

**YÜKSEK
LİSANS
TEZİ**

**X-IŞINI VE DERİN ÖĞRENME İLE
COVID-19 TESPİTİ**

NAMRİĞ HUSSAIN SULIMAN FEDAL

BİLGİSAYAR BİLİMLERİ ANA BİLİM DALI

TEMMUZ 2021



X-IŞINI VE DERİN ÖĞRENME İLE COVID-19 TESPİTİ

Namrig HUSSAIN SULIMAN FEDAL

YÜKSEK LİSANS TEZİ
BİLGİSAYAR BİLİMLERİ ANA BİLİM DALI

GAZİ ÜNİVERSİTESİ
BİLİŞİM ENSTİTÜSÜ

TEMMUZ 2021

Namrig HUSSAIN SULIMAN FEDAL tarafından hazırlanan “X-IŞINI VE DERİN ÖĞRENME İLE COVID-19 TESPİTİ ” adlı tez çalışması aşağıdaki jüri tarafından OY BİRLİĞİ ile Gazi Üniversitesi Bilgisayar Bilimleri Ana Bilim Dalında YÜKSEK LİSANS TEZİ olarak kabul edilmiştir.

Danışman: Prof.Dr. Mahir DURSUN

Elektrik - Elektronik Mühendisliği, Gazi Üniversitesi

Bu tezin, kapsam ve kalite olarak Yüksek Lisans Tezi olduğunu onaylıyorum

Başkan: Prof.Dr.M.Ali AKÇAYOL

Bilgisayar Bilimleri, Gazi Üniversitesi

Bu tezin, kapsam ve kalite olarak Yüksek Lisans Tezi olduğunu onaylıyorum

Üye: DOÇ. DR. Mehmet SERDAR GÜZEL

Bilgisayar Bilimleri, Ankara Üniversitesi

Bu tezin, kapsam ve kalite olarak Yüksek Lisans Tezi olduğunu onaylıyorum

Tez Savunma Tarihi: 09/07/2021

Jüri tarafından kabul edilen bu tezin Yüksek Lisans Tezi olması için gerekli şartları yerine getirdiğini onaylıyorum.

.....
Prof. Dr. Aslıhan TÜFEKÇİ
Bilişim Enstitüsü Müdürü

ETİK BEYAN

Gazi Üniversitesi Bilişim Enstitüsü Tez Yazım Kurallarına uygun olarak hazırladığım bu tez çalışmada;

- Tez içinde sunduğum verileri, bilgileri ve dokümanları akademik ve etik kurallar çerçevesinde elde ettiğimi,
- Tüm bilgi, belge, değerlendirme ve sonuçları bilimsel etik ve ahlak kurallarına uygun olarak sunduğumu,
- Tez çalışmada yararlandığım eserlerin tümüne uygun atıfta bulunarak kaynak gösterdiğimi,
- Kullanılan verilerde herhangi bir değişiklik yapmadığımı,
- Bu tezde sunduğum çalışmanın özgün olduğunu,

bildirir, aksi bir durumda aleyhime doğabilecek tüm hak kayıplarını kabullendiğimi beyan ederim.

İmza

Namrig HUSSAİN SULİMAN FEDAL

09/07/2021

X-IŞINI VE DERİN ÖĞRENME İLE COVID-19 TESPİTİ

(Yüksek Lisans Tezi)

Namrig HUSSAİN SULİMAN FEDAL

GAZİ ÜNİVERSİTESİ
BİLİŞİM ENSTİTÜSÜ

Temmuz 2021

ÖZET

Çin'de, 2019 yılının sonlarında ortaya çıkan koronavirüs, dünya geneline yayıldı ve büyük bir salgın haline geldi. Yüksek yayılma oranı nedeniyle dünyanın küresel sağlık sistemlerini ve insanların başta sağlık, ekonomi, eğitim ve spor olmak üzere hemen hemen bütün günlük yaşamlarını etkilemiştir. Bu nedenle COVID-19'un hızlı ve zamanında teşhisi, pandeminin yayılma hızını ve bulaş riskini azaltmada çok önemli bir faktördür. Göğüs röntgen filmleri, son pandemide, COVID-19 hastalığının test edilmesi ve teşhisinde önemli bir rol oynamaktadır. Bu röntgen filmlerinin teşhiste, Derin Öğrenme gibi gelişmiş yapay zeka teknikleri ile kullanılması doktorların gözünden kaçabilecek küçük ayrıntıların fark edilmesi veya insan gözü ile fark edilemeyecek hastalıkların da teşhisinde önemli bir fayda sağlayacaktır. Bu çalışmada, göğüs röntgeni radyografileri kullanılarak Covid-19 virüsü enfekte hastaların tespit edebilmek amacıyla paralel bir Evrişimli Sinir Ağları (Convolutional Neural Networks [CNN]'ler) modeli önerilmiştir. Bu araştırma ayrıca, torasik röntgenlerden Covid-19'un otomatik teşhisinde, bilim insanları tarafından önerilen son teknoloji ürünü CNN'nin etkinliğinin nasıl değerlendirileceğini de sunmaktadır. Sonuçları doğrulamak için 500 Covid-19 Pozitif görüntü ve 1600 Covid-19 Negatif görüntü kullanılmış olup, çözümlemelerimiz test kurulumunda % 90 doğruluk ve % 88 sınıflandırma doğruluğu sağlamıştır.

Bilim Kodu : 92432

Anahtar Kelimeler : Covid 19, X-Ray görüntüsü, derin öğrenme, CNN

Sayfa Adedi : 67

Danışman : Prof.Dr. Mahir DURSUN

COVID-19 DETECTION BY X-RAY IMAGES AND DEEP LEARNING

(M. Sc. Thesis)

Namrig HUSSAIN SULIMAN FEDAL

GAZI UNIVERSITY

INFORMATICS INSTITUTE

July 2021

ABSTRACT

The coronavirus, which appeared in China in 2019, has become an epidemic. Global health has affected the economy and the daily lives of individuals. Therefore, it is important in this case that the diagnosis of this disease is made immediately and that all efforts cooperate to provide a fast and accurate method at the same time to limit the exacerbation and spread of the epidemic. X-rays are important in these cases of the covid disease. In our time, artificial intelligence provides important and very useful techniques for solving complex and large problems that need accuracy that a person may not be able to provide it. In our study, we have proposed new parallel convolutional neural networks to be able to detect patients with this disease and also provide a method for determining whether our network is working efficiently or not. We used about 500 pictures as a positive covid entry .1,600 as a negative covid entry, most of which were pictures of pneumonia. our solution achieved 90% accuracy and 88% classification accuracy in the test setup.

Science Code : 92432

Key Words : Covid-19, X-Ray image, deep learning, CNN

Page Number : 67

Supervisor : Prof. Dr. Mahir DURSUN

TEŞEKKÜR

Bu tez çalışması için azami seviyede çaba gösterdim. Ancak, danışmanım Prof. Dr. Mahir Durusun'un nazik desteği ve yardımı olmadan bu çalışmayı gerçekleştirmem asla mümkün olmazdı. Bana sunduğu bu destekler ve yardımları için en içten teşekkürlerimi, şükranlarımı ve saygılarımı sunuyorum. Bana, tez konusunu belirlememde yardım ve tavsiyelerde bulunan, Sayın Prof. Dr. Mahir DURSUN, tezimin ışığı oldu. Belki gerçeği tam ifade edemiyorum. Çünkü doğru danışmana, nazik bir Hoca'ya ve dürüst bir kardeşe sahip olduğumu söylemek istiyorum. Bilgisini bana açtı, bana güven verdi ve bana kararlılığın gücünü aşıladı. Ayrıca hiçbir çabadan ve yardımdan kaçınmadı. Rehberliği ve sürekli gözetimi ile tez hakkında gerekli bilgileri sağlamanın yanı sıra, tezimi tamamlama konusundaki desteği için de çok şey borçluyum. Bu tez çalışmasının tamamlanmasında bana yardımcı olan ve teşviklerini esirgemeyen aileme ve Gazi Üniversitesi üyelerine de ayrıca minnettarlığımı ifade etmek istiyorum. Bana bu kadar özen gösterdikleri ve zaman ayırdıkları için de ayrıca çok teşekkür ediyorum.

İÇİNDEKİLER

	Sayfa
ÖZET.....	iv
ABSTRACT.....	v
TEŞEKKÜR.....	vi
İÇİNDEKİLER.....	vii
ÇİZELGELERİN LİSTESİ.....	ix
ŞEKİLLERİN LİSTESİ.....	x
SİMGELER VE KISALTMALAR.....	xii
1. GİRİŞ.....	1
2. İLGİLİ ARAŞTIRMALAR.....	11
3. YAPAY ZEKA VE MAKİNE ÖĞRENMESİ.....	27
3.1. Denetimli ve Denetimsiz Öğrenme.....	27
3.2. Yapay Sinir Ağları (Neural Networks).....	27
3.3. Derin Öğrenme.....	28
3.3.1. Derin öğrenmenin çalışması.....	29
3.3.2. Derin öğrenme algoritmaları.....	30
4. MALZEMELER VE YÖNTEMLER.....	35
4.1 Veri Kümesinin Toplanması.....	35
4.2 Görüntü Ön İşleme.....	36
4.3. Evrişimli Sinir Ağının (CNN) Eğitimi.....	37
4.3.1.Paralel evrişimli sinir ağı.....	38
4.3.2. Paralel evrişimli sinir ağının mimarisi.....	38
4.3.3.Evrişimli katmanlar.....	39
5. SONUÇLAR.....	45

Sayfa

5.1. Değerlendirme.....	47
5.2. Web Uygulaması.....	50
5.3. Tartışma.....	52
5.4. Sonuç ve Öneriler.....	52
KAYNAKLAR.....	55
EKLER.....	57
EK-1. Veri Kümesinin Toplanması Kodu.....	58
EK-2. Görüntü Ön İşleme Kodu.....	60
EK-3. Evrişimli Sinir Ağının (CNN) Eğitimi kodu.....	62
EK-4. Modeli web'e bağlamak için kodumuz.....	64
EK-5. HTML ve JavaScript ile web sitesinin ön yüzü.....	66
ÖZGEÇMİŞ.....	67

ÇİZELGELERİN LİSTESİ

Çizelge	Sayfa
Çizelge 1.1. Coronavirus detay	9
Çizelge 1.2. Mers, Sars ve Covid-19 karşılaştırması	9
Çizelge 5.1. Paralel CNN'nin sınıflandırma performansı	49
Çizelge 5.2. Paralel CNN'nin karışıklık matrisi.....	49



ŞEKİLLERİN LİSTESİ

Şekil	Sayfa
Şekil 1.1. COVID-19 başlatma vakalarının dünya çapında dağılımı.....	2
Şekil 1.2. Hastalık grip, virüs, medikal.....	3
Şekil 1.3. Virüslerin çeşitleri	4
Şekil 1.4. Covid-19 virüsü	5
Şekil 1.5. Koronavirüs belirtileri	6
Şekil 1.6. Hızlı seroloji antikor testi	6
Şekil 1.7. ELISA testi	7
Şekil 1.8. Teşhis amaçlı Rt-PCR sürüntü testi.....	7
Şekil 1.9. Aşı çeşitleri.....	9
Şekil 1.10. Covid-19 olgularında antiviral tedavi algoritması.....	10
Şekil 2.1. Çalışmanın önerilen metodolojisine genel bakış	16
Şekil 2.2. Çalışmanın önerilen metodolojisine genel bakış	18
Şekil 2.3. Çalışmanın Göğüs röntgeni görüntüsünü belirlemeye iş akışı	20
Şekil 2.4. Çalışmanın kullanılan MobileNetV2 modelinin genel tasarımı	24
Şekil 2.5. Çalışmanın[13] kullanılan SqueezeNet şablonunun genel tasarımı	25
Şekil 3.1. Sinir ağları türü	28
Şekil 3.2. Derin öğrenme mimarisi	29
Şekil 3.3. Makine öğrenimi ve derin öğrenme arasındaki fark.....	29
Şekil 3.4. Sınıflandırılmış evrişimli sinir ağı modeli.....	30
Şekil 3.5. Tekrarlayan sinir ağı açıklaması	31
Şekil 3.6. Derin inanç ağı açıklaması.....	32
Şekil 3.7. Derin inanç ağı modeli.....	32
Şekil 3.8. Otomatik kodlayıcı açıklaması	33

Şekil	Sayfa
Şekil 4.1. (a)Sağlıklı birey, (b) COVID-19 pozitif hasta veri seti, (c) pnömoni hastası..	36
Şekil 4.2. Evrişimli sinir ağı mimarisi	39
Şekil 4.3. CNN katman.larımızın çıktı şekli	42
Şekil 4.4. X-Ray görüntülerinde COVID-19 durumunu sınıflandırmak için iş akışı	43
Şekil 5.1. Sinir ağı eğitiminin geçmişi	46
Şekil 5.2. Eğitim kaybı ve doğrulama kaybı	47
Şekil 5.3. Eğitim doğruluğu ve doğrulama doğruluğu	48
Şekil 5.4. Covid-19'u tespit etmek için grafiksel kullanıcı arayüzü tabanlı bir araç.....	50
Şekil 5.5. Anaconda komut isteminde web uygulamasının yürütülmesi	51

SİMGELER VE KISALTMALAR

Bu çalışmada kullanılmış bazı simgeler ve kısaltmalar, açıklamaları ile birlikte aşağıda sunulmuştur.

Simgeler	Açıklama
α	Alpha
β	Beta
Σ	Sigma
Kısaltmalar	Açıklama
2D	İki boyutlu(Two dimensional)
AI	Yapay zeka(Artificial intelligence)
CNN	Evrışimli sinir ağı(Convolutional neural network)
CPU	Merkezi işlem birimi(Central processing unit)
DBN	Derin inanç ağı(Deep belief network)
FC	Tamamen bağlı(Fully connected)
GPU	Grafik işleme ünitesi(Graphics processing unit)
LSTM	Uzun kısa süreli bellek(Long short-term memory)
MXP	Metropolitan değişim noktası (Metropolitan exchange point)
RBM	Kısıtlı boltzmann makinesi (Restricted boltzmann machine)
RNN	Tekrarlayan bir sinir ağı (Recurrent neural network)
SIANN	Uzayda değişmeyen yapay sinir ağları (Space invariant artificial neural networks)

1. GİRİŞ

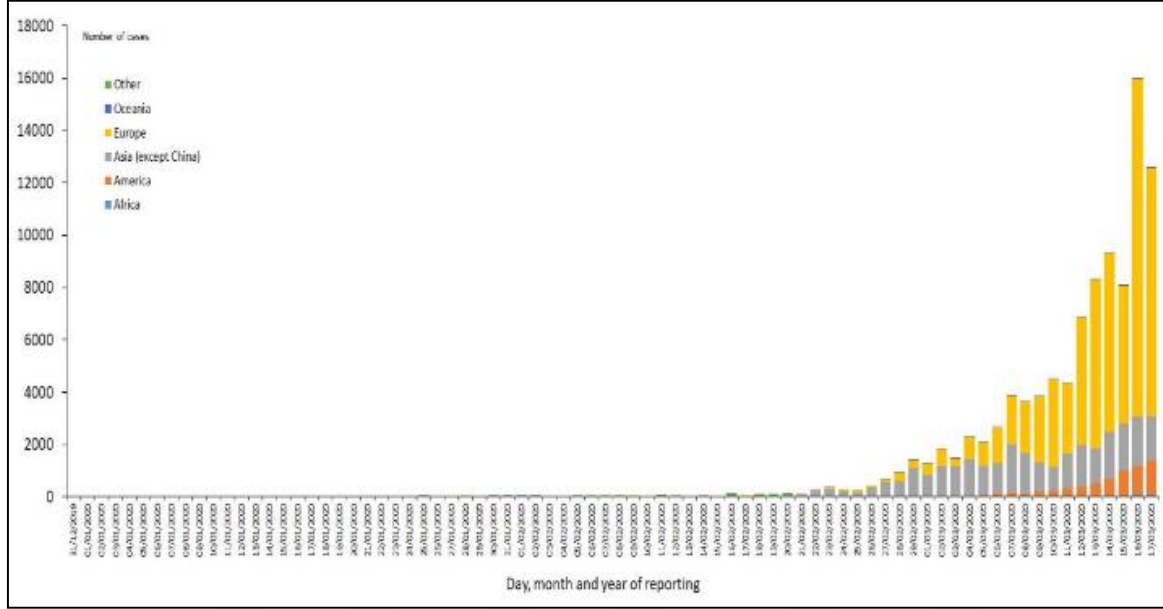
COVID-19 olarak adlandırılan yeni Corona virus, dünya genelinde çok hızlı bir şekilde yayılmış, SARS-CoV-2 virüsü gibi, COVID-19 hastalığını da dünya çapında bir salgın haline getirmiştir. Salgın başta insan sağlığı olmak üzere ekonomi, eğitim, spor ve bireylerin günlük yaşamlarını ciddi oranda etkilemiştir. Bu nedenle COVID-19'un hızlı ve zamanında teşhisi, pandeminin yayılma hızını ve bulaş riskini azaltmada çok önemli bir faktördür. Röntgen filmlerinin derin öğrenme gibi gelişmiş yapay zeka teknikleri ile teşhiste kullanılması, doktorların gözünden kaçabilecek küçük ayrıntıların fark edilmesi veya insan gözü ile fark edilemeyecek hastalıkların da teşhisinde önemli faydalar sağlamaktadır. Bu nedenle göğüs röntgen filmleri, yaşanan son pandemide COVID-19 hastalığının test edilmesi ve teşhisinde önemli bir rol oynamaktadır. Bu virüs, soğuk algınlığı gibi diğer virüs türlerinde ortak olan bir grup semptom olan koku ve tat kaybı, öksürük ve ateş [1] gibi çoğunlukla solunum sistemi ile ilgili ve nefes darlığı içeren semptomlarla karakterizedir. Diğer virüslerin aksine, COVID-19, ortalama olarak 3 gün ile 13 gün arasında değişen uzun bir kuluçka dönemine sahiptir ve virüse maruz kaldıktan sonra semptomların başlamasına kadar yaklaşık beş ila altı günlük bir süre geçmektedir.

Uzun kuluçka dönemi, virüsü taşıyan insanların virüsü taşıdıklarını fark edinceye kadar rutin günlük yaşamlarına devam edeceklerinden, diğer insanlarla etkileşime gireceklerdir. Bunun sonucu olarak, daha fazla enfeksiyona yol açan bu durum COVID-19' u daha bulaşıcı hale getirmektedir. 2020 Yılı'nın başından beri virüsü taşıyan hastaları tanımlamak için bir metodoloji geliştirmeye yönelik birkaç çalışma yapılmıştır.

Ciddi hastalık başlangıcı, büyük alveolar hasar ve ilerleyici solunum yetmezliği nedeniyle ölümle sonuçlanabilmektedir. Covid-19'un erken ve otomatik teşhisi, hastaların zamanında karantinaya sevk edilmesi ve özel hastanelerde şiddetli vakaların hızlı entübasyonu ve hastalığın yayılmasının izlenmesi ülkeler açısından faydalı olabilir.

Teşhis nispeten hızlı bir süreç haline gelmesine rağmen, teşhis testlerinin maliyetinden kaynaklanan finansal sorunlar hem devletleri hem de hastaları zor duruma düşürmektedir. Özellikle yasaklayıcı düzenlemeler ve fiyatlar nedeniyle sağlık sistemlerine sınırlı erişime sahip ülkeler veya özel sağlık sistemleri sınırlı olan ülkelerde [2] finansal sorunları da beraberinde getirmektedir. Şekil 1.1'de 17 Mart 2020 tarihi itibarıyla COVID-19 vakalarının

dünya çapında dağılımı verilmiştir. Tablo incelendiğinde COVID-19'un özel olarak bu tarihten sonra tüm dünyada logaritmik olarak artış göstermiş olduğu ve hastalığın yayılımı noktasında kontrolün kaybedilmiş olduğu gözükmemektedir. 17 Mart 2020 itibariyle COVID-19 vakalarının dünya çapında dağılımı Şekil 1.1'de görülmektedir.

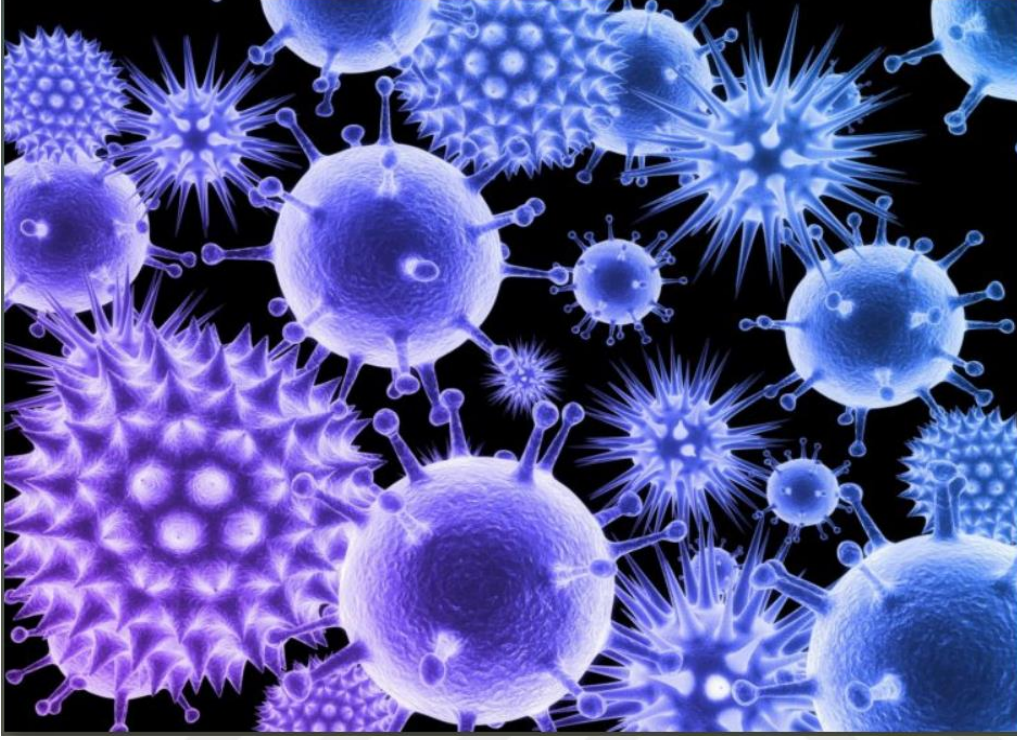


Şekil 1.1. COVID-19 başlatma vakalarının dünya çapında dağılımı

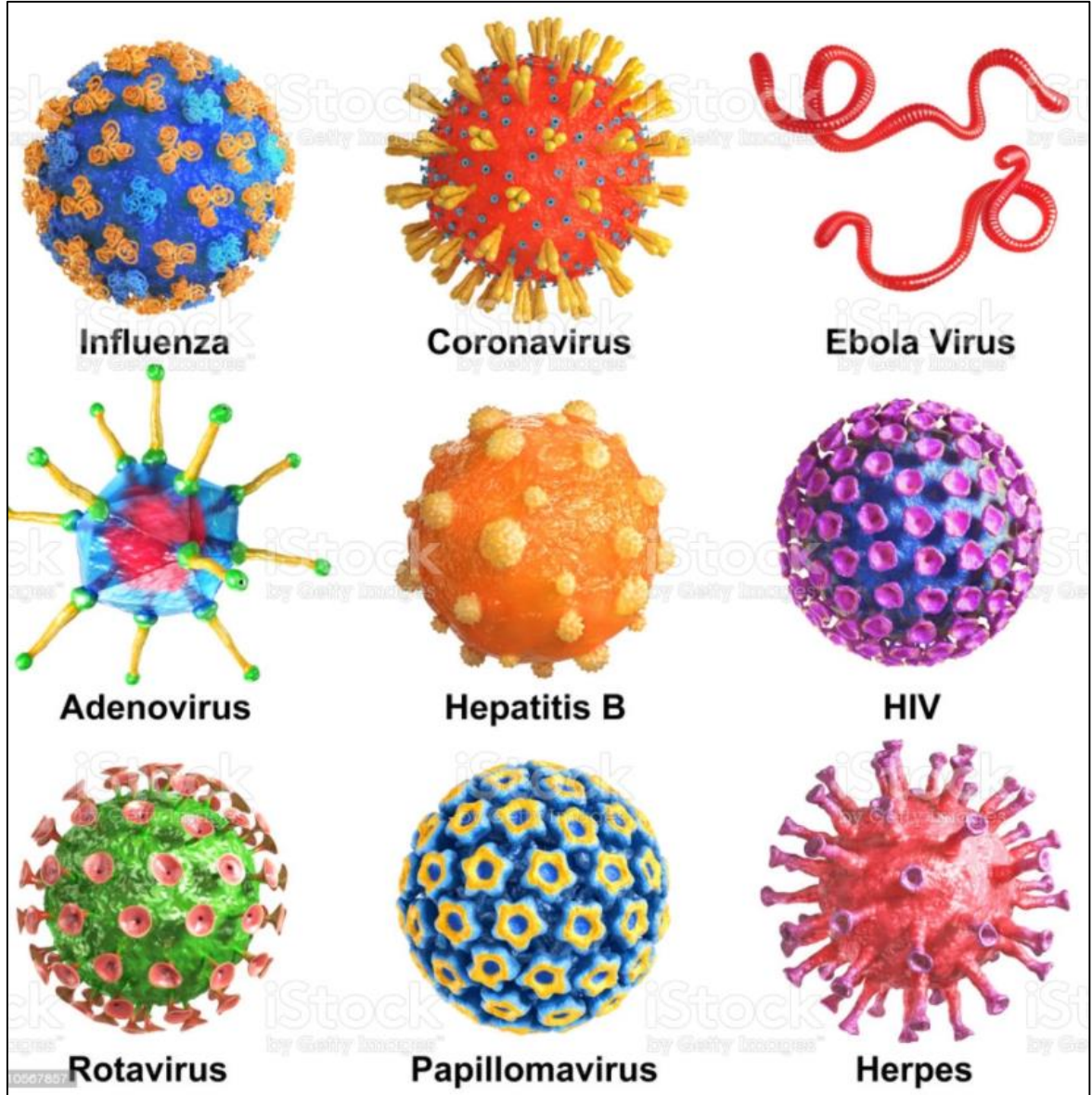
Virüs Nedir?

Virüsler, var olan en küçük parazit türü olup, her ne kadar bazı virüsler 1 μm büyüklükte olsada, virüslerin boyutları genellikle 0,02 ila 0,3 μm arasında değişmektedir. Viral partikül veya virion, bir protein kaplaması ve bazen viral replikasyonu başlatmak için gerekli olan enzimlerle çevrili tek bir nükleik asit (RNA veya DNA) çekirdeği içerir. Virüsler yalnızca hayvanların, bitkilerin ve bakterilerin hücrelerinde çoğalabilir ve bu nedenle zorunlu hücre içi parazitler olarak adlandırılır. Virüsler neden oldukları hastalıklara göre sınıflandırılmaz; bunun yerine nükleik asidin tek veya çift sarmallı olmasına, bir viral zarfın mevcut olup olmamasına ve bunların replikasyon tarzına göre, farklı aileler halinde gruplandırılırlar. Tek sarmallı RNA virüsleri ayrıca pozitif veya negatif anlamlı RNA'ya sahip olup olmadıklarına göre sınıflandırılır. DNA virüsleri, konakçı hücrelerin çekirdeği içinde çoğalma eğilimindeyken, RNA virüsleri genellikle bunu sitoplazmada yapar. Tarihsel olarak virüslerin enfekte olduğu hayvanların, bitkilerin ve insanların yok olmasına neden olan az sayıda parazit ile karşılaşmıştır. Çocuk felci ve çiçek hastalığı gibi hastalıkların tümü,

insanlar ve hayvanlar üzerindeki yaygın, yıkıcı etkisiyle bilinir. Hakkında daha az bilinen şey ise, çok sayıda virüsün bitkiler için de verebileceği ürün zararları problemleridir [3].



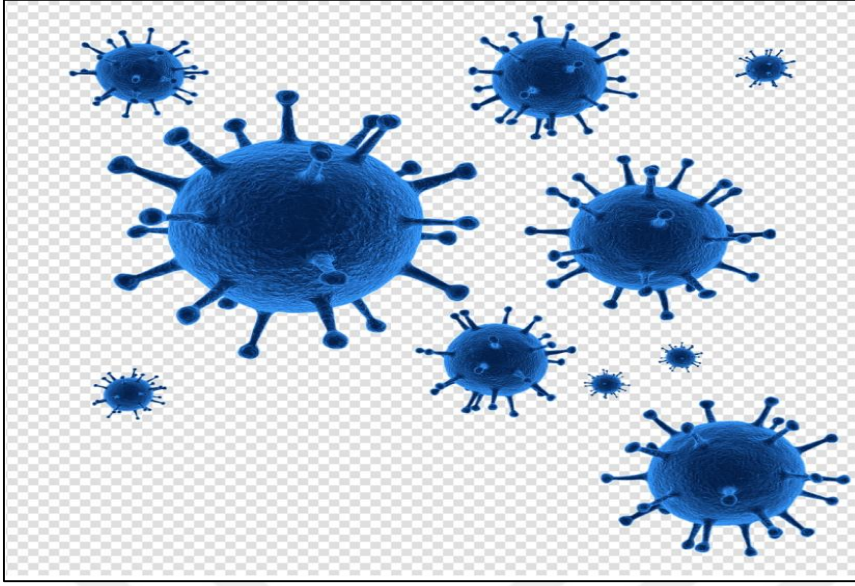
Şekil 1.2. Hastalık grip, virüs, medikal



Şekil 1.3. Virüslerin çeşitleri

COVID-19 nedir?

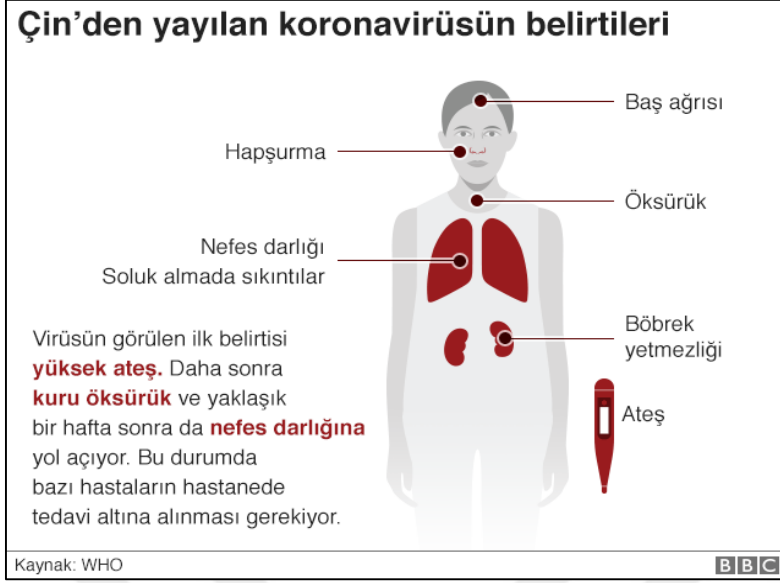
Korona virüs, nükleokapsid, zar, zarf adı verilen bir yağ tabakası ve bu virüsü diğer virüslerden ayıran tek şey, kendisine dışarıdan bağlanan protein çıkıntıları içermesidir.



Şekil 1.4. Covid-19 virüsü

Koronavirüs belirtileri

Bu tehlikeli virüsü karakterize eden, en önemli semptomlardan biri, yaklaşık 39 dereceye kadar ulaşabilen yüksek ateştir. Bununla birlikte virüse enfekte olan kişi ayrıca baş ağrısı ve boğaz ağrısı çekebiliyorken, bazı hastalar ise burun akıntısı ve bel ağrısından muzdariptir. Ayrıca kişi, kendini halsiz, yorgun ve uzun süre hareket edemeyecek şekilde hisseder. En zor semptomlardan biri nefes darlığıdır ve bu durum, virüs kişinin bağışıklık sistemine saldırdığında ortaya çıkar. Bu belirtiyeye en çok maruz kalanlar, 65 yaşın üzerindeki, kanser, nefes darlığı, diyabet ve diğerleri gibi kronik hastalıkları bulunan yaşlı insanlardır. Bu hastalığın genç grup üzerindeki etkisi, özellikle güçlü bağışıklığa sahip olanlar için hafiftir.

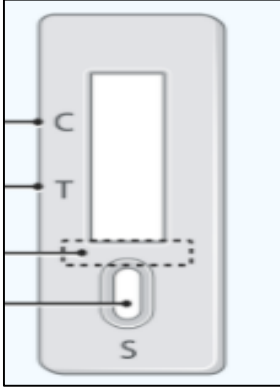


Şekil 1.5. Koronavirüs belirtileri

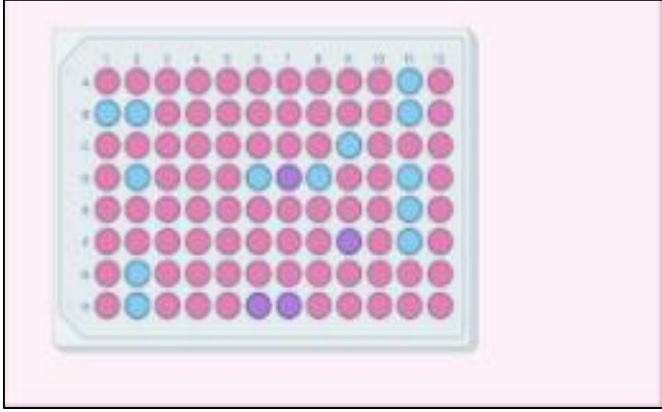
Covid-19 test türü

Covid testi üç tür içerir:

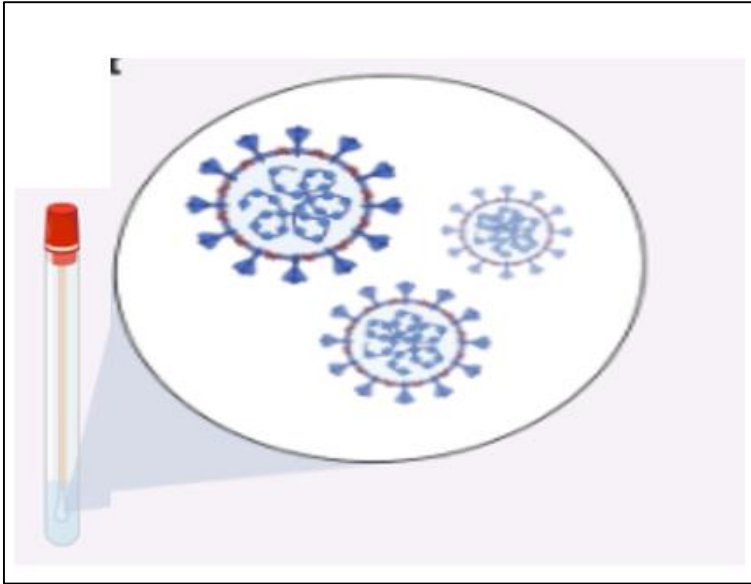
- Hızlı seroloji Antikor testi
- ELISA Testi
- Teşhis amaçlı Rt-PCR sürüntü testi



Şekil 1.6. Hızlı seroloji antikor testi



Şekil 1.7. ELISA testi



Şekil 1.8. Teşhis amaçlı Rt-PCR sürüntü testi

COVID-19 aşısı çeşitleri

Bir aşı tasarımının, Tam Mikrop Yaklaşımı, Alt Birim Yaklaşımı ve Genetik Yaklaşım olmak üzere, üç ana yaklaşımı vardır. Bu yaklaşımların farklılıkları, bütün bir virüs veya bakteri kullanıp kullanmamaları, sadece mikropların bağışıklık sistemini etkileyen kısımlarına yönelik ya da tüm virüsü değil sadece belirli proteinleri yapmak için talimatlar veren genetik materyale yönelik olmalarından kaynaklanır.

Tam Mikrop Yaklaşımı

Bu aşılardan tümü vücudun bağışıklık sistemine COVID-19'a neden olan virüsü güvenli bir şekilde tanıtmayı ve yok etmeyi öğretecek şekilde tasarlanmıştır.

Etkisizleştirilmiş aşı

Aşı yapmanın ilk yolu, hastalık taşıyan virüs veya bakterileri almaktır. Hastalığa neden olmamakla birlikte bağışıklık yanıtı oluşturan etkisizleştirilmiş virüs içeren ve ısı ya da radyasyon kullanılarak virüsün etkisiz hale getirildiği aşılarıdır.

Zayıflatılmış aşı

Canlı zayıflatılmış bir aşı, virüsün yaşayan ancak zayıflatılmış bir versiyonunu veya virüsün çok benzerinin kullanıldığı yaklaşımdır.

Viral vektör aşısı

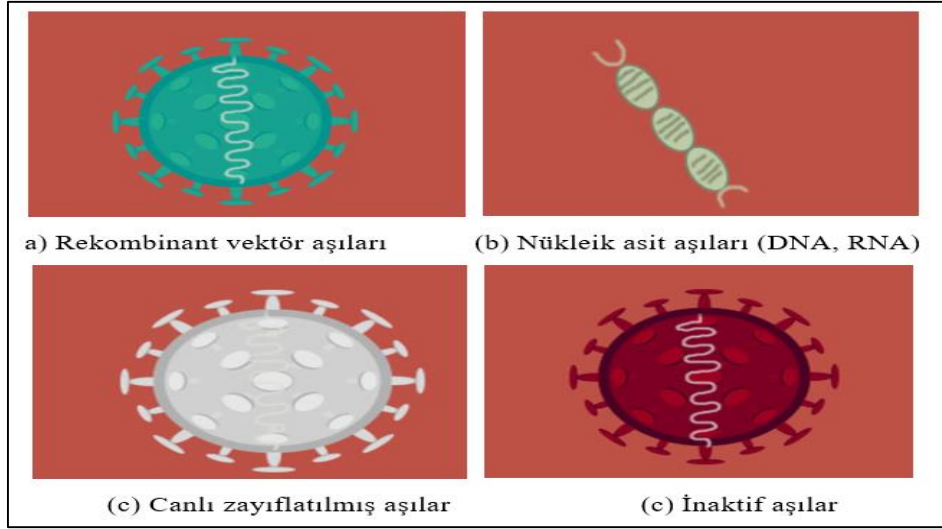
Bu aşı, protein sentez mekanizmalarının belirli alt kısımlarına bağışıklık sistemlerinin harekete geçmesini sağlamak için verilen güvenli (hastalık yapmayan) virüslerin kullanıldığı aşı türüdür.

Alt birim yaklaşımı

Alt birim aşı, bağışıklık sisteminin tanınması gereken bir virüs veya bakterinin yalnızca çok özel kısımlarını (alt birimlerini) kullanan bir aşıdır.

Genetik Yaklaşım (Nükleik Asit Aşısı)

Zayıflatılmış veya ölmüş bir mikrobu veya bir parçasını kullanan aşı yaklaşımlarının aksine, bir nükleik asit aşısı, yalnızca belirli proteinler için talimatlar sağlayan bir genetik materyal bölümü kullanır. Bu aşı yaklaşımında hücrelerin protein yapmaları için DNA ve RNA parçacıkları kullanılmaktadır.



Şekil 1.9. Aşı çeşitleri

Çizelge 1.1. Coronavirus detay

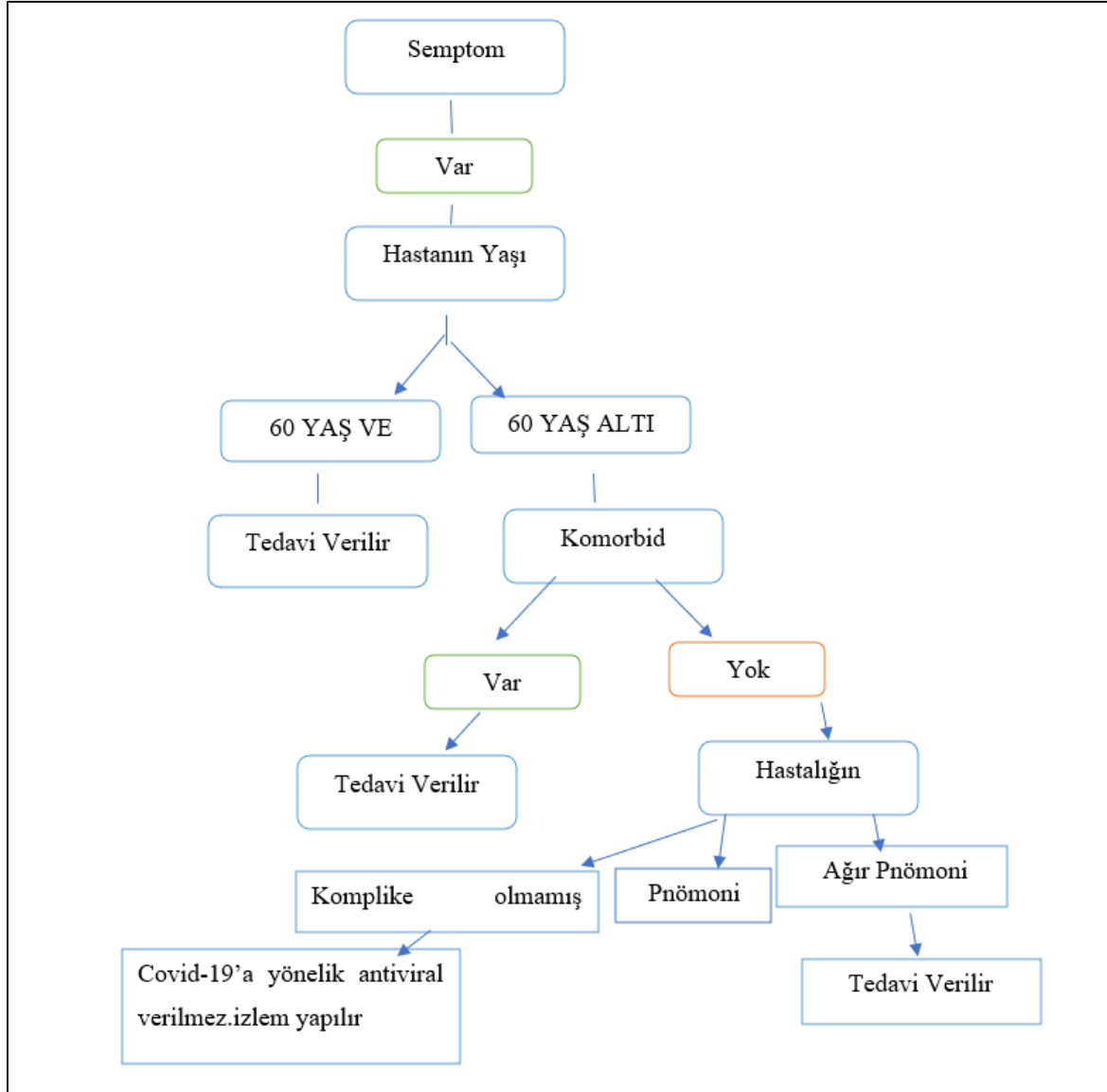
CoV	Yıl	Menşei	Ölüm Oranı (%)
SARS	2002	Guangdong rovince, China	10
MERS	2013	Saudi Arabia	34
COVID-19	2019	Wuhan, China	3.4

Mers, Sars ve Covid-19 klinik özelliklerinin karşılaştırılması

SARS, MERS ve COVID-19 ise koronavirüs adı verilen büyük bir virüs ailesinin neden olduğu hayvansal kökenli solunum yolu hastalıklarıdır.

Çizelge 1.2. Mers, Sars ve Covid-19 karşılaştırması

Karşılaştırma	SARS	Mers	Covid-19
Semptomlar	Ateş Baş ağrısı, Kuru öksürük, Nefes darlığı	Ateş Öksürük	Ateş Baş ağrısı, Kuru öksürük,
Kuluçka dönemi	2-10 gün	14 gün	2-14 gün
Kişiden başka kişiye aktarma	Evet	Evet	Evet
Ölüm	Evet	Evet	Evet



Şekil 1.10. Covid-19 olgularında antiviral tedavi algoritması

2. İLGİLİ ARAŞTIRMALAR

Asif ve diğerleri [4], dijital göğüs röntgen görüntülerini kullanarak, COVID-19 pnömoni hastalarını otomatik olarak tespit etmeyi amaçlamıştır. Derin Evrişimli Sinir Ağlarını (DCNN) kullanarak, algılamadaki doğruluğu en üst düzeye çıkarırken, çalışmanın veri seti 864 COVID-19, 1345 viral pnömoni ve 1341 normal göğüs röntgeni görüntüsünden oluşmuştur. Bu çalışmada, korona virüs pnömonisini tespit etmeyi öğrenen DCNN tabanlı bir Inception V3 modeli önerilmiştir.

DCNN genellikle, daha büyük bir veri kümesi ile daha iyi performans gösterir. Transfer Öğrenme Yöntemi, ImageNet gibi büyük veri setlerinden eğitilmiş (sınıflandırılmış büyük veri setlerinden amaca uygun olanların seçilmesi) modeli kullanır. Bu, büyük veri kümelerine sahip olma ihtiyacını ortadan kaldırır ve aynı zamanda derinlemesine analiz gerektiren uzun araştırma süresini kısaltır.

COVID-19 hastalarının göğüs röntgeni görüntüleri, Dr. Joseph Cohen tarafından paylaşılan açık kaynak GitHub deposundan elde edilmiştir. Veri ön işleme aşamaları sırasında, algoritma için görüntü girişi farklı olduğundan X-Ray görüntüleri işlenebilirliğini artırmak ve ayrıca Inception V3'e uygun hale getirmek için tüm resimleri 299x299x3 olarak yeniden boyutlandırılmıştır. Eğitimde yetersiz veriyi artırmak için veri büyütme tekniklerini kullanarak, görüntü büyütme ve kullanılan teknikler arasında dikey çevirme, yataya çevirme, görüntü çevirme, bulanıklaştırma ve görüntüyü 60°, 90°, 180°, 270° döndürme yer alır.

Evrişim, havuzlama, softmax ve tamamen bağlantılı prosedürler gerçekleştiren Inception V3 modelinin mimarisi iki parçadan oluşur:

- Evrişimli Sinir Ağı'nın çıkarılması.
- Tamamen bağlı ve Softmax katmanları kullanma.

TensorFlow, Google tarafından oluşturulmuş derin bir öğrenme çerçevesidir. Sistemdeki tüm nöronları (düğümleri) kontrol edebilen ve % 98'in üzerinde bir sınıflandırma doğruluğu olarak en iyi performansın elde edildiği görüntü işlemeye uygun bir kütüphane imkanı sunmaktadır. 4000 yinelemeden sonra eğitim ilerledikçe doğruluğu artarak son test

doğruluğu % 96 olarak belirlenmiştir. Elde edilen sonuçlar, transfer öğrenmenin etkili olduğunu kanıtlar nitelikte olup, COVID-19 tespiti için güçlü performans ve kolayca uygulanabilir bir yaklaşım olduğunu göstermektedir.

Inception V3 modeli, nispeten daha düşük bir görüntü koleksiyonundan kendisini etkili bir şekilde eğiterek COVID-19 pnömonisini sınıflandırmada mükemmel bir performans sergiler.

Bu bilgisayar destekli teşhis aracının COVID-19 ile vakaları teşhis etme hızını ve doğruluğunu önemli ölçüde artırabileceğine inanılmaktadır. Bu durum, bir pandemide oldukça faydalı olabilir.

Erdaş ve Ölçer [5] göğüs röntgenlerinin incelendiği çalışmalarında, derin öğrenme yaklaşımının, geleneksel tanı araçlarına alternatif ve destekleyici olarak kullanmışlardır. Evrişimli Sinir Ağları (CNNs) mimarisine sahip olan bu derin öğrenme yaklaşımının amacı, (1) bir koronavirüsün neden olduğu pnömoniyi teşhis etmek, (2) akciğer grafisinde pnömoni semptomları olan bir hastanın bakteri veya koronavirüs kaynaklı olup olmadığını bulmaktır.

X-ışını görüntüleri, Cohen veri seti, İtalyan Tıbbi ve Girişimsel Radyoloji Derneği (SIRM) web sitesi, Radyopedi ve Kuzey Amerika Radyoloji Derneği (RSNA) gibi farklı kaynaklardan elde edilen verileri yeni bir veri seti olarak birleştirilerek elde edilmiştir. X-ışını görüntüleri 224x224 olarak yeniden ölçeklendirilmiştir. Konvansiyonel sinir ağlarının aksine, CNN birçok bilgisayarla görme alanında, özellikle sınıflandırmada, evrişim, havuzlama ve tam bağlantılı katmanlar içermektedir. Evrişim sol üstten başlar ve belirli bir genişlik ve yükseklikte küçük bir filtre penceresi alır ve üzerinde işlem yapar. Yapılan işlem genellikle en iyi nihai sonucu elde etmek için gradyan inişiyle karar verilen matris çarpımıdır. Bu süreç, görüntünün tamamı boyunca sırayla devam eder ve yeni bir görüntü oluşturur.

Literatürde, güvenilirliği çeşitli çalışmalarla kanıtlanmış bir CNN türü olan AlexNet mimarisinden faydalanılarak sağlanmaya çalışılmıştır. AlexNet, bugüne kadarki en popüler sinir ağı mimarilerinden biridir. AlexNet mimarisi, ilk beşi evrişimli katmanlar ve softmax katmanı tam olarak bağlanmadan önceki son iki katman olmak üzere toplamda sekiz katmandan oluşur.

İlk iki evrişimli katman, Yerel Yanıt Normalleştirme Yi (LRN) yaklaşımları kullanarak çok çeşitli özellikleri çıkarmak için örtüşen maksimum havuzlama (MXP) katmanlarına bağlanır. Corona virüs temelli pnömoni teşhis görevlerinde (1) % 92 doğruluk ve pnömoninin kökenini bulma görevinde % 81 oranında başarılı oldu (2). Ayrıca, ROC Eğrisi Altındaki Alan (ROC_AUC), Kesinlik, Geri Çağırma, F1-skoru, Özgüllük ve Negatif Öngörücü Değer (NPV) ölçümlerindeki başarılar bu makalede belirtilmiştir.

Narin, Kaya ve Pamuk [6], çalışmalarında göğüs filmleri kullanarak koronavirüs pnömonisi ile enfekte hastanın tespiti için önceden eğitilmiş beş evrişimli sinir ağı tabanlı model (ResNet50, ResNet101, ResNet152, InceptionV3 ve Inception-ResNetV2) önermiştir. X-ışını radyografileri, 5 kat çapraz doğrulama kullanarak dört sınıfla (COVID-19, normal (sağlıklı), viral pnömoni ve bakteriyel pnömoni) üç farklı ikili sınıflandırma uygulanmıştır. CNN, bu işlemleri etkin bir şekilde gerçekleştirmek için bir evrişimli katman, havuzlama katmanı ve tam bağlantılı katman olmak üzere üç farklı katmandan oluşur. Önceden eğitilmiş modelin büyük bir veri seti üzerinde elde ettiği bilgiler, eğitilecek modele aktarılır. Ayrıca yetersiz veri ve eğitim süresinin üstesinden gelmek için ImageNet verileri kullanılarak gerçekleştirilen aktarım öğrenme tekniği uygulanmıştır.

Kalıbın özelliklerini belirlemekten sorumlu olan Evrişimli Katman, CNN'nin temel katmanıdır. Bu katmanda, giriş görüntüsü bir filtreden geçirilir. Adım parametresi, giriş matrisi üzerinden geçiş yapmak için ayarlanmış adım sayısıdır. Evrişimli katmanın çıktısı Eş. 2.1' deki gibi verilir:

$$x_j^l = f \left(\sum_{a=1}^N w_j^{l-1} * y_a^{l-1} + b_j^l \right) \quad (2.1)$$

burada x_j^l , l katmanındaki j-th özellik haritasıdır, w_j^{l-1} , l-1 katmanındaki j-th çekirdeklerini belirtir. y_a^{l-1} , l-1 katmanındaki a-th özellik haritasını temsil eder, b_j^l , l katmanındaki j-th özellik

haritasının sapmasını gösterir. N, l katmanındaki toplam özelliklerin sayısıdır ve (*), vektör evrişim sürecidir.

Evrişimli katmandan sonraki ikinci katman, havuz (pool) katmanıdır. Bu çalışmada, maksimum paylaşım ve küresel ortalama havuzlama kullanılmıştır. Maksimum havuzlama işlemi, her özellik haritasında belirtilen matris boyutunu kullanarak yalnızca maksimum değeri seçer ve sonuçta çıktı nöronlarının azalmasına neden olur. Bu katmanın temel amacı, ağın aşırı uyumunu ve sapmasını önlemektir.

Tamamen bağlı katman, CNN'nin son ve en önemli katmanıdır. Bu katman, çok katmanlı algılayıcı gibi işlev görür. Aktivasyon fonksiyonları olarak Doğrultulmuş Doğrusal Birim (ReLU) ve Softmax kullanılmıştır, Bu iki aktivasyon fonksiyonunun matematiksel hesaplaması Eş. 2.2 ve Eş 2.3'te gösterilmektedir.

$$ReLU(x) = \begin{cases} 0, & x < 0 \\ x, & x \geq 0 \end{cases} \quad (2.2)$$

$$Soft\ Max(x_i) = \frac{e^{x_i}}{\sum_{x=1}^m e^{x_y}} \quad (2.3)$$

ile yapılmaktadır. Burada x_i ve m sırasıyla girdi verilerini ve sınıf sayısını temsil eder. Tamamen bağlı bir katmandaki nöronlar, önceki katmandaki tüm aktivasyon işlevleriyle tam bağlantılara sahiptir.

Önceden eğitilmiş ResNet50, ResNet101, ResNet152, InceptionV3 dahil olmak üzere geleneksel CNN'nin şematik gösterimi. kullanılan veri kümesi, eğitim ve test için sırasıyla % 80 ve % 20 olmak üzere rastgele iki bağımsız veri kümesine bölünmüştür. Çapraz doğrulama yöntemi olarak k-kat seçilerek 5 farklı k değerine göre sonuçlar elde edilmiştir ($k = 1-5$).

Sonuçlar, ResNet50'nin önceden eğitilmiş modelinin, kullanılan üç farklı veri kümesi için beş model arasında en yüksek doğruluğu sağladığını göstermektedir (Veri Kümesi-1:% 96,1, Veri Kümesi-2:% 99,5 ve Veri Kümesi-3:% 99,7). ResNet 50 ve ResNet 101,% 96.1 ile en yüksek genel performansa sahiptir. Normal verilerin fazlalığının tüm modellerde daha yüksek performansla sonuçlandığı açıktır. Bulgular ışığında, daha yüksek performans nedeniyle radyologların klinik pratikte karar vermelerine yardımcı olacağına inanılmaktadır. COVID-19'u erken bir aşamada tespit etmek için, bu çalışma derin transfer öğrenme yöntemlerinin nasıl kullanılabileceği konusunda fikir vermektedir.

Das ve diğerleri [7], çalışmalarında, göğüs röntgeni görüntülerini kullanarak Covid-19 + ve hastalarını tespit edebilen Derin Evrişimli Sinir Ağı tabanlı bir çözüm önermişlerdir. Çözümün etkinliğini test etmek için Covid + ve Covid - vakalarının herkese açık göğüs röntgen görüntülerini kullanmışlardır. Covid + hastalarının 538 görüntüsü ve Covid - hastalarının 468 görüntüsü, 771 eğitilebilir görüntü ve 235 test görüntüsüne bölünmüştür. Yapılan bu çalışmada önceden eğitilmiş DenseNet201, Resnet50V2 ve Inceptionv3 üç CNN modelinden oluşan bu yeni önerilen birleştirme yöntemi, tahmini daha sağlam hale getireceği belirtilmiştir.

Dense Convolution Network veya DenseNet'in en büyük avantajı, benzer geleneksel CNN türlerinden nispeten daha az parametre gerektirmesidir. DenseNet'i seçmenin ek bir nedeni de, her katmanın önceki tüm katmanların özellik haritalarını girdi olarak almasıdır. Bu, özelliğin yayılmasını güçlendirmeye yardımcı olur ve özelliğin yeniden kullanımını teşvik eder. Bu yöntemde, bir model, örneğin ResNet50v2, diğer iki modelden daha iyi performans gösteriyorsa, yani daha düşük doğrulama hatası veriyorsa, sınıf değerine karar vermedeki katkısı daha yüksek olacak şekilde daha yüksek bir ağırlık atanır.

$$w_i = \frac{A_i}{\sum A_i} \quad (2.4)$$

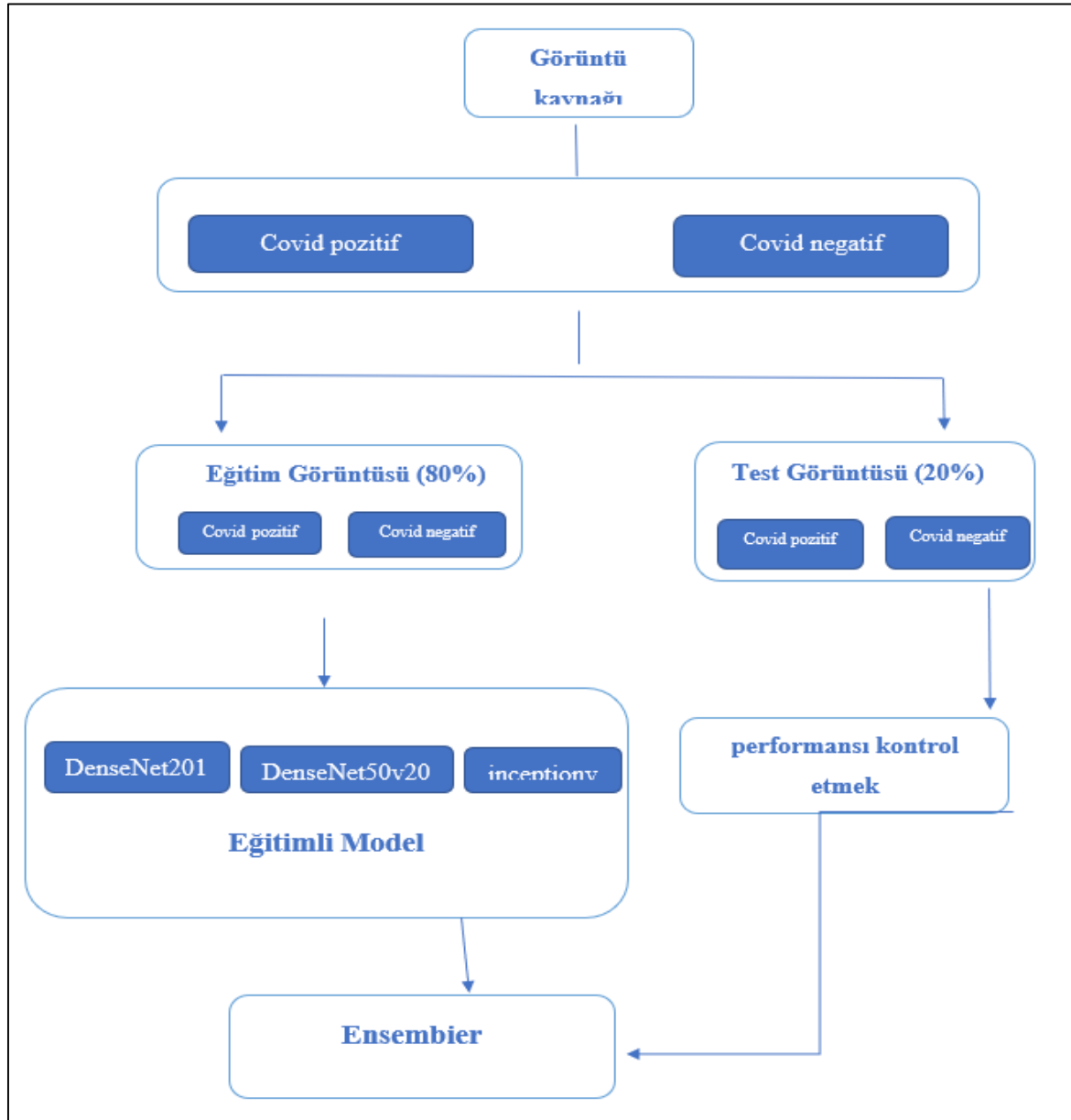
Eş. 2.4'te. Modelin doğruluk yüzdesini A_i olarak varsayarsak, doğrulama hatası $(100 - A_i)$ olur. Birleştirilmiş görüntüler ilk olarak normalleştirilir ve 224×224 şekilli görüntülere yeniden boyutlandırılır. Görüntüler karıştırılır ve eğitim ve test verilerine bölünür; burada test boyutu% 20 ve geri kalanı eğitim amaçlıdır.

$$w_i = \frac{R(A_i)}{\sum R(A_i)} \quad (2.5)$$

Eş. 2.5'te. Yaygın olarak kullanılan üç grup tekniği vardır:

- Doğruluğa dayalı olarak ağırlıkları hesaplayan ağırlıklı ortalama yaklaşım (doğrulukla).

- Dereceli doğruluğa göre ağırlıkları hesaplayan ağırlıklı ortalama yaklaşım .
- Çıktı puanının / olasılığının ağırlıksız ortalamasını alan ağırlıksız ortalama yaklaşımı.



Şekil 2.1. Çalışmanın önerilen metodolojisine genel bakış [4]

Hastalığın yayılmasını önlemek ve kontrol altında tutmak için Covid + hastalarının hızlı ve zamanında tespiti gereklidir. Bu araştırma çalışması, Covid + hastalarını Göğüs Röntgeni

görüntülerinden basit ve ucuz bir şekilde tespit etmek için yapılmıştır. Bu çalışmada, son teknoloji ürünü üç derin öğrenme modeli benimsenmiş ve bir araya getirilmiştir. Önerilen model, % 95,7'lik bir sınıflandırma doğruluğuna ulaşmıştır. Daha da önemlisi, % 98 hassasiyet vermiş olmasıdır ki, bu 100 Covid + hastasının 98'i önerilen modele göre doğru şekilde teşhis edilebilmesini sağlamıştır.

Khan, Shah ve Bhat [8], tarafından yapılan çalışmada, göğüs röntgeni görüntülerinden COVID-19 enfeksiyonunu otomatik olarak tespit etmek için Derin Evrişimli Sinir Ağı modeli olan CoroNet önerilmiştir. Önerilen model, ImageNet veri setinde önceden eğitilmiş Xception mimarisine dayanmaktadır ve halka açık iki farklı veritabanından COVID-19 ve diğer göğüs pnömonisi X-ray görüntülerini toplayarak hazırlanan bir veri setinde uçtan uca eğitilmiştir. Derin öğrenme, tamamen bu öğrenme modellerinde yakıt görevi gören verilerle ilgilidir.

COVID-19 yeni bir hastalık olduğundan, bu çalışma için kullanılabilecek uygun büyüklükte bir veri kümesi mevcut değildir. Bu nedenle, kamuya açık iki farklı görüntü veritabanından göğüs röntgeni görüntüleri toplayarak bir veri kümesi oluşturmak zorunda kalınmıştır. COVID-19 X-ışını görüntüleri Joseph ve diğerleri tarafından açık kaynaklı bir Github deposunda mevcuttur.

COVID-19 ile ilgili yapılan çalışmaların çoğu, Kuzey Amerika Radyoloji Derneği (RSNA) ve Github tarafından sağlanan görüntüleri kullanmaktadır. Yapılan çalışmada bu iki kaynaktan toplam 1300 görüntü toplanmış olup, katmanları aşağıda belirtildiği gibi olan Evrişimli Sinir Ağı modeli kullanılmıştır:

Evrişimli Katman

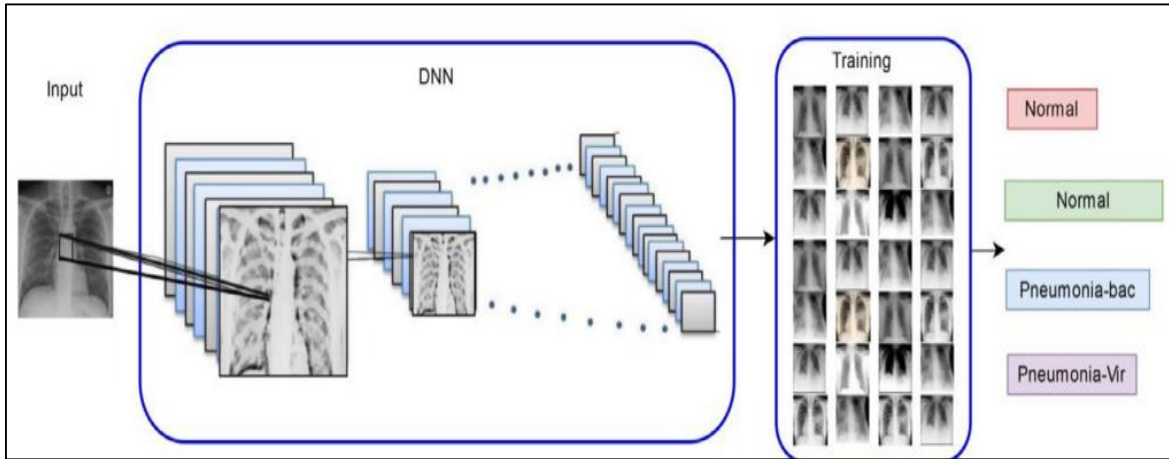
Parametreleri, çekirdek olarak da bilinen bir dizi öğrenilebilir filtreden oluşur. Evrişimli katmanın ana görevi, veri kümesi boyunca ortak olan giriş görüntüsünün yerel bölgelerinde bulunan özellikleri tespit etmek ve görünümlerini bir özellik haritasına eşlemektir.

Alt Örneklem (Havuzlama) Katmanı

CNN'de, evrişim katmanı dizisini, girdinin uzamsal boyutunu küçültmek ve dolayısıyla ağıdaki parametre sayısını azaltmak için isteğe bağlı bir havuzlama veya aşağı örneklem katmanı izler.

Tamamen Bağlı Katman

Tamamen bağlanmış son katmanın çıktısı daha sonra sınıf puanlarını çıkaran bir etkinleştirme işlevine iletilir. Softmax ve Destek Vektör Makineleri (SVM), CNN'de kullanılan iki ana sınıflandırıcıdır. CoroNet, göğüs röntgeni görüntülerinden COVID 19 enfeksiyonunun tespiti için tasarlanmış bir CNN mimarisidir. Xception CNN mimarisine dayanmaktadır. Başlangıcın Uç versiyonu anlamına gelen Xception, ImageNet veri kümesinde önceden eğitilmiş 71 katmanlı derinlikte bir CNN mimarisidir. Xception, klasik evrişimler yerine artık bağlantılı derinlemesine ayrılabilir evrişim katmanları kullanmaktadır. Derinlemesine Ayrılabilir Evrişim, klasik evrişim işlemini noktasal evrişim işlemi ve ardından kanal-bazlı $n \times n$ uzaysal evrişim işlemi ile değiştirir. Bu şekilde işlem sayısı $1/k$ ile orantılı bir faktör kadar azaltılır.



Şekil 2.2. Çalışmanın önerilen metodolojisine genel bakış [5]

COVID-19'u göğüs röntgeni görüntülerinden tespit etmek için önerilen üç model uygulanmıştır. İlk model, göğüs röntgeni görüntülerini dört kategoriye ayırmak üzere eğitilmiş çok sınıflı ana modeldir (4 sınıf CoroNet): COVID-19, Normal, Pnömoni-bakteriyel

ve Pnömoni-viral. Diğer iki model 3 sınıflı CoroNet (COVID-19, Normal ve Pnömoni) ve ikili 2 sınıf CoroNet modeli (COVID-19, Normal ve Pnömoni) çok sınıflı ana modelin modifikasyonlarıdır.

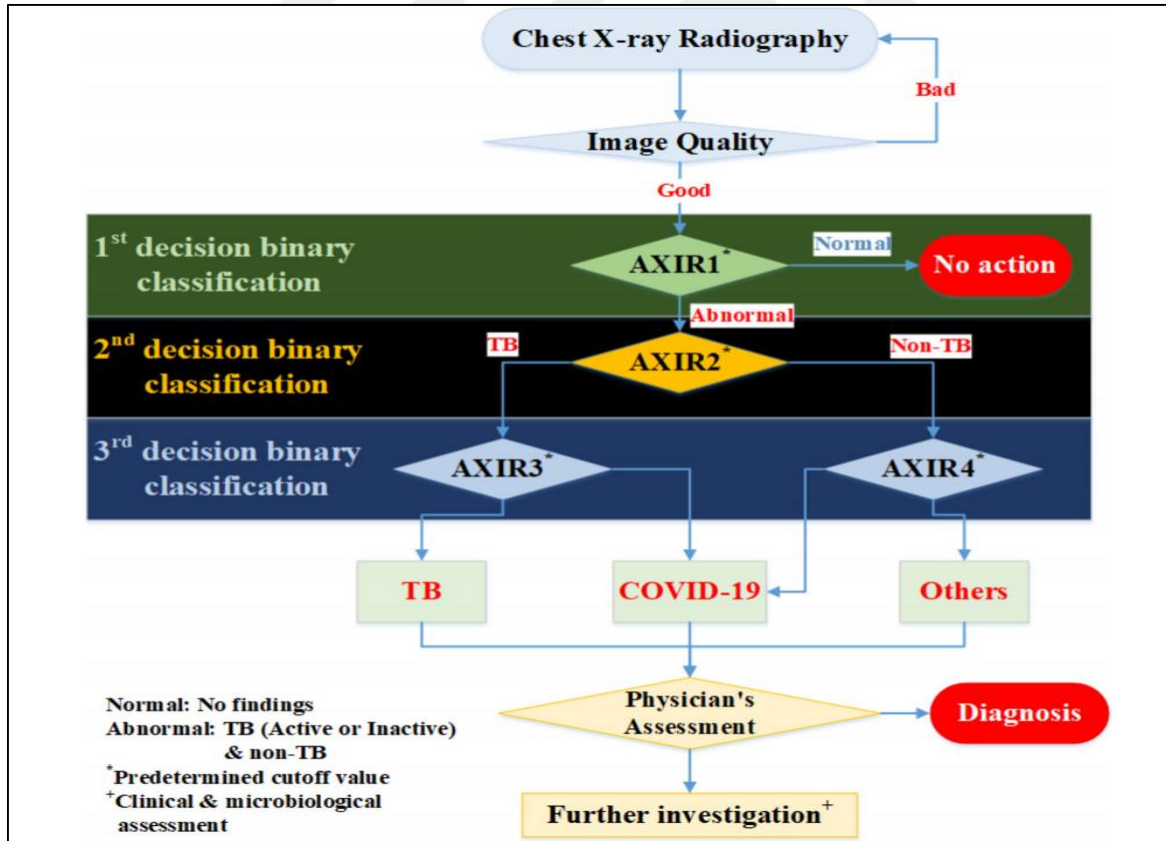
CoroNet, hazırlanan veri seti üzerinde eğitilmiş ve test edilmiştir. Deneysel sonuçlar, önerdikleri modelin % 89,6 genel doğruluk sağlandığını göstermektedir. Bununla birlikte daha da önemli olarak, COVID-19 vakaları için kesinlik ve geri çağırma oranı % 93 ve 4 sınıf vakalar için % 98,2. (COVID vs Pnömoni bakteriyel vs pnömoni viral vs normal) elde ettiğini göstermiştir. Ayrıca 3 sınıflı sınıflandırma için (COVID - Pnömoni - normal), önerilen model % 95 sınıflandırma doğruluğu sağlamıştır. Bu çalışmanın ön analizleri modelin kullanılabilirliği ile ilgili olumlu sonuçlar vermekle birlikte, daha fazla eğitim verisi elde edildikçe model daha da geliştirilebilecektir. Diğer bir deyişle CoroNet, daha fazla veri verildiğini gösteren küçük hazırlanmış bir veri setinde umut verici sonuçlar elde ettiği için önerilen model, verilerin minimum ön işleme tabi tutulması ile daha iyi sonuçlar sağlayacaktır.

Yoo ve diğerleri [9], yaptıkları çalışmada, her biri, PyTorch çerçevesine dayalı evrişim sinir ağına sahip bir derin öğrenme modelinde eğitilmiş üç adet ikili karar ağacı oluşturmuşlardır. İlk karar ağacı, CXR görüntülerini normal veya anormal olarak sınıflandırır. İkinci ağaç, tüberküloz belirtileri içeren anormal görüntüleri tanımlar, üçüncü ise aynı işlemi COVID-19 için yapmaktadır. Ön tarama hastalarında önerilen derin öğrenmeye dayalı karar ağacı sınıflandırıcısını, RT-PCR sonuçları çıkmadan önce triyaj yapmak ve hızlı karar vermeyi gerçekleştirmek için kullanmıştır. Veri kümesi ABD Ulusal Tıp Kütüphanesi, posteroanterior (PA) göğüs radyografilerinin iki veri setini kullanıma sunmuştur:

- MC seti ve Shenzhen seti, Her iki veri seti de TB'nin tezahürleri olan normal ve anormal göğüs röntgenlerini içerir ve ilgili radyolog okumalarını içerir.
- Ayrıca, NIH (Ulusal Sağlık Enstitüleri, ABD), NIH'nin veri tabanı <https://nihcc.app> adresinde çevrimiçi olarak mevcuttur.
- Patolojik tanıya dayalı akciğer hastalığı alt kalıpları. Bu veri seti, patolojik onay ile TB veya TB dışı sınıflandırma göstermez.
- Yukarıdaki açık veri setine ek olarak, TB ve verem dışı hastalıklar için Doğu Asya Hastanesi veri setini (Radisen ile işbirliğine dayanan) kullanılmıştır.

Bu çalışmada CXR'yi hastaneden veya muayene veri yolundan dört klinik duruma (normal, TB, TB olmayan ve COVID-19) ayırmak için üç seviyeli bir karar ağacı sınıflandırıcısı önerilmektedir. Her bir CXR'nin görüntü kalitesi çekildiğinde CAD modeli herhangi bir tahminde bulunmadan önce radyograflar tarafından kontrol edilir, Dört adet otomatik X-ışını görüntüleme radyografi sistemi (AXIR) vardır. AXIR1 ve AXIR2, görüntülerin sırasıyla anormal / normal ve TB / TB olmayan olarak sınıflandırılması içindir. AXIR 3 ve AXIR 4, görüntünün TB ile veya TB olmadan COVID-19 veya COVID-19 dışı olarak sınıflandırıldığı işlemin üçüncü adımında kullanılır.

İki aşamalı CAD sürecimizin eğitimi ve testi için PyTorch çerçeveli iki boyutlu bir CNN algoritması, ImageNet veri setini kullanan önceden eğitilmiş ResNet18'e dayalı CNN modeli (ResNet. Şekil 2.3), göğüs röntgeni görüntüsünün normal mi, tüberküloz (TB) veya COVID-19 ile enfekte bir akciğer gösterip göstermediğini belirlemeye yönelik AXIR (Otomatik X-ray Görüntüleme Radyografi sistemi) iş akışı aşağıdaki gibidir:



Şekil 2.3. ÇalışmanınGöğüs röntgeni görüntüsünü belirlemeye iş akışı [9]

Sonuçlar: Tüberküloz ve tüberküloz olmayan hastalıkların tespiti için derin öğrenme tabanlı üç seviyeli bir karar ağacı sınıflandırıcısı, COVID19 dahil, hasta verileri kullanılarak sunulmuş ve doğrulanmıştır. Her seviye için, optimize edilmiş eğitilmiş verilerle PyTorch çerçeveli iki boyutlu bir CNN algoritması (ResNet18 modeli) kullanılmıştır. AXIR1 (anormal ve normal veriler) ve AXIR2 (Tüberküloz ve tüberküloz olmayan veriler) için sırasıyla % 98 ve % 80 doğruluk elde edilmiştir.

AXIR2' nin daha düşük doğruluğu, FP atelektazi ve plevral kalınlaşma tahminleri ve FN infiltrasyonu, konsolidasyonu ve infiltrasyonundan kaynaklanmaktadır. AXIR 3 ve AXIR 4 ile ortalama% 95 doğruluk elde edilmiştir.

Wang, Lin ve Wong [10], yaptıkları çalışmada COVID-19 vakalarını tespit etmek için tasarlanan COVID-Net'i tanıtmışlardır. COVID-Net, ilk sürümde CXR görüntülerinden COVID-19 tespiti için ilk açık kaynaklı ağ tasarımlarından biridir. Araştırmacılar COVID-Net'in bir girişimde açıklanabilirlik yöntemi kullanarak nasıl tahminler yaptığını araştırıyorlar. Bu yöntem ile yalnızca COVID vakalarıyla ilişkili kritik faktörlere ilişkin daha derin bilgiler elde etmekle kalmaz, klinisyenlere gelişmiş taramada yardımcı olabilir. Bununla birlikte CXR görüntülerinden alınan ilgili bilgilere dayanarak kararlar aldığını doğrulamak için COVID-Net kullanılabilir. COVIDx veri kümesi, Önerilen COVID-Net'i eğitmek ve değerlendirmek için kullanılan Te veri kümesi, COVIDx olarak adlandırılan, 13.870 hasta vakasında toplam 13.975 CXR görüntüsünden oluşmaktadır. Önerilen COVID veri kümesi, COVID-19 pozitif hasta vakalarının sayısı açısından en büyük açık erişim karşılaştırma veri kümesidir. Bu çalışmada, COVID-Net oluşturmak için insan-makine işbirliğine dayalı tasarım stratejisi kullanılmıştır. Bu stratejide COVID-19 vakalarının CXR görüntülerinden algılanması için uyarlanmış bir ağ mimarisi üretmek amacıyla insan odaklı ilkel ağ tasarımı prototiplemesi makine odaklı tasarım araştırmasıyla birleştirilmiştir.

COVID-Net'in eğitimi ve değerlendirmesini kolaylaştırmak için COVIDx adlı bir açık erişim karşılaştırma veri kümesi de oluşturulmuştur. Hafif tasarım desenleri ve çok düşük mimari çeşitliliğe sahiptir. Ayrıca ResNet-50/40 Kalan tasarım ilkelerinden ve hafif tasarım modellerinden (ör. Darboğaz tasarım modelleri) yararlanan derin bir sinir ağı mimarisi kullanılmıştır. Orta düzeyde mimari çeşitliliğe sahiptir, ancak hafif PEPX tasarım modellerinden ve seçici uzun menzilli bağlantıdan yararlanmamaktadır. Bu iki derin sinir ağı mimarisinin seçimi, COVID-Net'in temel tanımlayıcı özelliklerine sahip olmadıkları

gerçeğine dayanmaktadır. Böylece COVID-Net'in benzersiz özelliklerinin faydalarının daha iyi anlaşılmasına olanak tanır.

COVID-Net ile COVID-19 vakaları için iyi bir hassasiyet elde edilmiştir (% 91.0 hassasiyet). Bu hassasiyet yakalanamayan COVID-19 vakalarının sayısını sınırlamak isteği için önemlidir. COVID-Net, COVID-19 vakaları için (% 98,9 PPV) yüksek PPV elde eder, bu da çok az sayıda yanlış pozitif COVID-19 tespitini gösterir. Yüksek PPV, çok fazla sayıda yanlış pozitifin, ek PCR testi ve ek bakıma ihtiyaç duyulması nedeniyle sağlık sistemi yükünü artıracak için önemlidir. Bu nedenle, bu sonuçlara dayanarak, COVID-Net'in CXR görüntülerinden COVID-19 vakalarını tespit etmede bir bütün olarak iyi performans gösterdiği görülebilir. Ek veri toplamak suretiyle birkaç iyileştirme alanı bulunmaktadır ve bu tür senaryolarda daha iyi genelleme yapmak için temel eğitim metodolojisinin iyileştirilmesi önemlidir.

Zebin ve Rezvy diğerleri [11], tarafından yapılan çalışmada ,COVID-19 göğüs röntgeni görüntülerini halka açık iki göğüs röntgeni veri kümesinden sınıflandırmak için bir aktarım öğrenme kanalı uyguladılar. Sınıflandırıcı, COVID-19 ve Pnömoniye bağlı akciğerlerdeki iltihabı, enfeksiyonu olmayanlardan (normal) etkili bir şekilde ayırmaktadır. Özellik çıkarıcı olarak çok sayıda önceden eğitilmiş evrişimli omurga kullanılmıştır. Sağlıklı hastalar için normal, etiketli COVID-19 hastaları için ve Pnömoni - viral ve bakteriyel Pnömoni hastaları için üç sınıflı bir veri kümesi oluşturulmuştur. Github'da herkese açık olan "COVID-19 Görüntü Verisi Toplama" COVID-19 görüntülerin ana kaynağını oluşturmuş olup, anonim COVID-19 görüntüleri tıbbi ve bilimsel derneklerin web sitelerinden ve COVID ile ilgili araştırma makalelerinden faydalanılmıştır. Bu veri kümesi, sürekli büyüyen bir veri kümesi olup toplam veri seti, COVID-19'dan ve MERS, SARS ve ARDS gibi diğer hastalıklardan etkilenen 349 hastadan alınan 673 X-ray ve CT görüntüsünden oluşmaktadır.

Ön işlemede, birleştirilmiş veri kümesindeki her görüntü, hesaplama süresini kısaltmak amacıyla 224x224 piksel olarak yeniden boyutlandırılarak normalleştirme aşaması uygulanmıştır. Bu aşama piksel yoğunluklarını [0, 255] aralığına normalleştirilir. Veri kümesine, artırma tekniğinin iki farklı versiyonu uygulanır. İlkinde rastgele döndürme TensorFlow ,Keras kullanarak görüntü büyütme teknikleri uygulanmıştır. Yeni görüntülerin kaynak veri kümesinden farklı kategorileri vardır. Önceden eğitilmiş model, aktarım öğrenme ilkelerine dayalı olarak bu görüntülerden ilgili özellikleri çıkarmak için kullanılır.

İşlem hattının çeşitli işlevlerini oluşturmak için Python'da TensorFlow, Keras, PyTorch, Scikit-learn ve OpenCV kitaplıklarını kullanmışlardır. Göğüs röntgeni görüntü verilerinden jenerik düşük seviyeli tanımlayıcıları veya modelleri çıkarmaya yardımcı olmak için önceden eğitilmiş omurganın önceki katmanlarının ağırlıklarını dondururlar. Yukarı çıktıkça, özellikler, modelin eğitildiği veri kümesine giderek daha spesifik hale gelmektedir. İnce ayarlamamanın amacı, genel öğrenmenin üzerine yazmak yerine, bu özel özellikleri yeni beslenen COVID-19 veri kümesiyle çalışacak şekilde uyarlamaktır. Özellik çıkarma deneyinde, bir temel modelin üzerinde yalnızca birkaç katman eğitilmiştir.

Sonuç: Özellik çıkarıcı olarak birden çok önceden eğitilmiş evrişimli omurga kullanılmış ve VGG16, ResNet50 ve EfficientNetB0 omurgaları için sırasıyla % 90, % 94,3 ve % 96,8'lik genel algılama doğruluğu elde etmişlerdir. Ek olarak, yaklaşımlarında azınlık COVID19 sınıfını oluşturmak ve artırmak için üretken bir karşıt çerçeve (bir CycleGAN) eğitmişlerdir. Görsel açıklamalar ve yorumlama amacıyla, tahminler için önemli olan girdi görüntüsünün bölgelerini vurgulamak için bir gradyan sınıfı etkinleştirme haritalama tekniği uygulanmıştır. Bu görselleştirmeler, hastalığın ilerlemesi ve ciddi aşamaları sırasında etkilenen akciğer bölgelerini izlemek için kullanılabilir.

Brunese, Mercaldo, Reginelli ve Santone [12], yaptıkları çalışmada üç aşamadan oluşan bir yaklaşım önermektedirler. Araştırmalarında, aşağıdakileri amaçlayan üç aşamalı bir yöntem önermektedirler: (i) göğüs röntgeni sağlıklı bir hastayla mı yoksa jenerik akciğer hastalığı olan bir hastayla mı ilişkili olduğunu saptamak; (ii) jenerik akciğer hastalıkları ile COVID-19 arasında ayırım yapmak. (iii) göğüs röntgeni içindeki COVID-19 hastalığının semptomatik alanlarını vurgulamak.

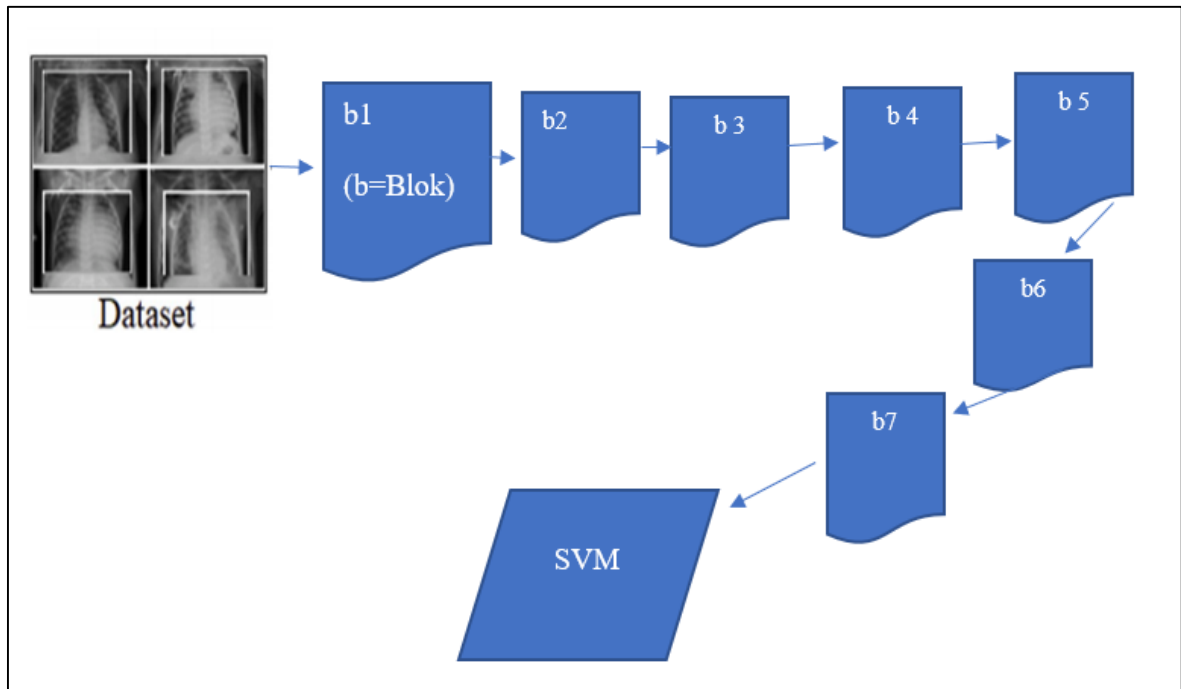
İlgili tüm veri setleri “Ulusal Sağlık Enstitüleri Göğüs Röntgeni”den alınan göğüs röntgeni koleksiyonları olup özel olarak aşağıdaki amaçlara yönelik araştırma yapılmıştır:

- Göğüs röntgeninin sağlıklı bir hastayla mı yoksa bir akciğer hastalığından etkilenen bir hastayla mı ilgili olduğunu tespit etmek.
- Göğüs röntgeni akciğer hastalığı etiketi ile işaretlendiğinde akciğer hastalığının pnömoni veya COVID-19 olup olmadığını tespit etmek.

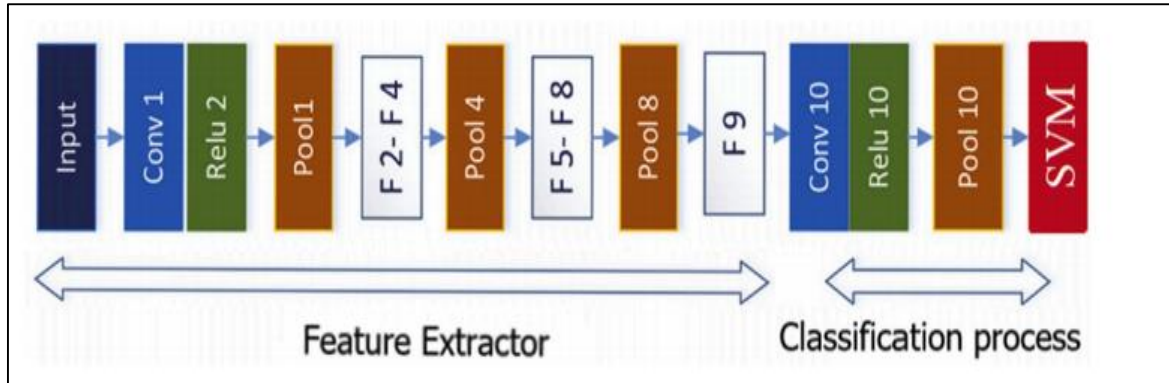
Çalışmada Transfer Öğrenmeden yararlanılarak 16 katmanlı evrişimli bir sinir ağı olan VGG-16 (yani Görsel Geometri Grubu) modeli kullanılmıştır. Transfer öğrenimi, tipik olarak, bir problem üzerinde eğitilen bir modun, ikinci bir problemle ilgili etiketi tahmin etmek amacıyla kullanıldığı bir süreçle ilgilidir. Transfer öğrenmenin benimsenmesindeki ana fayda, bir sinir ağı modeli için eğitim görevinin azalan süresiyle temsil edilebilir olmasıdır. Dahası, COVID-19 tespiti için ortalama 2,5 saniyelik ortalama süre ve 0,97'ye eşit ortalama doğruluk sonuçları ile daha düşük genelleme hatası sağlayabilmektedir.

Toğaçar, Ergen ve Cömert [13], tarafından yapılan çalışmada yapay zeka modelleri, biyomedikal alanda başarılı analizler için yardımcı olmuştur. Bu çalışmada Koronavirüs, yapay zekanın bir alt dalı olan derin öğrenme modeli kullanılarak tespit edilmiştir. Deneysel analizde, herkes tarafından erişilebilen üç veri kümesi sınıfını kullanmıştır. Bu sınıflar normal, zatürre ve COVID-19 göğüs görüntüleridir.

Tüm veri kümeleri X-ışını görüntüleridir ve her görüntü JPG formatına dönüştürülür. COVID-19 görüntülerinden oluşan halka açık iki veritabanından alınan görüntüleri birleştirmişlerdir. Bu veri tabanlarının ilki COVID-19 veri kümesi GitHub web sitesinde Montreal Üniversitesi'nden Joseph Paul Cohen adlı bir araştırmacı tarafından paylaşılmış olup ikinci olarak COVID-19 veri kümesi Kaggle web sitesinden faydalanılmıştır.



Şekil 2.4. Çalışmanınkullanılan MobileNetV2 modelinin genel tasarımı [13]



Şekil 2.5. Çalışmanın[13] kullanılan SqueezeNet şablonunun genel tasarımı

Araştırmada düşük donanım maliyetli cihazlarda kullanılmak üzere tasarlanmış bir derin öğrenme modeli olan MobileNet kullanılmıştır. MobileNet modeli kullanılarak nesne tanımlama, bölümleme ve sınıflandırma çalışmaları gerçekleştirilebilir. Bu yeni model, katmanlar arasındaki doğrusallık sorunlarına en büyük katkıyı sunmaktadır.

Katmanlar arasında doğrusal darboğazlar oluşursa, sorunlar bu sürümde kısayollarla giderilir. Ayrıca, $224 * 224$ piksellik girdi boyutuna sahip derinlemesine bir öğrenme modeli olan SqueezeNet kullanılmıştır. Evrişimli katmanlar, havuz katmanları, ReLU ve Fire katmanlarından oluşur. SqueezeNet, tamamen bağlı katmanlara ve yoğun katmanlara sahip değildir. Bu modelin en büyük yararı, parametre sayısını azaltarak analizleri başarılı bir şekilde gerçekleştirmesi ve böylelikle model boyutu kapasitesini azaltmasıdır. SqueezeNet modeli, AlexNet modeline göre yaklaşık 50 kat daha az parametre ve daha başarılı sonuçlar vermiştir, dolayısıyla modelin maliyetini düşürmektedir.

Bulanık renk tekniği, bulanık kavramı, bir sonraki derecesi belirsiz olan doğruluk derecesine göre kabul edilir. Bulanık Renk algoritmaları, görüntü analizinde önemli bir rol oynar ve elde edilen sonuçlar, renk ayrımı için kullanılan benzerlik / farklılık fonksiyonlarına bağlıdır. Bulanık renk tekniğinde, girdi görüntülerinin her biri üç girdi değişkeni (kırmızı, yeşil ve mavi - RGB) içerir. Bu işlem sonucunda tek bir çıktı değişkeni geçirilir.

SVM, regresyon ve sınıflandırma analizi için kullanılan bir makine öğrenimidir. Bu yöntem, veri sınıflarındaki özellikleri ayırmak için sınıflandırma sürecinde bir hiper düzlem çizgisi

kullanır. Bir çizgi belirlemek için sınıfların özelliklerinden uzak bir lokasyon seçer. Her sınıfın mesafeleri, SVM yöntemi ile belirlenen hiper düzleme göre ölçülür. En yüksek oy puanına sahip olan, en yüksek oylama puanı olarak adlandırılan sınıfa aktarılır.



3. YAPAY ZEKA VE MAKİNE ÖĞRENMESİ

Yapay zeka (AI), düşünen bir insan gibi, yani onu taklit etmek için tasarlanmış sistemlerdir. Aynı zamanda insanlar yerine insanlar için zor olan bazı zor görevleri yerine getirir. Büyük verileri analiz etmek için kullanılan sohbet robotları ve robotlar dahil olmak üzere birçok forma sahiptir.

Makine öğrenimi, büyük veri kümelerinin sağladığı önemli algoritmalar yardımıyla yazılımsal makinelerin tıpkı bir insan beyni gibi çalışmasının sağlanmaya çalışıldığı özetle makinelerin kendi kendilerine öğrenmeleri anlamına gelen bir yaklaşımdır. Makine öğreniminde yazılımsal olarak birçok bilgi ve veri formunun sistemlere yüklenmesi ile makinelerin bu veri setlerini sınıflaması, yeni karşılanan benzer durumları algoritmalar kullanarak belirlemesi ve hatta insanlar gibi istatistiksel hesaplamalar yaparak tahminlerde bulunmasını sağlamaktadır [13].

3.1. Denetimli ve Denetimsiz Öğrenme

Denetimli bir öğrenme algoritması, etiketli eğitim verilerinden öğrenir, öngörülemeyen veriler için sonuçları tahmin etmeye yardımcı olur. Denetimli öğrenme, önceki deneyimden veri toplamaya veya bir veri çıktısı üretmeye izin verir. Deneyimi kullanarak performans kriterlerini optimize etmeye yardımcı olur ve Denetimli Makine Öğrenimi, çeşitli gerçek dünya sorunlarının çözülmesine yardımcı olur. Bu nedenle veri bilimcisi, verilen içgörülerin verileri değişene kadar doğru kalmasını sağlamak ve veri değişimlerine yönelik modelleri yeniden oluşturabilmelidir.

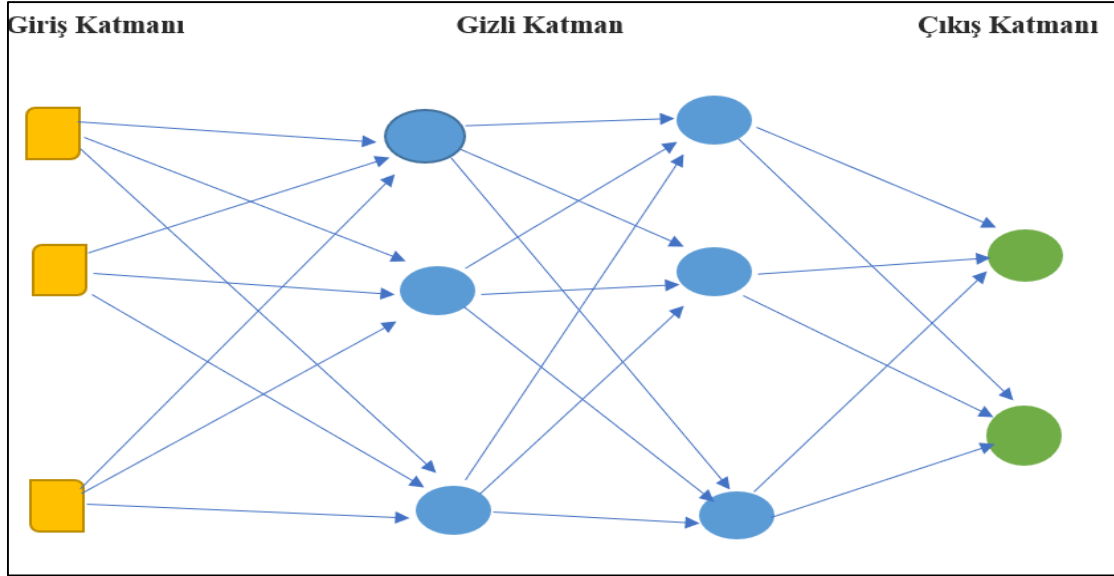
Denetimsiz Öğrenme, Bu tipte, belirli bir mimariye sahip olmayan bir veri kümesi kullanılır. Ayrıca, kontrol sürecini gerçekleştirmek için forma ihtiyacınız yoktur. En önemli faydalarından biri, çok zor ve hassas görevleri yerine getirebilmesidir [14].

3.2. Yapay Sinir Ağları (Neural Networks)

Yapay sinir ağları insan beynini uyaran ve insan beyninin bilgi işleme süreçlerinden esinlenerek geliştirilmiş bir işletim sistemi gibidir. Tıpkı insan sinir ağlarının çalışma

prensibini temele alan sinir ağlarının etkili şekilde çalıştığı bilgi işleme teknolojisidir. Yapay Sinir Ağları üç katmandan oluşur:

- Birincisi, veri giriş katmanıdır
- İkincisi, üzerinde çok sayıda veri işleme işleminin gerçekleştirildiği gizli katmandır.
- Üçüncü Katman sonuçların çıktısının alındığı çıktı katmanıdır.



Şekil 3.1. Sinir ağları türü

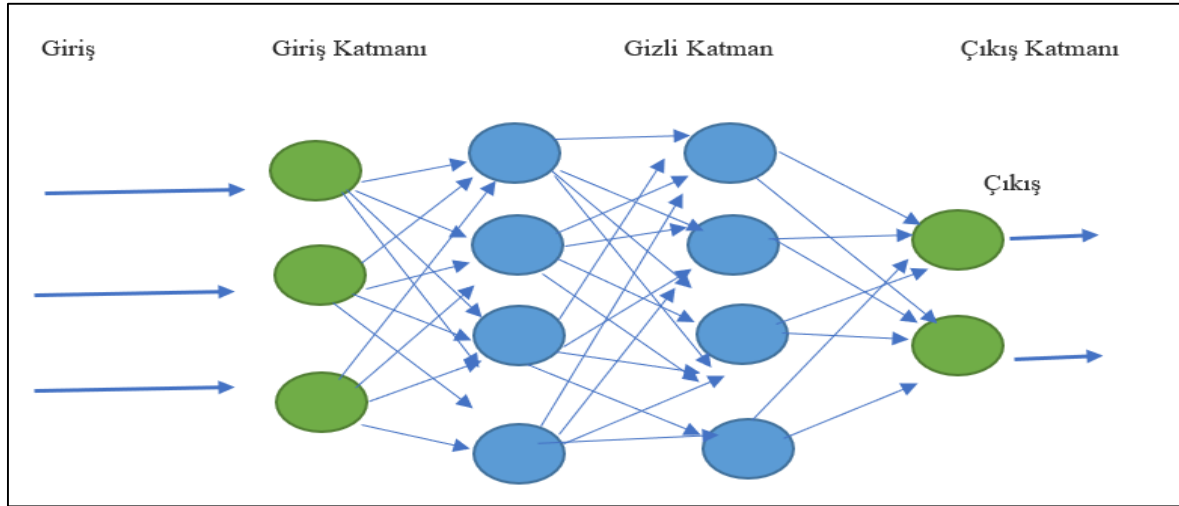
3.3. Derin Öğrenme

Farklı bir veri setini kullanma olasılığını sağlayan öğrenmedir. Bilgisayar, bir grup ses arasından belirli bir sesi çıkarmak ve tanımlamak için görüntüleri sınıflandırmak veya sesi sınıflandırmak gibi çeşitli sınıflandırma görevlerini öğrenir ve gerçekleştirir.

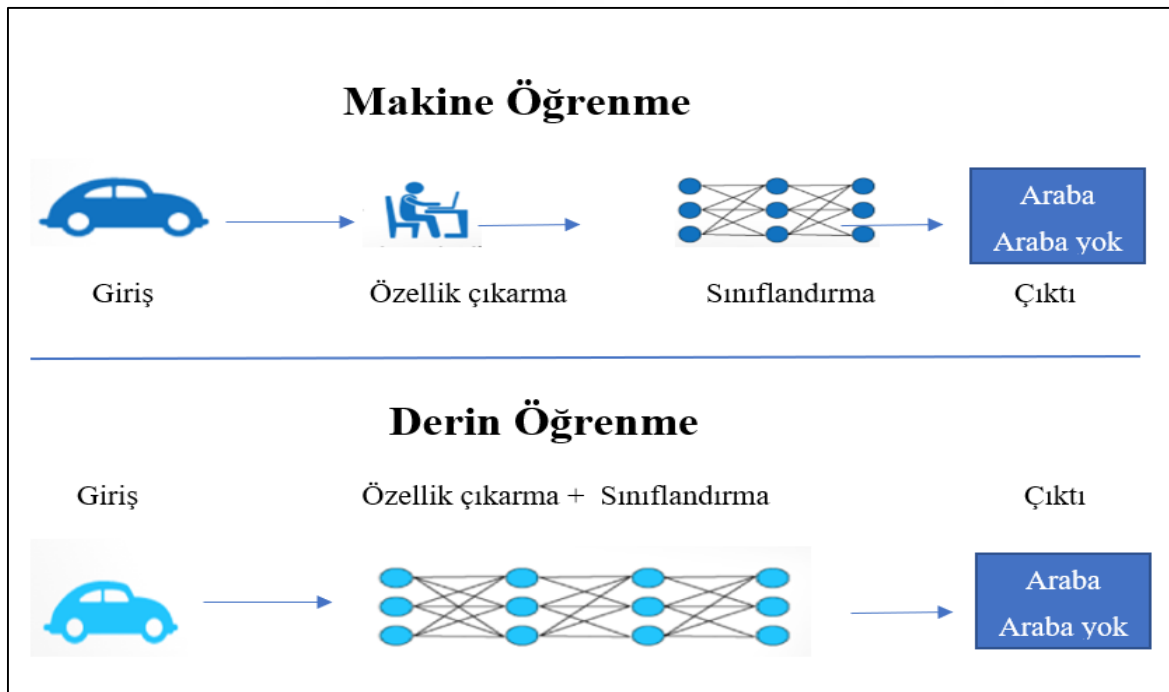
Son beş yılda derin öğrenme uygulamalarının gelişimi hakkında ciddi mesafeler kat edilmiştir. Derin Öğrenme, esas olarak görüntülerden otomatik özellik çıkarma ve sınıflandırmaya odaklanan Makine Öğrenimi yöntemlerinin bir kombinasyonudur. Geniş bir uygulama alanı olan derin öğrenmenin nesne algılama veya tıbbi görüntü sınıflandırma görevleri bulunmaktadır. Makine öğrenimi ve Derin Öğrenme, verilerden kalıpları çıkarmak, analiz etmek ve tanımak için yapay zeka uygulamasında yerleşik disiplinler haline gelmiştir [15-17].

3.3.1. Derin öğrenmenin çalışması

Derin öğrenme çalışmasındaki ağlar kendi kendilerine öğrenirler. Bu ağlar, tüm aşamalarda birçok katman içerebilir. Bu ağların çalışabilmesi için, özellikleri eğitilmesinden önce hesaplamalar ve veri işlemlerinin ağlara girilmesi gerekmektedir.



Şekil 3.2. Derin öğrenme mimarisi



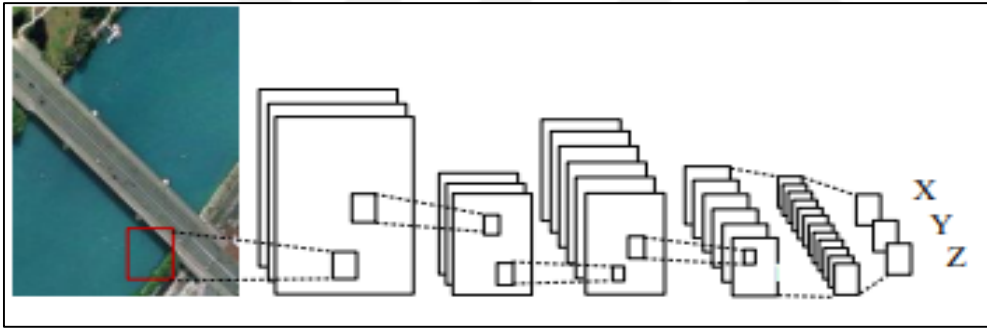
Şekil 3.3. Makine öğrenimi ve derin öğrenme arasındaki fark

3.3.2. Derin öğrenme algoritmaları

Derin öğrenme mimarileri oluşturulurken, kullanılacak algoritma büyük önem taşımaktadır. Bu algoritmalar verinin türüne, boyutuna, hacmine ve yapısına, kullanılacak olan parametrelere göre farklılık gösterebilmektedir.

Konvolüsyonel Sinir Ağları (CNN)

Bu ağ, görüntü ve uydu işlemlerinde kullanışlıdır. Çok normal bir sinir ağı türü olan bu tür derin öğrenme algoritmalarından olmasına rağmen diğer algoritmalarından farklı bir mimariye sahiptir. Ana fikir, resimler üzerinde çalıştığı ve bunlara katman grupları ve yeni katmanları uyguladığı, sonunda bir nöron ortaya çıktığı ve daha sonra bunun ne olduğunun belirlendiği son katmana olan çıktı katmanına ulaşmasıdır.



Şekil 3.4. Sınıflandırılmış evrişimli sinir ağı modeli

Nasıl çalışır

Onlarca veya yüzlerce gizli katman kullanarak bir görüntünün farklı özelliklerini algılamayı öğrenir. Her gizli katman, öğrenilen görüntü özelliklerinin karmaşıklığını artırır. Örneğin, ilk gizli katman kenarları nasıl algılayacağını öğrenebilir ve sonuncusu, tanımaya çalıştığımız nesnenin şekline özel olarak hazırlanmış daha karmaşık şekilleri nasıl algılayacağını öğrenebilir. Özellikle görüntüler dört boyutlu tensörler olarak değerlendirilir. Skaler sıfır boyutlu bir nesneyse, vektör tek boyutludur. Bir matris veya vektör koleksiyonu iki boyutludur ve bu tür matrislerden oluşan bir yığın (küp olarak gösterilmiştir) üç boyutludur.

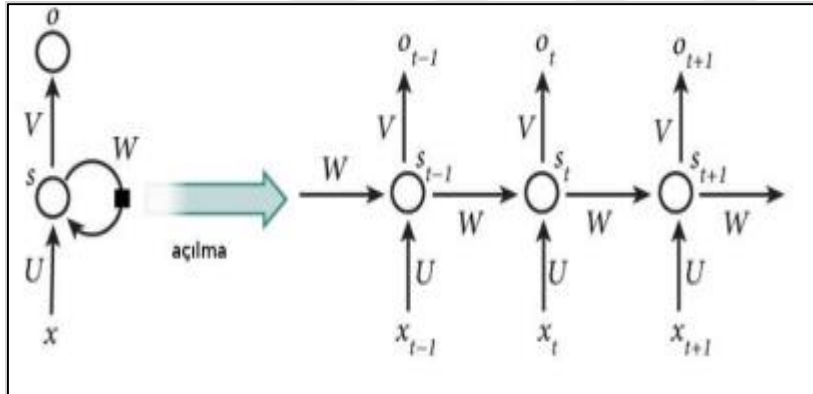
Faydaları

CNN algoritması, tanımda etkilidir ve yüksek oranda uyarlanabilir. Daha az eğitim parametresi olduğu ve geri yayılımla birleştirildiğinde ölçeklenebilir olduğu için eğitilmesi de kolaydır [12]. Kullanım örnekleri: CNN algoritması aşağıdakilerle kullanılabilir:

Görüntü işleme, Tanıma, sınıflandırma Video tanıma, Doğal dil işleme görevleri, Desen tanıma, Öneri motorları ve Tıbbi görüntü analizi.

Tekrarlayan Sinir Ağı (RNN)

Bu ağı en önemli özelliklerinden biri, önceki sonuçlara dayanarak ardışık ve birbirinden bağımsız verileri kullanmasıdır. Bu ağ, geleneksel ağlardan farklı olarak farklı girdi verileriyle ilgilenir.



Şekil 3.5. Tekrarlayan sinir ağı açıklaması

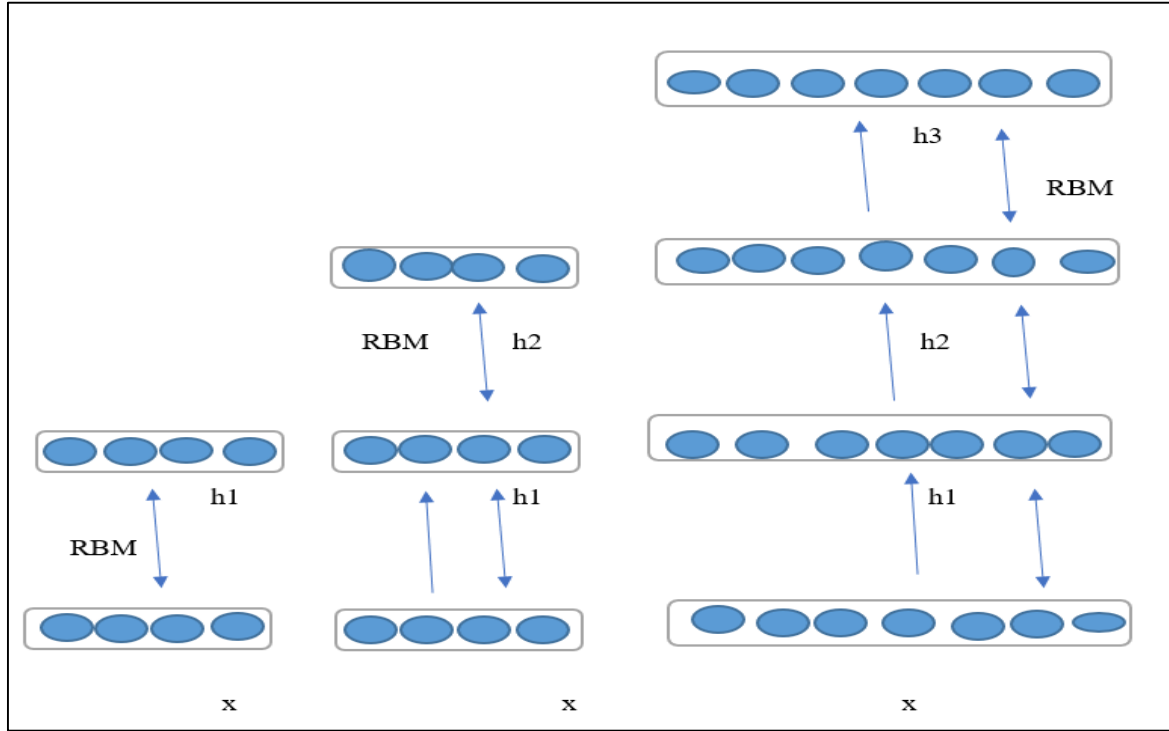
İleri beslemeli sinir ağı (FFNN)

Sinir ağının olduğu yerlerde sesleri tanımak için kullanılan ağlardan biridir, Sesleri birbirinden ayırmak için kullanılan özel bir ses eklenir. Her ses yalnızdır ve sesin kaydını yaparak metne dönüştürür. Aynı zamanda, belirli kameraların bir şeyleri sınıflandırmak için kullanıldığı bilgisayar görüşünde ve sürücüsüz arabalarda da kullanılır.

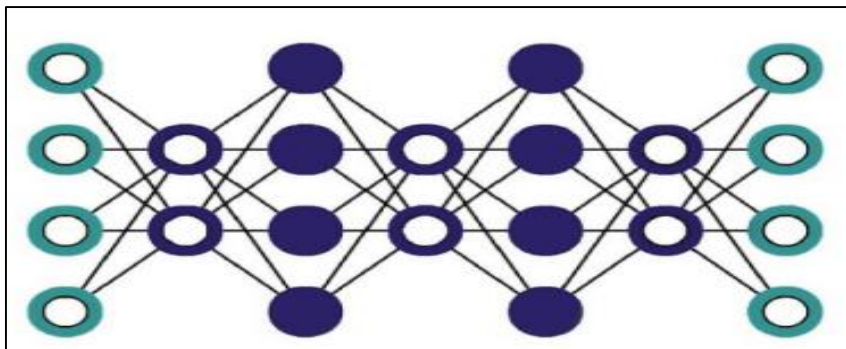
Derin İnanç Ağı(DBN)

Bu ağlar, h harfi ile temsil edilen bir giriş katmanı içerdikleri için hiyerarşik gösterimi grafik olarak kullanılır.Bu algoritmanın en önemli özelliklerinden biri, kalıp oluşturmak ve sonra

onları tanımlamak için kullanılır. Bu, yapay zekayı insanlığa yardımcı olan gelişim ve öğrenmenin başka bir yönüne aktaran çok önemli bir özelliktir.



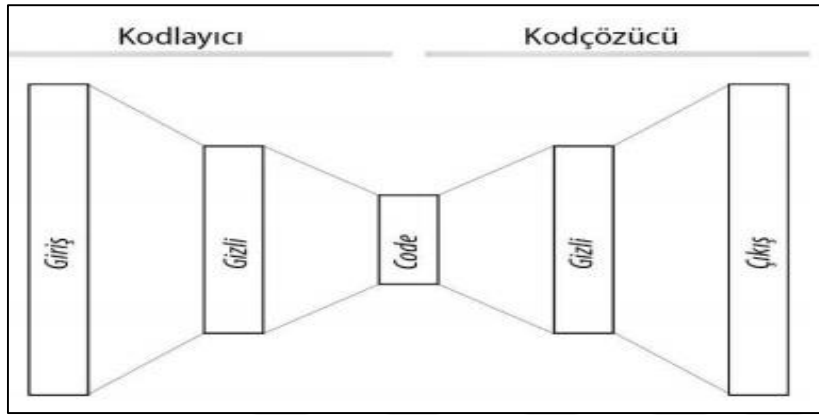
Şekil 3.6. Derin inanç ağı açıklaması



Şekil 3.7. Derin inanç ağı modeli

Derin otokodlayıcılar

Bu sinir ağıları modelleri denetimsiz öğrenmeyle ilgilenirler. Öğrenme sırasında kaybı azaltmak için verileri sıkıştırırlar ve diğer modellerde olduğu gibi 3 katman içerirler. Bu algoritma 2009 yılında bu ağın oluşumuna büyük katkıda bulunan kişi olduğu göz önüne alındığında Dibalou'nun adı verilerek 2012'de derin öğrenme algoritması olarak kabul edilmiştir.



Şekil 3.8. Otomatik kodlayıcı açıklaması



4. MALZEMELER VE YÖNTEMLER

Malzemeler ve yöntemler, test aletleri, teknik ekipman, kitap, resim veya araştırma sırasında kullanılan diğer malzemeleri içerebilir. Bu kısmının ana amacı, deney tasarımını tarif etmek (ve gerekliyse savunmak) ve sonra, konuyu bilen bir kişinin bu deneyleri tekrar edebileceği ayrıntıları vermektir.

4.1 Veri Kümesinin Toplanması

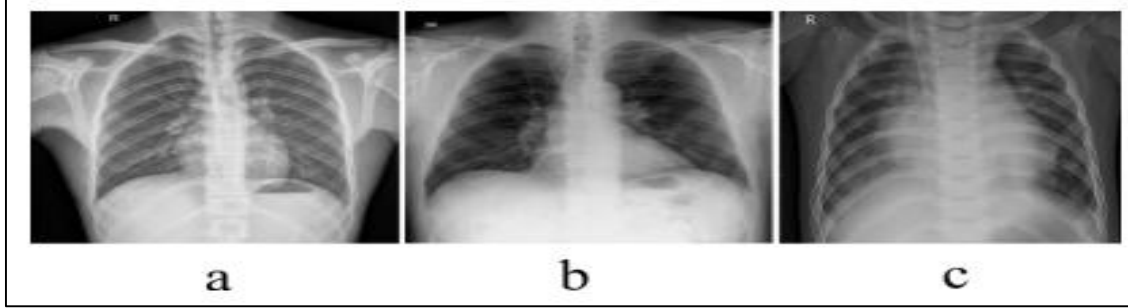
Bir Derin Öğrenme projesinin ilk ve en önemli kısmı veri toplamaktır. Araştırmanın veri kümesi, Kaggle Göğüs X-ışını veri kümesinin, Praveen veri kümesinden toplanan pnömoni Göğüs X-ışını veri kümesi ile birleştirilmesi ile toplanmıştır. Her iki veri seti de en çok tercih edilen ve en sık görülen göğüs röntgeni görüntüleri olan hastaların arka-ön göğüs görüntülerinden oluşmaktadır. Montreal Üniversitesi'nde Doktora Sonrası Araştırmacı olan Dr. Joseph Paul Cohen, Yakın zamanda, COVID-19'un yanı sıra MERS, SARS ve ARDS den muzdarip hastaların göğüs röntgeni ve BT görüntülerini içeren halka açık bir veri seti yayınlamıştır.

COVID 19 veri kümesi, daha fazla vaka yayınlandıkça günlük olarak güncellenmektedir. Bu projede kullanılan görüntülere 18 Mart 2020 tarihinde erişilmiştir. Kaggle'dan alınan veri seti, Kuzey Amerika Radyoloji Derneği (RSNA) tarafından sağlanmaktadır.

RSNA, Ağustos 2018 de Kaggle'da 30.000 \$ 'lık para ödüllü bir yarışmaya ev sahipliği yaparak, makine öğrenimi topluluğunu göğüs radyografilerinden pnömoniyi tespit etmek için bir algoritma oluşturmaya davet etti [18]. Rekabetin özel gerekliliği, modelin, bir nesne algılama sorunu olan ve bu projedeki görüntü sınıflandırmasından daha zor olan akciğer opasitelerini bulması gerektiğidir. Bununla birlikte, bu veri seti hala bu proje için değerli olan binlerce normal görüntü ve pnömoni ile enfekte hasta sunmaktadır.

Eğitime hazır bir veri kümesi oluşturmak için, önce bu iki veri kümesinden görüntüler indirilmelidir. Ardından, istenen görüntüler seçilir ve doğru şekilde etiketlenmiş klasöre kaydedilir. Daha sonra görüntüler, eğitim sürecine beslenmeye hazır olmak için istenen boyutta NumPy dizilerine dönüştürülerek yüklenir ve ön işlemden geçirilir. Veri kümesi, 500 Covid-19 pozitif görüntü ve 1600 Covid-19 negatif görüntü içermektedir. Cohen veri

kümesi ve Praveen tarafından hazırlanan pnömoni veri kümesi ilişkin olarak Praveen veri setini covid-19 Negatif görüntüsü olarak kullanılmıştır.



Şekil 4.1. (a) Sağlıklı birey, (b) COVID-19 pozitif hasta veri seti, (c) pnömoni hastası.

4.2 Görüntü Ön İşleme

Verileri sinir ağlarına yüklemekten önceki adım, verileri önceden işlemektir. Bu, görüntüleri NumPy dizileri olarak okumayı, görüntüleri istenen boyuta yeniden boyutlandırmayı, görüntüleri normalleştirmeyi (piksel değerini 0 ile 1 arasında yapma) içerir. İşleme süreci etiketleri kodlamak, veri kümesini eğitime ve verileri doğrulamaya bölmek ve son olarak veri büyütme şeklinde gerçekleştirilir. Veri artırma, aşırı uyum sağlamak için yardımcı olabilecek bir tekniktir.

Temel olarak, eğitim aşamasında, orijinal görüntü üzerinde döndürme, kesme, kaydırma, çevirme gibi bazı dönüşümleri rastgele gerçekleştirecektir. Sonuç olarak, model aynı görüntüyü iki kez beslemeyecek ve böylece eğitim setindeki tüm görüntüleri hatırlayamayacak. Veri artırma ile eğitilen model daha sağlamdır ve 'ImageDataGenerator' ile daha basit olup onu Keras'ta uygulayarak daha iyi genelleştirilebilir.

Bu çalışmada, CNN modelini kullanarak göğüs röntgeni görüntülerinden Covid-19'u tespit etmek için derin bir öğrenme yaklaşımına odaklanılmıştır. Verilerin ön işlenmesindeki önemli aşamalardan biri de görüntülerin boyutu, algoritmanın girdisi farklı olduğu için röntgen görüntülerinin boyutu değiştirilmiştir. Tüm Görüntüler, bilgisayarla görme problemlerini çözmek için tasarlanan, OpenCV (cv2) python görüntü işleme kütüphanesi tarafından okunmuştur. Siyahtan, beyaza ve orta gri tonlarına kadar değişen düzenli aralıklı tonlardan oluşan ve yalnızca gri tonlamadan oluşan görüntüyü 100x100 olarak yeniden boyutlandırılmış ve {0,1} kategorize etmek için görüntü eklenmiştir. Burada {0} negatif bir

covid-19'u ve {1} pozitif bir covid-19'u temsil etmektedir. Görüntü normalleştirme adımında tüm piksel değerleri değiştirilmiştir.

Normalleştirme, aralıktaki (Min, Maks) yoğunluk değerlerine sahip n boyutlu bir gri tonlamayı, aralıktaki (newMin, newMax) yoğunluk değerlerine sahip yeni bir görüntüye dönüştürür. Gri tonlamalı bir dijital görüntünün doğrusal normalizasyonu aşağıdaki formüle göre gerçekleştirilir:

$$I_N = (I - Min) \frac{newMax - newMin}{Max - Min} + newMin \quad (4.1)$$

Eş.4.1'de . Örneğin, görüntünün yoğunluk aralığı 50 ile 180 ve istenen aralık 0 ile 255 ise, işlem piksel yoğunluğunun her birinden 50 çıkarılmasını gerektirir, aralığı 0 ile 130 yapar. Daha sonra her piksel yoğunluğu 255/130 ile çarpılarak 0 ile 255 aralığı oluşturulur. Normalleştirme ayrıca doğrusal olmayabilir, Doğrusal olmayan normalizasyonun bir örneği, normalleştirme bir sigmoid işlevini takip ettiğinde, bu durumda normalleştirilmiş görüntünün aşağıdaki formüle göre hesaplanmasıdır:

$$I_N = (newMax - newMin) \frac{1}{1 + e^{-\frac{1 - \beta}{\alpha}}} + newMin \quad (4.2)$$

Eş.4.2'de. α giriş yoğunluğu aralığının genişliğini ve β aralığın ortalanmış olduğu yoğunluğu tanımladığı yerde. Görüntü işleme yazılımındaki otomatik normalleştirme, tipik olarak görüntü dosyası biçiminde belirtilen sayı sisteminin tam dinamik aralığına normalleştirilmiştir [19].

4.3. Evrişimli Sinir Ağının (CNN) Eğitimi

Evrişimli sinir ağı, ismi ağın evrişim adı verilen matematiksel bir işlem kullandığını gösterir. Evrişimli ağlar, derin öğrenmede katmanlarından en az birinde genel matris çarpımı yerine evrişimi kullanan özel bir sinir ağları türüdür. Bu ağın isimlerinden bir diğeri, sabit bir ağ anlamına gelen SIANN'dir. Gizli katmanları ve çeviri değişmezlik özelliklerini tarayan evrişim çekirdeklerinin paylaşılan ağırlıklı mimarisine dayanmaktadır. Görüntü ve video tanıma, tavsiye sistemleri, görüntü sınıflandırma, görüntü bölütleme, tıbbi görüntü analizi,

doğal dil işleme, beyin-bilgisayar arayüzleri ve mali zaman serilerinde uygulamaları vardır [20,21].

Çalışmada, en popüler derin sinir ağlarından CNN modelleri benimsenmiştir. Evrişim, özel bir doğrusal işlem türüdür. CNN, özellikleri doğrudan görüntülerden çıkararak çalışır. Ağ bir görüntü koleksiyonu üzerinde eğitim alırken ilgili özellikler eğitilmez. Bu otomatik özellik çıkarma, derin öğrenme modellerini görüntü sınıflandırma gibi bilgisayarla görme görevleri için son derece hassas hale getirir.

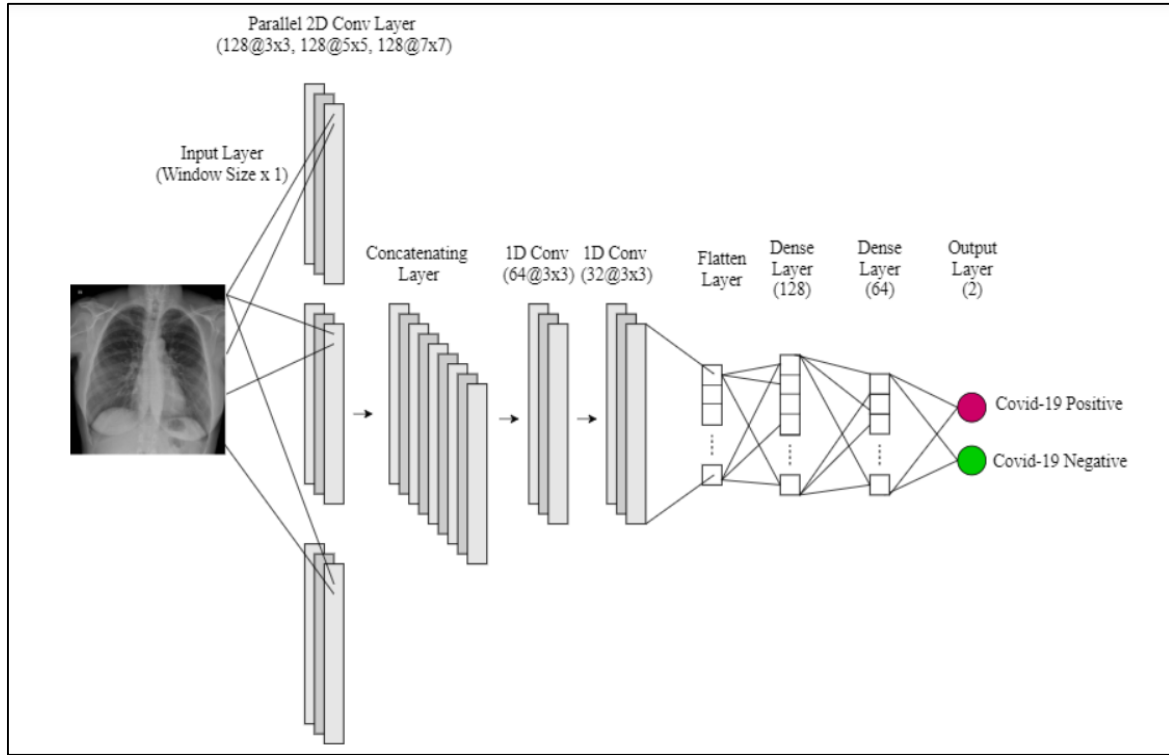
4.3.1. Paralel evrişimli sinir ağı

Paralel evrişimli sinir ağı olan yeni bir evrişimli sinir ağı modellenmiştir. Tek bir evrişimli katmandır, ancak farklı filtre boyutlarına sahiptir. Ana avantajı, farklı boyutlarda filtreleri açıkça göstermesidir. Paralel CNN mimarisi, giriş verileriyle paralel olarak kullanılan CNN dosyalarının diğerlerinden farklı olarak evrişimden sonra sonuçlar toplanır ve tamamen bağlantılı bir katmanla çıktı oluşturulur. Görüntüler katmanlardan geçirilir ve iki nöron elde edilir. Sonuç olarak olasılık hesaplaması ile covid-19 için negatif veya pozitifdir.

4.3.2. Paralel Evrişimli Sinir Ağının mimarisi

TensorFlow ve Keras paralel CNN mimarisi için kullanılır, Paralel CNN üç filtre içerir: 128 @ 3×3 , 128 @ 5×5 , 128 @ 7×7 .

Şekil 4.2 'de Mimarisi, yüzün görüntüsünü gireceğiz, sonra bu CNN katmanından geçmektedir. Daha sonra, sınıflandırma tipi bir sinir ağı olacak ve çıktı katmanında iki nöron alınacaktır. Son olarak, covid 19 pozitif veya negatif olmasına yönelik olasılık verilmektedir.



Şekil 4.2. Evrişimli sinir ağı mimarisi

4.3.3. Evrişimli katmanlar

Herhangi bir ileri beslemeli sinir ağında, herhangi bir orta katman gizli olarak adlandırılır. Bunun nedeni, girişleri ve çıkışları aktivasyon işlevi ve son evrişim tarafından maskelenmektedir. Evrişimli bir sinir ağında, gizli katmanlar, kıvrımlar gerçekleştiren katmanlar içerir. Tipik olarak bu, çarpma veya başka iç çarpım yapan bir katmanı içerir ve aktivasyon işlevi genellikle ReLU'dur. Bunu, havuzlama katmanları, tamamen bağlantılı katmanlar ve normalleştirme katmanları gibi diğer evrişim katmanları izler. Evrişimli Sinir Ağlarındaki Katmanlar aşağıdaki gibidir:

Görüntü Giriş Katmanı

Girdi katmanı ilk katman olup girdiler (veriler) ham olarak bu katmanda ağı eklenir. Tasarlanacak modelin başarılı olması için giriş boyutu burada belirtilmelidir.

Evrişimli Katman

Evrişim bu katmanda gerçekleştirilir ve görüntü algılayıcılara (algoritma) bölünür, yerel alanlar oluşturulur, bu da algılayıcıların $m \times n$ boyutunda bir matris olarak özellik haritalarına sıkıştırılmasına yol açar.

Bir boyutlu evrişim

Özellikleri kullanırken azaltmaya gidin. Örneğin, unsurlar bir boyuta ($128 * 128 * 2048$) sahipler ve bunları evriştirmek istiyorsunuz. Fakat, bu çok maliyetlidir. İlk olarak bir $1 * 1 * n$ evrişim (burada $n \ll 2048$), sonra bir evrişim ağı ile devam edilir. Ağımız, 1Dense ($64 @ 3 \times 3$) ve 1Dense ($32 @ 3 \times 3$) içerir.

Katmanı düzleştirin

Düzleştirme, havuzlanmış özellik haritası matrisinin tamamının, daha sonra işlenmek üzere sinir ağına beslenen tek bir sütuna dönüştürülmesini içerir. Ağ bu katmanda yapılan aktifleştirme ile daha hızlı öğrenir.

Yoğun katman

İsmiinde bulunduğu aşamadan alan bu katman, kendinden önceki bütün katmanlarla bağlantılı olup bunlardan girdi almaktadır. Bu nedenle, yoğun şekilde bağlı katmanlardır. Başka bir deyişle, yoğun katman tamamen bağlantılıdır.

Doğrusal Olmayan Katman

Burada özellik haritaları giriş olarak alınır ve aktivasyon fonksiyonu yardımıyla aktivasyon haritaları çıktı olarak verilir. Aktivasyon fonksiyonu genellikle sigmoid veya hiperbolik tanjant fonksiyonları olarak uygulanır.

Düzeltilme Katmanı

CNN'in en önemli bileşeni olan bu katman, hassasiyeti düşürmeden eğitimi daha hızlı yapar. Aktivasyon haritalarında eleman bazında mutlak değer işlemi gerçekleştirir.

Doğrultulmuş Doğrusal Birimler (ReLU)

ReLU, doğrusal olmayan katmanları ve düzeltme katmanlarını CNN'de birleştirir. Bu, negatif değerlerin sıfıra dönüştürüldüğü eşik işlemini yapar. Ancak ReLU, girişin boyutunu değiştirmez.

(Pooling) Katmanı

Pooling katmanına etkinleştirme haritalarının boyutunu küçültmekten sorumlu olduğundan alt örnekleme katmanı da denir. Giriş hacmine aynı uzunlukta bir filtre ve adım uygulanır. Daha az önemli veriler bu katman tarafından göz ardı edilir, bu nedenle görüntü tanıma daha küçük bir temsilde yapılır. Pooling katmanı kullanılarak parametre miktarı azaltıldığından, maliyet de azalır. Girdi, dikdörtgen pooling bölgelerine bölünür ve maksimum veya ortalama hesaplanır, bu da sonuç olarak maksimum veya ortalama verir. Max Pooling popüler olanıdır.

Çıkarma Katmanı

Bu katman, girdi katmanını belirli bir olasılıkla rastgele sıfıra ayarlar. Bu işlemden sonra farklı elemanlarda daha fazla sonuç düşürülür. Bu katman aynı zamanda aşırı uyum gösterme azaltmaya yardımcı olur. Ağın yedekli olmasını sağlar. Bu katmanda hiçbir öğrenme gerçekleşmez. Bu işlem sadece eğitim sırasında gerçekleştirilir.

Tamamen Bağlı Katman

Önceki katmanların çıktısı olan aktivasyon haritaları bu katmanda bir sınıf olasılık dağılımına dönüştürülür. FC(Fully Connected)katmanı, girdiyi bir ağırlık matrisiyle çarpar ve önyargı vektörünü ekler.

Regresyon Katmanı

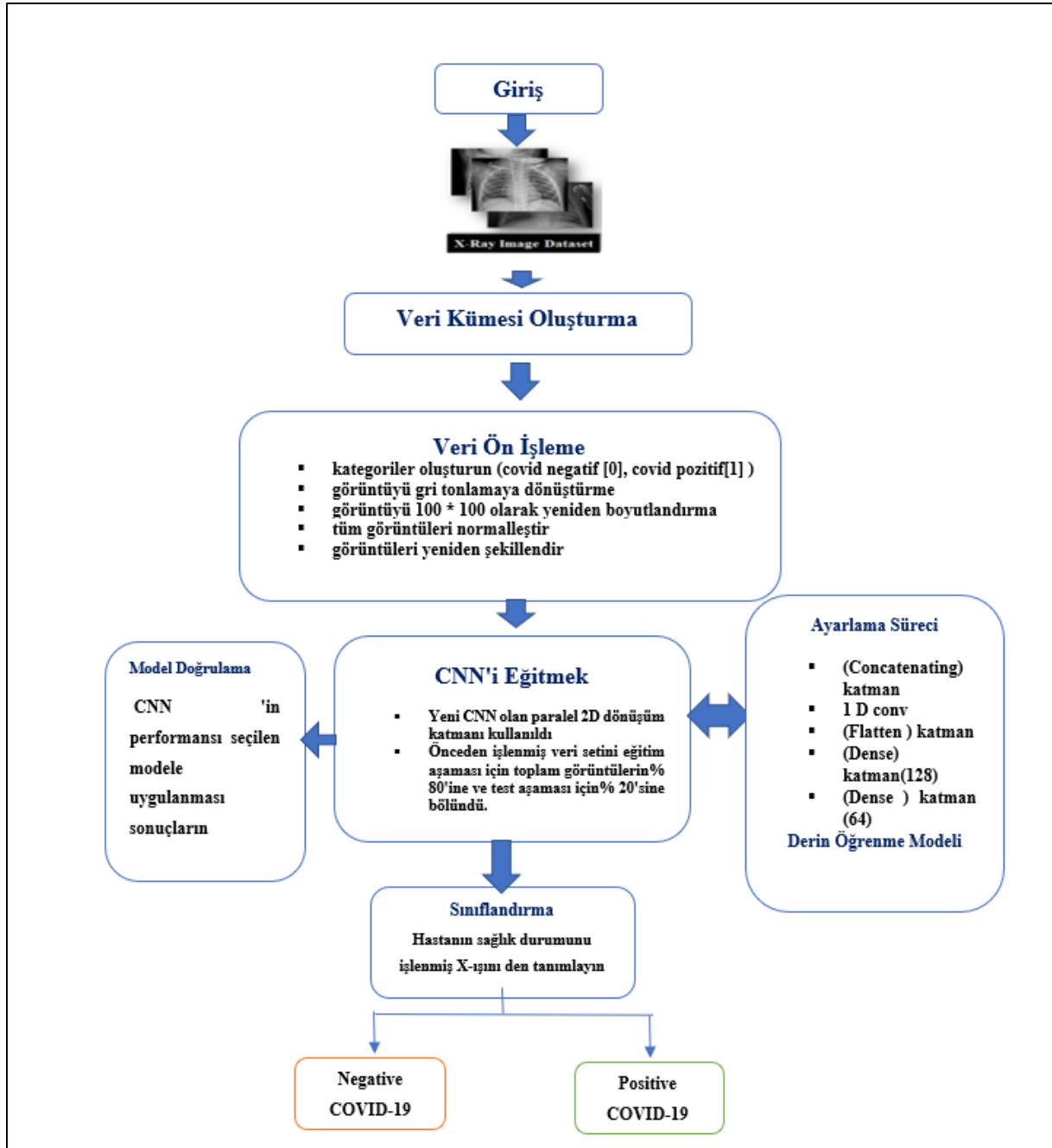
Bu katmanda ortalama hata karesinin yarısı hesaplanır. Bu katman, FC(Fully Connected) katmanını takip etmelidir.

Çıktı Katmanı

Bir CNN'deki çıktı katmanı, diğer katmanlardan gelen girdinin düzleştirildiği, tamamen bağlantılı bir katmandır. Çıktıyı ağın istediği şekilde sınıf sayısına dönüştürmesi için gönderilen model, covid-19 pozitif ve covid-19 negatif olmak üzere iki nöron içermektedir. Softmax bir aktivasyon olarak kullanılır, çünkü softmax aktivasyonu normalde ReLU, sigmoid, tanh veya başka bir aktivasyon fonksiyonu kullanmak yerine bir sinir ağındaki en son katmana uygulanır. Softmax'ın yararlı olmasının nedeni, sinir ağıımızdaki son katmanın çıktısını esasen bir olasılık dağılımına dönüştürmesidir.

Layer (type)	Output Shape	Param #
model_1 (Model)	(None, 100, 100, 384)	11008
conv2d_4 (Conv2D)	(None, 98, 98, 64)	221248
activation_1 (Activation)	(None, 98, 98, 64)	0
max_pooling2d_1 (MaxPooling2D)	(None, 49, 49, 64)	0
conv2d_5 (Conv2D)	(None, 47, 47, 32)	18464
activation_2 (Activation)	(None, 47, 47, 32)	0
max_pooling2d_2 (MaxPooling2D)	(None, 23, 23, 32)	0
flatten_1 (Flatten)	(None, 16928)	0
dropout_1 (Dropout)	(None, 16928)	0
dense_1 (Dense)	(None, 128)	2166912
dropout_2 (Dropout)	(None, 128)	0
dense_2 (Dense)	(None, 64)	8256
dropout_3 (Dropout)	(None, 64)	0
dense_3 (Dense)	(None, 2)	130
Total params: 2,426,018		
Trainable params: 2,426,018		
Non-trainable params: 0		

Şekil 4.3. CNN katmanlarımızın çıktı şekli



Şekil 4.4. X-Ray görüntülerinde COVID-19 durumunu sınıflandırmak için iş akışı.



5. SONUÇLAR

Yeni Corona virüs tüm dünyaya yayıldı. Bu nedenle, SARS-CoV-2 virüsünün yayılması, COVID-19 hastalığını dünya çapında bir salgın hastalık haline getirdi. COVID-19 küresel salgını başta sağlık olmak üzere ekonomi, eğitim ve bireylerin günlük yaşamlarının hemen her alanını ciddi anlamda etkiledi. Bununla birlikte COVID-19'un hızlı ve zamanında teşhisi, pandeminin yayılma oranını ve bulaşma riskini azaltmada çok önemli bir faktördür.

Göğüs röntgeni, son pandemide COVID-19 hastalığının test edilmesi ve teşhisinde önemli bir rol oynamaktadır. Bu röntgenlerin Derin Öğrenme gibi gelişmiş yapay zeka teknikleriyle tanıda kullanılması, doktorlar tarafından gözden kaçabilecek küçük detayların tespitinde veya insan gözü ile fark edilemeyen hastalıkların teşhisinde önemli bir fayda sağlamaktadır. Bu tezde, torasik röntgenlerden Covid-19'un otomatik teşhisi konusundaki uzmanlıkları ile ilgili olarak bilim camiası tarafından önerilen son teknoloji evrişimli sinir ağlarının etkinliği araştırılmıştır.

Veriler, eğitim için % 80 ve test için % 20 olmak üzere iki bölüme sınıflandırılmıştır. Önerilen bu yöntemin etkinliğini doğrulamak için, 500 Covid-19 Pozitif görüntü ve 1600 Covid-19 Negatif görüntü kullanarak CNN model ağını eğittik. Göğüs röntgeni taramaları işlendi ve CNN'leri eğitmek ve test etmek için kullanıldı. Bu nedenle, evrişimli sinir ağları, enfekte hastaları göğüs röntgeni görüntülerinden tespit etme yetenekleri açısından değerlendirildi. % 90 sınıflandırma doğruluğu ve % 88 doğrulama doğruluğu elde edildi.

```

Train on 551 samples, validate on 62 samples
Epoch 1/20
551/551 [=====] - 58s 106ms/step - loss: 0.6134 - accuracy: 0.7096 - val_loss: 0.4940 - val_accuracy: 0.7903
Epoch 2/20
551/551 [=====] - 62s 113ms/step - loss: 0.5435 - accuracy: 0.7822 - val_loss: 0.4422 - val_accuracy: 0.7903
Epoch 3/20
551/551 [=====] - 58s 105ms/step - loss: 0.4702 - accuracy: 0.8004 - val_loss: 0.3950 - val_accuracy: 0.7903
Epoch 4/20
551/551 [=====] - 56s 101ms/step - loss: 0.4372 - accuracy: 0.7949 - val_loss: 0.4278 - val_accuracy: 0.7903
Epoch 5/20
551/551 [=====] - 55s 101ms/step - loss: 0.4518 - accuracy: 0.8221 - val_loss: 0.4033 - val_accuracy: 0.8871
Epoch 6/20
551/551 [=====] - 56s 101ms/step - loss: 0.4149 - accuracy: 0.8548 - val_loss: 0.3607 - val_accuracy: 0.8871
Epoch 7/20
551/551 [=====] - 56s 101ms/step - loss: 0.3739 - accuracy: 0.8748 - val_loss: 0.4035 - val_accuracy: 0.8871
Epoch 8/20
551/551 [=====] - 56s 102ms/step - loss: 0.4017 - accuracy: 0.8748 - val_loss: 0.3382 - val_accuracy: 0.9032
Epoch 9/20
551/551 [=====] - 56s 102ms/step - loss: 0.3768 - accuracy: 0.8675 - val_loss: 0.3758 - val_accuracy: 0.8710
Epoch 10/20
551/551 [=====] - 56s 102ms/step - loss: 0.3383 - accuracy: 0.9002 - val_loss: 0.3462 - val_accuracy: 0.8710
Epoch 11/20
551/551 [=====] - 56s 101ms/step - loss: 0.3323 - accuracy: 0.8893 - val_loss: 0.3216 - val_accuracy: 0.8871
Epoch 12/20
551/551 [=====] - 56s 102ms/step - loss: 0.3297 - accuracy: 0.8929 - val_loss: 0.3389 - val_accuracy: 0.8710
Epoch 13/20
551/551 [=====] - 57s 104ms/step - loss: 0.2902 - accuracy: 0.8984 - val_loss: 0.3373 - val_accuracy: 0.8871
Epoch 14/20
551/551 [=====] - 56s 102ms/step - loss: 0.2896 - accuracy: 0.8947 - val_loss: 0.3844 - val_accuracy: 0.8548
Epoch 15/20
551/551 [=====] - 56s 102ms/step - loss: 0.3052 - accuracy: 0.9020 - val_loss: 0.3168 - val_accuracy: 0.8710
Epoch 16/20
551/551 [=====] - 56s 102ms/step - loss: 0.3046 - accuracy: 0.8911 - val_loss: 0.3039 - val_accuracy: 0.9032
Epoch 17/20
551/551 [=====] - 56s 102ms/step - loss: 0.2715 - accuracy: 0.9093 - val_loss: 0.5164 - val_accuracy: 0.8710
Epoch 18/20
551/551 [=====] - 56s 102ms/step - loss: 0.3023 - accuracy: 0.9038 - val_loss: 0.2857 - val_accuracy: 0.8871
Epoch 19/20
551/551 [=====] - 56s 102ms/step - loss: 0.2914 - accuracy: 0.8947 - val_loss: 0.3385 - val_accuracy: 0.8871
Epoch 20/20
551/551 [=====] - 56s 102ms/step - loss: 0.2659 - accuracy: 0.9074 - val_loss: 0.2952 - val_accuracy: 0.8871

```

Şekil 5.1. Sinir ağı eğitiminin geçmişi

Şekilde sinir ağı eğitiminin tarihi verilmektedir. Süreç içerisinde analizlerden elde edilen doğrulamalar alınır ve tüm nöral modelleri buraya kaydedilir. Buradaki koşul, doğrulama önceki dönemden daha az ise, yalnızca bu modeller kaydedilecektir.

Bu çalışmada, göğüs röntgeni radyografileri kullanılarak covid-19 ile enfekte hastaların tespiti için paralel bir CNN modeli önerilmiştir. Bu sinir ağı 20 dönem (epoch) için eğitilmiş olup eğitim geçmişi iyi performans göstermektedir. Bu nedenle önerilen model iyi olarak kabul edilmektedir.

5.1. Değerlendirme

Eğitim geçmişi incelendiğinde, doğruluğu ve doğrulama doğruluğunun artırıyor olması iyi performans gösterdiğini kanıtlamaktadır. Bununla birlikte doğrulama kaybı da azalmaktadır. Kayıp için elde edilen grafikler incelendiğinde doğruluk, aşırı uyum gösterme için çok fazla kanıt göstermiyor. Bu nedenle çalışılan model iyi bir model çünkü doğruluk kaybı epoch azalmış olup daha yüksek bir doğruluk elde etmek için daha fazla epoch eğitmek gerekmektedir. Bu amaçla çalışmada 20 epoch eğitilmiştir.

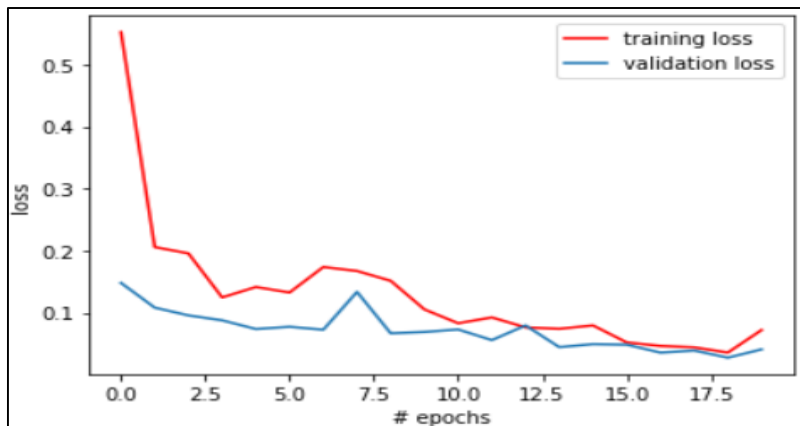
Performans Metrikleri

TP : Doğru covid 19 tespiti.

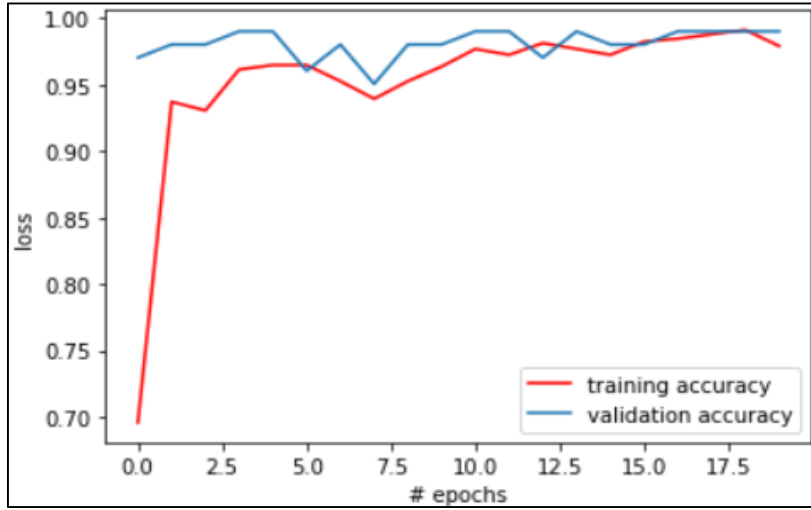
FP : Pnömoni vakası covid 19 olarak sınıflandırılmıştır.

TN : Normal veya pnömoni vakası covid -19 değil olarak sınıflandırılmıştır.

FN : Covid 19 vakaları pnömoni veya normal olarak sınıflandırılmıştır.



Şekil 5.2. Eğitim kaybı ve doğrulama kaybı



Şekil 5.3. Eğitim doğruluğu ve doğrulama doğruluğu

Performans için 5 kriter kullanılmıştır.

$$\text{Accuracy (Doğruluk)} = (TN + TP) / (TN + TP + FN + FP) \quad (5.1)$$

Eş.5.1’de. Accuracy değeri modelde doğru tahmin ettiğimiz alanların toplam veri kümesine oranı ile hesaplanmaktadır. Özellikle eşit dağılmayan unbiased veri kümelerinde model doğruluğu tek başına yeterli değildir.

$$\text{Recall (Geri Çağırma)} = TP / (TP + FN) \quad (5.2)$$

Eş.5.2’de. Duyarlılık (Recall) ise Positive olarak tahmin etmemiz gereken işlemlerin ne kadarını Positive olarak tahmin ettiğimizi gösteren bir metriktir.

$$\text{Specificity (Özgüllük)} = TN / (TN + FP) \quad (5.3)$$

Eş.5.3’te. Özgüllük (Doğru Negatif oranı), doğru bir şekilde tanımlanan negatiflerin oranını ölçer (yani, koşulu taşımayanların (etkilenmemiş) ve bu koşulu taşımadıkları doğru olarak tanımlananların oranı).

$$\text{Sensitivity (duyarlılık)} = TP / (TP + FP) \quad (5.4)$$

Eş.5.4’te. Duyarlılık (Doğru Pozitif oranı), doğru olarak tanımlanan pozitiflerin oranını ölçer (yani, bir koşulu olan (etkilenen) ve koşulu olduğu doğru olarak tanımlananların oranı).

$$F1\text{-Score} = 2 \times ((\text{Precision} \times \text{Recall}) / (\text{Precision} + \text{Recall})) \quad (5.5)$$

Eş.5.5'te, Doğruluk (Accuracy) yerine F1 Score değerinin kullanılmasının en temel sebebi eşit dağılmayan veri kümelerinde hatalı bir model seçimi yapmamaktır.

Etiket	Recall(Sensitivity)	Precision	F1-Score	Specificity
Covid	%92	%92	%92	%94
Normal	%90	%92	%88	%96
Pneumonia	%88	%90	%78	%90

Çizelge 5.1. Paralel CNN'nin sınıflandırma performansı

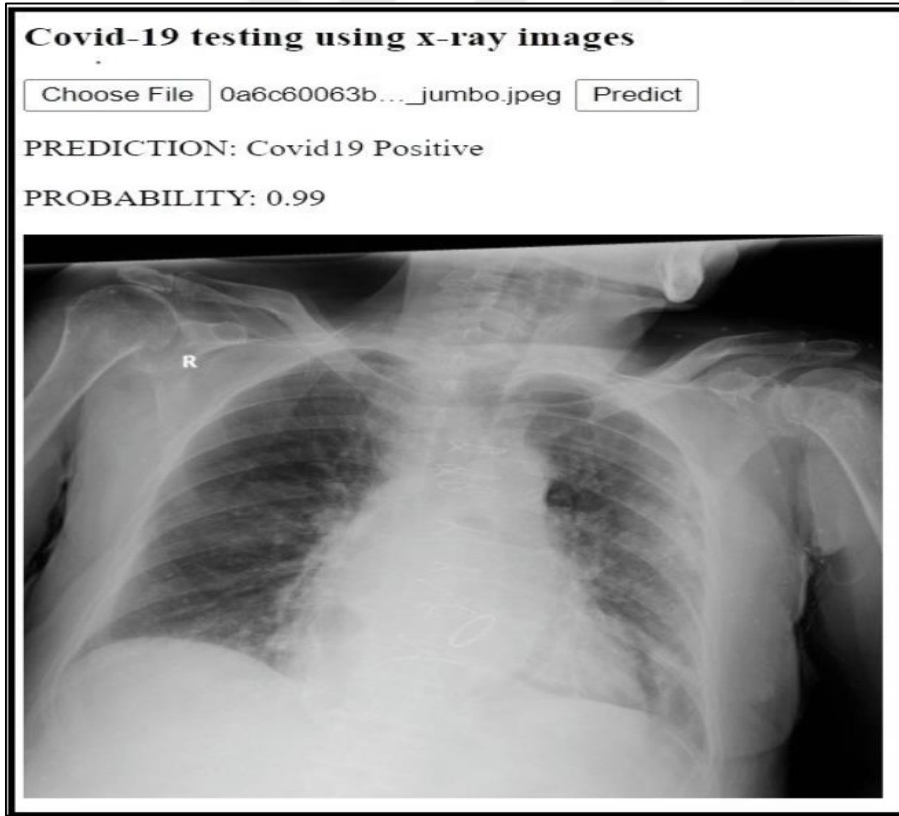
Predicted Class			True Class
Covid-19	Pneumonia	Normal	
292	1	2	
2	55	8	
2	4	92	

Çizelge 5.2 . Paralel CNN'nin karışıklık matrisi

5.2. Web Uygulaması

Web uygulaması oluşturmak için eğitimli modelimizle Python programlama dili kullanılmıştır. Html ve Javascript de çalışma için bir web sayfası oluşturma önemli bileşenler arasında yer almıştır. Python ve diğer modüllerin kullanılması ile bir web uygulamasına entegre olarak minimum doğrulama kaybına sahip en iyi tren modeli kullanılmıştır.

Grafiksel Kullanıcı Ara Yüzü'nde, yerel ana bilgisayar çalıştırılır ve tarayıcıdaki yerel adres ana bilgisayara kopyalanır. Bundan sonra ilgili klasörü açılarak istenilen resim yüklenilir. Görüntü, sınır ağına uygulanmadan önce yeniden şekillendirilecektir. Yeniden şekillendirilen görüntü, tahmin düğmesine geri dönecektir. Tahmin düğmesi olasılık yüzdesini ve tahmini sonucu verecektir.



Şekil 5.4. Covid-19'u tespit etmek için grafiksel kullanıcı arayüzü tabanlı bir araç

```

Anaconda Prompt (anaconda3) - python app.py

(base) C:\Users\ASUS> cd "C:\Users\ASUS\Desktop\my code\webapp"

(base) C:\Users\ASUS\Desktop\my code\webapp>python app.py
Using TensorFlow backend.
2021-06-04 10:59:44.151749: W tensorflow/stream_executor/platform/default/dso_loader.cc:55]
ry 'cudart64_101.dll'; dlerror: cudart64_101.dll not found
2021-06-04 10:59:44.152463: I tensorflow/stream_executor/cuda/cudart_stub.cc:29] Ignore above
ot have a GPU set up on your machine.
2021-06-04 10:59:53.241023: I tensorflow/stream_executor/platform/default/dso_loader.cc:44]
library nvcuda.dll
2021-06-04 10:59:53.817824: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1561] Found d
pciBusID: 0000:01:00.0 name: GeForce GTX 960M computeCapability: 5.0
coreClock: 1.176GHz coreCount: 5 deviceMemorySize: 4.00GiB deviceMemoryBandwidth: 74.65GiB/s
2021-06-04 10:59:53.819875: W tensorflow/stream_executor/platform/default/dso_loader.cc:55]
ry 'cudart64_101.dll'; dlerror: cudart64_101.dll not found
2021-06-04 10:59:53.821663: W tensorflow/stream_executor/platform/default/dso_loader.cc:55]
ry 'cublas64_10.dll'; dlerror: cublas64_10.dll not found
2021-06-04 10:59:53.823458: W tensorflow/stream_executor/platform/default/dso_loader.cc:55]
ry 'cufft64_10.dll'; dlerror: cufft64_10.dll not found

Use a production WSGI server instead.
* Debug mode: off
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
127.0.0.1 - - [18/May/2021 16:01:56] "[37mGET / HTTP/1.1[0m" 200 -
127.0.0.1 - - [18/May/2021 16:01:57] "[37mGET /favicon.ico HTTP/1.1[0m" 404 -
HERE
[[0.00188338 0.99367625]] 1 0.9936762452125549
127.0.0.1 - - [18/May/2021 16:02:08] "[37mPOST /predict HTTP/1.1[0m" 200 -
HERE
[[0.13803495 0.6375614 ]] 1 0.6375613808631897
127.0.0.1 - - [18/May/2021 17:53:41] "[37mPOST /predict HTTP/1.1[0m" 200 -
HERE
[[2.8021225e-06 9.9987519e-01]] 1 0.9998751878738403
127.0.0.1 - - [18/May/2021 17:53:48] "[37mPOST /predict HTTP/1.1[0m" 200 -
HERE
[[1.0000000e+00 2.2236875e-09]] 0 1.0
127.0.0.1 - - [18/May/2021 17:53:54] "[37mPOST /predict HTTP/1.1[0m" 200 -
HERE
[[1.2320636e-06 9.9991953e-01]] 1 0.9999195337295532
127.0.0.1 - - [18/May/2021 17:54:02] "[37mPOST /predict HTTP/1.1[0m" 200 -

```

Şekil 5.5. Anaconda komut isteminde web uygulamasının yürütülmesi

5.3. Tartışma

Evrişimli sinir ağı (CNN), özellikle tıbbi alanlarda derin öğrenme için uygun ve iyi bir yol olarak kabul edilmektedir. Covid 19 salgınının dünya genelinde hızlı bir şekilde tırmanması bu salgın hastalığa yakalananların en kısa sürede belirlenmesinin ve önlem alınmasının gerekliliğini artırmıştır. Bu amaç ışığında göğüs için tıbbi görüntüleme ile hızlı ve doğru bir inceleme geliştirmek amacıyla birçok çalışma yapılmıştır. Örneğin, Shiet ve diğerleri [22] Göğüs BT kullanarak görüntü sınıflandırması için enfeksiyon boyutuna duyarlı rastgele bir orman modeli sunmuşlardır. Başarılı sonuçlar elde etmelerine karşın bu modelin CT taraması X-ışınının analizlerinden daha uzun sürer ve ayrıca çok karmaşık sterilizasyon gerektirir. Bununla birlikte muayene sırasında hastalar arasında geçiş yapma dikkate alındığında bu modelin, kullanılabilirliği açısından sadece X-ray görüntülerini ilgili modele işleyerek sonuç elde edilebilen modele göre daha zayıf olduğu söylenebilir. Ayrıca, bazen yüksek kaliteli tomografi cihazları uzak bölgelerde bulunamayabilir, bu nedenle X-ışını görüntüleme en uygun görüntüleme tekniklerinden biri olarak kabul edilir.

Bu çalışmada önerilen model ise bize diğer modellere kıyasla uygun bir doğruluk sağlamış olup, bu yöntemle doktorlara yardımcı olabilecek verileri elde edebileceğimiz düşünülmektedir.

5.4. Sonuç ve Öneriler

COVID 19 virüsü ile enfekte hastaların erken teşhisi ve dolayısıyla erken tedavisi, hastalığın yayılmasını önlemek ve hastalığın üstesinden gelmek için önemlidir. Bu çalışmada, paralel bir Evrişimli Sinir Ağları (CNNs) modeli önerilmiştir.

Derin öğrenme algoritmaları uygulamaları kullanılarak, göğüs röntgeni görüntülerinin sınıflandırılması ve elde edilen sonuçlar dikkate alındığında, COVID-19' u saptamak ve ayırt etmek için göğüs röntgeni görüntülerinin PCR testlerine alternatif olarak kullanılabileceği kanıtlanmıştır. Ayrıca daha yüksek performans nedeniyle radyologların klinik uygulamalarında karar vermelerine yardımcı olacağına inanılmaktadır. COVID-19' u erken bir aşamada tespit etmek için, bu çalışma derin transfer öğrenme yöntemlerinin nasıl kullanılabileceği konusunda fikir vermektedir.

Bu çalışmada paralel Evrişimli Sinir Ağları (CNNs) modeli kullanılarak gerçekleştirilmiştir ve % 90 ve % 88 oranında başarılı sonuçlar alınmıştır.

Gelecekteki yapılacak çalışmalar için, modelin doğruluğu iyileştirmek amacıyla araştırmalar yapılması önerilmektedir. Bununla birlikte hastaların risk durumunun tahmin edilmesi ve triyaj için yararlı olabilecek hastanede kalış süresinin tahmin edilmesi, hasta popülasyon yönetimi ve kişiselleştirilmiş bakım planlamasını içeren iyileştirilmelerle model daha ideal olacaktır.





KAYNAKLAR

1. 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.
2. Abbas, A., Abdelsamea, M. M., and Gaber, M. M. (2020). Detrac: Transfer learning of class decomposed medical images in convolutional neural networks. *IEEE Access*, 8, 74901-74913.
3. İnternet: Türkiye Tabipleri Birliği. (2020). Yeni coronavirus (2019-ncov) hakkında bilgi notu. URL: https://ttb.org.tr/haber_goster.php?Guid=bffe89ae-3ea2-11ea-a1a2-6d7c2a5a4754, Son Erişim tarihi: 30 Ocak 2020.
4. Asif, S., Wenhui, Y., Jin, H., Tao, Y., and Jinhai, S. (2020). Classification of covid-19 from chest x-ray images using deep convolutional neural networks. *MedRxiv*.
5. Erdaş, Ç. B. ve Ölçer, D. (2020). Detection and differentiation of COVID-19 using deep learning approach fed by x-rays. *International Journal of Applied Mathematics Electronics and Computers*, 8(3), 97-101.
6. Narin, A., Kaya, C. ve Pamuk, Z. (2021). Automatic detection of coronavirus disease (covid-19) using x-ray images and deep convolutional neural networks. *Pattern Analysis and Applications*, 1-14.
7. Das, A. K., Ghosh, S., Thunder, S., Dutta, R., Agarwal, S., and Chakrabarti, A. (2021). Automatic COVID-19 detection from X-ray images using ensemble learning with convolutional neural network. *Pattern Analysis and Applications*, 1-14.
8. Khan, A. I., Shah, J. L., and Bhat, M. M. (2020). Coronet: A deep neural network for detection and diagnosis of COVID-19 from chest x-ray images. *Computer Methods and Programs in Biomedicine*, 196, 105581.
9. Yoo, S. H., Geng, H., Chiu, T. L., Yu, S. K., Cho, D. C., Heo, J., ... and Lee, H. (2020). Deep learning-based decision-tree classifier for COVID-19 diagnosis from chest X-ray imaging. *Frontiers in medicine*, 7, 427.
10. Wang, L., Lin, Z. Q., and Wong, A. (2020). Covid-net: A tailored deep convolutional neural network design for detection of covid-19 cases from chest x-ray images. *Scientific Reports*, 10(1), 1-12.
11. Zebin, T., and Rezvy, S. (2020). COVID-19 detection and disease progression visualization: Deep learning on chest X-rays for classification and coarse localization. *Applied Intelligence*, 1-12.

12. Brunese, L., Mercaldo, F., Reginelli, A., and Santone, A. (2020). Explainable deep learning for pulmonary disease and coronavirus COVID-19 detection from X-rays. *Computer Methods and Programs in Biomedicine*, 196, 105608.
13. Toğaçar, M., Ergen, B., ve 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.
14. Al Rahhal, M. M., Bazi, Y., AlHichri, H., Alajlan, N., Melgani, F., and Yager, R. R. (2016). Deep learning approach for active classification of electrocardiogram signals. *Information Sciences*, 345, 340-354.
15. Doğan, F. ve Türkoğlu, İ. (2019). Derin Öğrenme Modelleri ve Uygulama Alanlarına İlişkin Bir Derleme. *Dicle Üniversitesi Mühendislik Fakültesi Mühendislik Dergisi*, 10(2), 409-445.
16. Mitchell, R., Michalski, J., and Carbonell, T. (2013). *An Artificial intelligence approach*. Berlin: Springer.
17. Plis, S. M., Hjelm, D. R., Salakhutdinov, R., Allen, E. A., Bockholt, H. J., Long, J. D., ... and Calhoun, V. D. (2014). Deep learning for neuroimaging: A validation study. *Frontiers in neuroscience*, 8, 229.
18. Park, S. H., and Han, K., (2018). Methodologic guide for evaluating clinical performance and effect of artificial intelligence technology for medical diagnosis and prediction. *Radiology*, 286(3), 800-809.
19. Özkan, İ. N. İ. K. ve Ülker, E. (2017). Derin öğrenme ve görüntü analizinde kullanılan derin öğrenme modelleri. *Gaziosmanpaşa Bilimsel Araştırma Dergisi*, 6(3), 85-104.
20. Shen, D., Wu, G., and Suk, H. I. (2017). Deep learning in medical image analysis. *Annual review of biomedical engineering*, 19, 221-248.
21. Deng, L., and Yu, D. (2014). Deep learning: Methods and applications. *Foundations and trends in signal processing*, 7(3-4), 197-387.
22. Shi, F., Xia, L., Shan, F., Wu, D., Wei, Y., Yuan, H., ... and Shen, D. (2020). Large-scale screening of covid-19 from community acquired pneumonia using infection size-aware classification. *arXiv preprint arXiv*, 2003.09860.



EKLER

EK-1. Veri Kümesinin Toplanması Kodu

```

In [1]: import pandas as pd
import os
import numpy as np

datapath1='covid-chestxray-dataset-master'
dataset_path='dataset'

categories=os.listdir(dataset_path)
print(categories)

dataset=pd.read_csv(os.path.join(datapath1,'metadata.csv'))
findings=dataset['finding']
image_names=dataset['filename']

['Covid19 Negative', 'Covid19 Positive']

In [2]: positives_index=np.concatenate((np.where(findings=='COVID-19')[0],np.where(findings=='SARS')[0]))
positive_image_names=image_names[positives_index]

In [3]: import cv2

for positive_image_name in positive_image_names:
    image=cv2.imread(os.path.join(datapath1,'images',positive_image_name))
    try:
        cv2.imwrite(os.path.join(dataset_path,categories[1],positive_image_name),image)
    except Exception as e:
        print(e)

In [4]: datapath2='562468_1022626_bundle_archive'

dataset=pd.read_csv(os.path.join(datapath2,'Chest_xray_Corona_Metadata.csv'))
findings=dataset['Label']
image_names=dataset['X_ray_image_name']

In [5]: negative_index=np.where(findings=='Normal')[0]

```

Şekil 1.1. Veri Kümesinin Toplanması Kodları

EK-1. (devam) Veri Kümesinin Toplanması Kodu

```

In [6]: negative_image_names=image_names[negative_index]

In [7]: for negative_image_name in negative_image_names:
        image=cv2.imread(os.path.join(datapath2,'images',negative_image_name))
        try:
            cv2.imwrite(os.path.join(dataset_path,categories[0],negative_image_name),image)
        except Exception as e:
            print(e)

ror: (-215:Assertion failed) !_img.empty() in function 'cv::imwrite'

OpenCV(4.4.0) C:\Users\appveyor\AppData\Local\Temp\1\pip-req-build-h4wtvo23\opencv\modules\imgcodecs\src\loadsave.cpp:738: er
ror: (-215:Assertion failed) !_img.empty() in function 'cv::imwrite'

OpenCV(4.4.0) C:\Users\appveyor\AppData\Local\Temp\1\pip-req-build-h4wtvo23\opencv\modules\imgcodecs\src\loadsave.cpp:738: er
ror: (-215:Assertion failed) !_img.empty() in function 'cv::imwrite'

OpenCV(4.4.0) C:\Users\appveyor\AppData\Local\Temp\1\pip-req-build-h4wtvo23\opencv\modules\imgcodecs\src\loadsave.cpp:738: er
ror: (-215:Assertion failed) !_img.empty() in function 'cv::imwrite'

OpenCV(4.4.0) C:\Users\appveyor\AppData\Local\Temp\1\pip-req-build-h4wtvo23\opencv\modules\imgcodecs\src\loadsave.cpp:738: er
ror: (-215:Assertion failed) !_img.empty() in function 'cv::imwrite'

OpenCV(4.4.0) C:\Users\appveyor\AppData\Local\Temp\1\pip-req-build-h4wtvo23\opencv\modules\imgcodecs\src\loadsave.cpp:738: er
ror: (-215:Assertion failed) !_img.empty() in function 'cv::imwrite'

OpenCV(4.4.0) C:\Users\appveyor\AppData\Local\Temp\1\pip-req-build-h4wtvo23\opencv\modules\imgcodecs\src\loadsave.cpp:738: er
ror: (-215:Assertion failed) !_img.empty() in function 'cv::imwrite'

In [8]: negative_image_names.shape

Out[8]: (1576,)

```

Şekil 1.1. (devam)Veri Kümesinin Toplanması Kodları

EK-2. Görüntü Ön İşleme Kodu

```

In [1]: import cv2,os

data_path='dataset'
categories=os.listdir(data_path)
labels=[i for i in range(len(categories))]

label_dict=dict(zip(categories,labels)) #empty dictionary

print(label_dict)
print(categories)
print(labels)

{'Covid19 Negative': 0, 'Covid19 Positive': 1}
['Covid19 Negative', 'Covid19 Positive']
[0, 1]

In [2]: img_size=100
data=[]
target=[]

for category in categories:
    folder_path=os.path.join(data_path,category)
    img_names=os.listdir(folder_path)

    for img_name in img_names:
        img_path=os.path.join(folder_path,img_name)
        img=cv2.imread(img_path)

        try:
            gray=cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
            #Covertig the image into gray scale
            resized=cv2.resize(gray,(img_size,img_size))
            #resizing the gray scale into 100x100, since we need a fixed common size for all the images in the dataset
            data.append(resized)
            target.append(label_dict[category])
            #appending the image and the Label(categorized) into the List (dataset)

        except Exception as e:
            print('Exception:',e)
            #if any exception rasied, the exception will be printed here. And pass to the next image

Exception: OpenCV(4.4.0) C:\Users\appveyor\AppData\Local\Temp\1\pip-req-build-h4wtvo23\opencv\modules\imgproc\src\color.cpp:1
82: error: (-215:Assertion failed) !_src.empty() in function 'cv::cvtColor'

```

Şekil 2.1. Görüntü Ön İşleme Kodu

EK-2. (devam) Görüntü Ön İşleme Kodu

```
Exception: OpenCV(4.4.0) C:\Users\appveyor\AppData\Local\Temp\1\pip-req-build-h4wtvo23\opencv\modules\imgproc\src\color.cpp:182: error: (-215:Assertion failed) !_src.empty() in function 'cv::cvtColor'

Exception: OpenCV(4.4.0) C:\Users\appveyor\AppData\Local\Temp\1\pip-req-build-h4wtvo23\opencv\modules\imgproc\src\color.cpp:182: error: (-215:Assertion failed) !_src.empty() in function 'cv::cvtColor'

Exception: OpenCV(4.4.0) C:\Users\appveyor\AppData\Local\Temp\1\pip-req-build-h4wtvo23\opencv\modules\imgproc\src\color.cpp:182: error: (-215:Assertion failed) !_src.empty() in function 'cv::cvtColor'

Exception: OpenCV(4.4.0) C:\Users\appveyor\AppData\Local\Temp\1\pip-req-build-h4wtvo23\opencv\modules\imgproc\src\color.cpp:182: error: (-215:Assertion failed) !_src.empty() in function 'cv::cvtColor'

Exception: OpenCV(4.4.0) C:\Users\appveyor\AppData\Local\Temp\1\pip-req-build-h4wtvo23\opencv\modules\imgproc\src\color.cpp:182: error: (-215:Assertion failed) !_src.empty() in function 'cv::cvtColor'

Exception: OpenCV(4.4.0) C:\Users\appveyor\AppData\Local\Temp\1\pip-req-build-h4wtvo23\opencv\modules\imgproc\src\color.cpp:182: error: (-215:Assertion failed) !_src.empty() in function 'cv::cvtColor'
```

```
In [3]: import numpy as np

data=np.array(data)/255.0
data=np.reshape(data,(data.shape[0],img_size,img_size,1))
target=np.array(target)

from keras.utils import np_utils

new_target=np_utils.to_categorical(target)

Using TensorFlow backend.
```

```
In [4]: np.save('data',data)
np.save('target',new_target)
```

Şekil 2.1. (devam) Görüntü Ön İşleme Kodu

EK-3. Evrişimli Sinir Ağının (CNN) Eğitimi kodu

```

In [1]: import numpy as np

data=np.load('data.npy')
target=np.load('target.npy')

In [2]: from keras.models import Sequential,Model
from keras.layers import Dense, Dropout, Flatten
from keras.layers import Conv2D,Activation,MaxPooling2D
from keras.utils import normalize
from keras.layers import Concatenate
from keras import Input
from keras.callbacks import ModelCheckpoint

input_shape=data.shape[1:] #50,50,1
inp=Input(shape=input_shape)
convs=[]

parallel_kernels=[3,5,7]

for k in range(len(parallel_kernels)):

    conv = Conv2D(128, parallel_kernels[k],border_mode='same',activation='relu',input_shape=input_shape,strides=1)(inp)
    convs.append(conv)

out = Concatenate()(convs)
conv_model = Model(input=inp, output=out)

model = Sequential()
model.add(conv_model)

model.add(Conv2D(64,(3,3)))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2,2)))

model.add(Conv2D(32,(3,3)))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2,2)))

model.add(Flatten())
model.add(Dropout(0.5))
model.add(Dense(128,activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(64,activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(2,input_dim=128,activation='softmax'))
model.compile(loss='categorical_crossentropy',optimizer='adam',metrics=['accuracy'])

model.summary()

Using TensorFlow backend.
<ipython-input-2-e53ddc83638a>:17: UserWarning: Update your `Conv2D` call to the Keras 2 API: `Conv2D(128, 3, activation="relu", input_shape=(100, 100,..., strides=1, padding="same"))`
conv = Conv2D(128, parrallel_kernels[k],border_mode='same',activation='relu',input_shape=input_shape,strides=1)(inp)

```

Şekil 3.1. Evrişimli Sinir Ağının (CNN) Eğitimi kodu

EK-3. (devam) Evrişimli Sinir Ağının (CNN) Eğitimi kodu

```

Model: "sequential_1"

```

Layer (type)	Output Shape	Param #
model_1 (Model)	(None, 100, 100, 384)	11008
conv2d_4 (Conv2D)	(None, 98, 98, 64)	221248
activation_1 (Activation)	(None, 98, 98, 64)	0
max_pooling2d_1 (MaxPooling2D)	(None, 49, 49, 64)	0
conv2d_5 (Conv2D)	(None, 47, 47, 32)	18464
activation_2 (Activation)	(None, 47, 47, 32)	0
max_pooling2d_2 (MaxPooling2D)	(None, 23, 23, 32)	0
flatten_1 (Flatten)	(None, 16928)	0
dropout_1 (Dropout)	(None, 16928)	0
dense_1 (Dense)	(None, 128)	2166912
dropout_2 (Dropout)	(None, 128)	0
dense_2 (Dense)	(None, 64)	8256
dropout_3 (Dropout)	(None, 64)	0
dense_3 (Dense)	(None, 2)	130

```

Total params: 2,426,018
Trainable params: 2,426,018
Non-trainable params: 0

```

```

<ipython-input-2-e53ddc83638a>:17: UserWarning: Update your `Conv2D` call to the Keras 2 API: `Conv2D(128, 5, activation="relu", input_shape=(100, 100,..., strides=1, padding="same"))`
  conv = Conv2D(128, parallel_kernels[k],border_mode='same',activation='relu',input_shape=input_shape,strides=1)(inp)
<ipython-input-2-e53ddc83638a>:17: UserWarning: Update your `Conv2D` call to the Keras 2 API: `Conv2D(128, 7, activation="relu", input_shape=(100, 100,..., strides=1, padding="same"))`
  conv = Conv2D(128, parallel_kernels[k],border_mode='same',activation='relu',input_shape=input_shape,strides=1)(inp)
<ipython-input-2-e53ddc83638a>:21: UserWarning: Update your `Model` call to the Keras 2 API: `Model(inputs=Tensor("in...", output=Tensor("co..."))`
  conv_model = Model(input=inp, output=out)

```

```

In [3]: from sklearn.model_selection import train_test_split

train_data,test_data,train_target,test_target=train_test_split(data,target,test_size=0.1)

```

```

In [4]: checkpoint = ModelCheckpoint('model-{epoch:03d}.model',monitor='val_loss',verbose=0,save_best_only=True,mode='auto')
history=model.fit(train_data,train_target,epochs=20,callbacks=[checkpoint],validation_split=0.1)

Train on 551 samples, validate on 62 samples

```

```

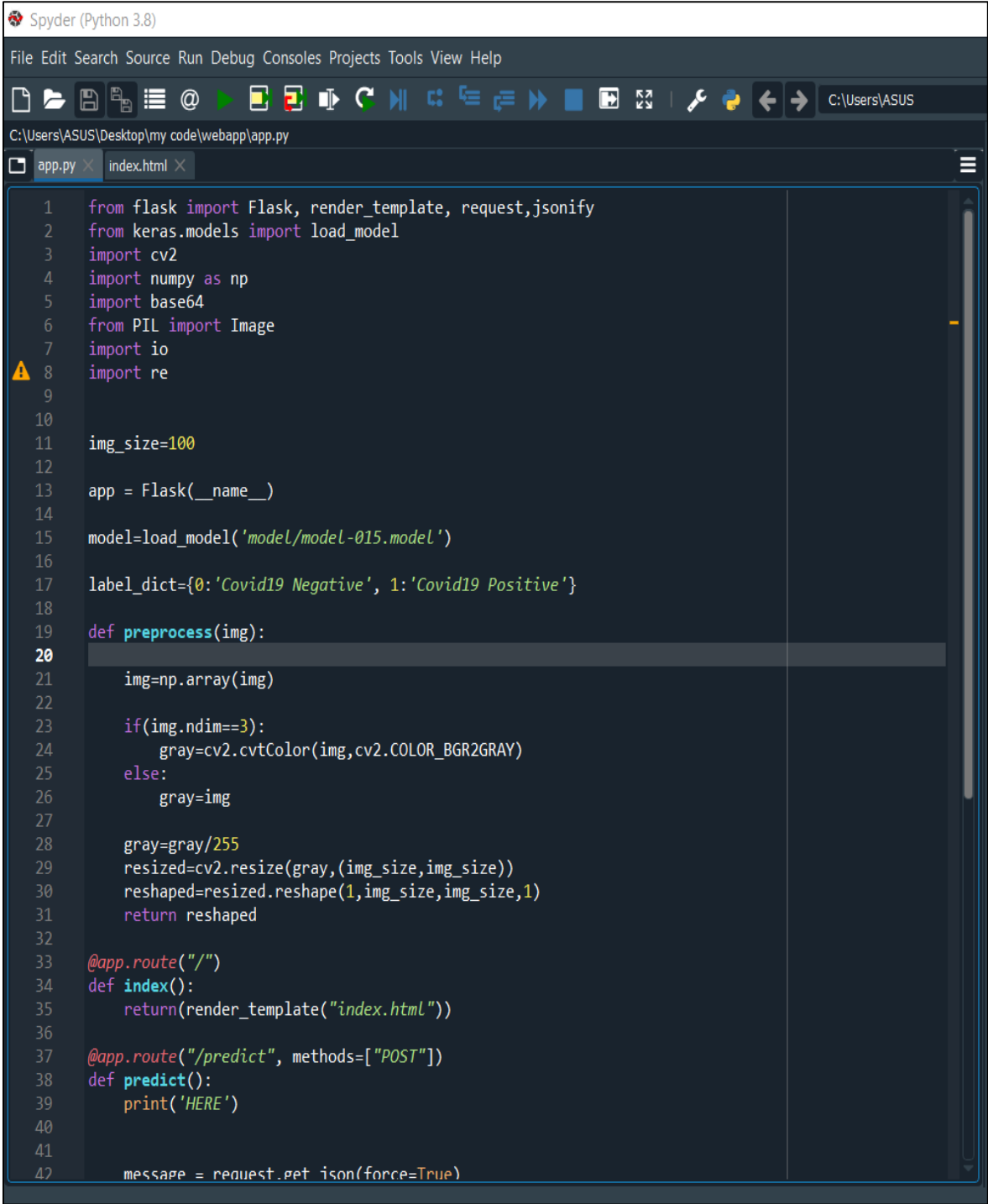
In [7]: print(model.evaluate(test_data,test_target))

69/69 [=====] - 2s 27ms/step
[0.3691935357840165, 0.8985507488250732]

```

Şekil 3.1. (devam) Evrişimli Sinir Ağının (CNN) Eğitimi kodu

EK-4. Modeli web'e bağlamak için kodumuz



```

1  from flask import Flask, render_template, request, jsonify
2  from keras.models import load_model
3  import cv2
4  import numpy as np
5  import base64
6  from PIL import Image
7  import io
8  import re
9
10
11  img_size=100
12
13  app = Flask(__name__)
14
15  model=load_model('model/model-015.model')
16
17  label_dict={0: 'Covid19 Negative', 1: 'Covid19 Positive'}
18
19  def preprocess(img):
20
21      img=np.array(img)
22
23      if(img.ndim==3):
24          gray=cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
25      else:
26          gray=img
27
28      gray=gray/255
29      resized=cv2.resize(gray,(img_size,img_size))
30      reshaped=resized.reshape(1,img_size,img_size,1)
31      return reshaped
32
33  @app.route("/")
34  def index():
35      return(render_template("index.html"))
36
37  @app.route("/predict", methods=["POST"])
38  def predict():
39      print('HERE')
40
41
42      message = request.get_json(force=True)

```

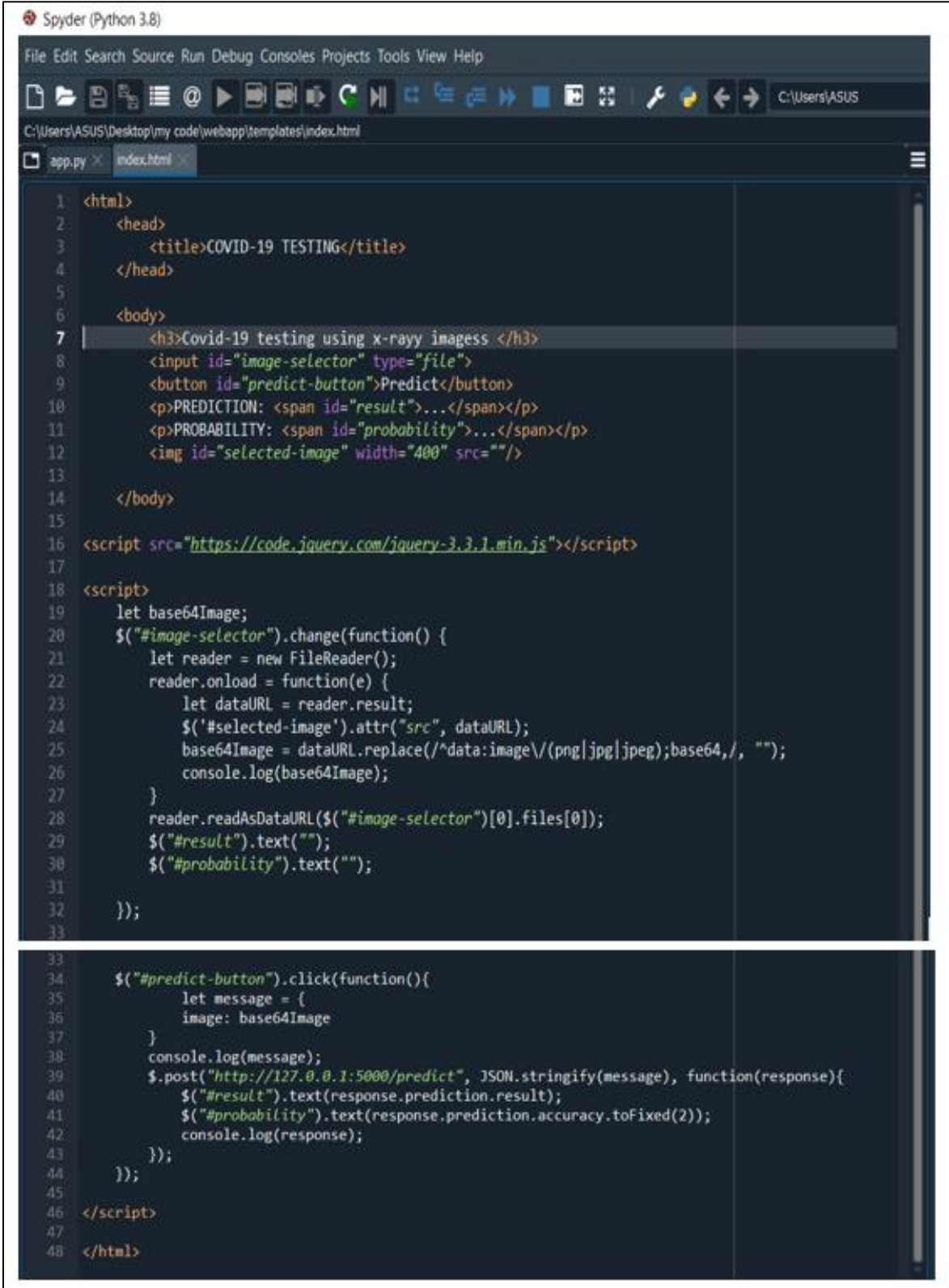
Şekil 4.1. Modeli web'e bağlamak için kodumuz

EK-4. (devam) Modeli web'e bağlamak için kodumuz

```
42     message = request.get_json(force=True)
43     encoded = message['image']
44     decoded = base64.b64decode(encoded)
45     dataBytesIO=io.BytesIO(decoded)
46     dataBytesIO.seek(0)
47     image = Image.open(dataBytesIO)
48
49     test_image=preprocess(image)
50
51     prediction = model.predict(test_image)
52     result=np.argmax(prediction,axis=1)[0]
53     accuracy=float(np.max(prediction,axis=1)[0])
54
55     label=label_dict[result]
56
57     print(prediction,result,accuracy)
58
59     response = {'prediction': {'result': label,'accuracy': accuracy}}
60
61     return jsonify(response)
62
63 #app.run(debug=True)
64 app.run(host="127.0.0.1",port=5000,threaded=False)
65
66 #<img src="" id="img" crossorigin="anonymous" width="400" alt="Image preview...">
```

Şekil 4.1. (devam) Modeli web'e bağlamak için kodumuz

EK-5. HTML ve JavaScript ile web sitesinin ön yüzü



```

Spyder (Python 3.8)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\ASUS\Desktop\my code\webapp\templates\index.html
app.py index.html
1 <html>
2 <head>
3 <title>COVID-19 TESTING</title>
4 </head>
5
6 <body>
7 <h3>Covid-19 testing using x-ray images</h3>
8 <input id="image-selector" type="file">
9 <button id="predict-button">Predict</button>
10 <p>PREDICTION: <span id="result">...</span></p>
11 <p>PROBABILITY: <span id="probability">...</span></p>
12 <img id="selected-image" width="400" src="">
13
14 </body>
15
16 <script src="https://code.jquery.com/jquery-3.3.1.min.js"></script>
17
18 <script>
19 let base64Image;
20 $("#image-selector").change(function() {
21 let reader = new FileReader();
22 reader.onload = function(e) {
23 let dataURL = reader.result;
24 $("#selected-image").attr("src", dataURL);
25 base64Image = dataURL.replace(/^data:image\/(png|jpg|jpeg);base64/, "");
26 console.log(base64Image);
27 }
28 reader.readAsDataURL($("#image-selector")[0].files[0]);
29 $("#result").text("");
30 $("#probability").text("");
31
32 });
33
34 $("#predict-button").click(function(){
35 let message = {
36 image: base64Image
37 }
38 console.log(message);
39 $.post("http://127.0.0.1:5000/predict", JSON.stringify(message), function(response){
40 $("#result").text(response.prediction.result);
41 $("#probability").text(response.prediction.accuracy.toFixed(2));
42 console.log(response);
43 });
44 });
45
46 </script>
47
48 </html>

```

Şekil 5.1. HTML ve JavaScript ile web sitesinin ön yüzü

ÖZGEÇMİŞ

Kişisel Bilgiler

Soyadı, adı : NAMRİĞ HUSSAİN SULİMAN, Fedal

Uyruğu : Sudanlı

Doğum tarihi ve yeri

Medeni hali

e-mail

Eğitim

Derece	Eğitim Birimi	Mezuniyet Tarihi
Yüksek lisans	Gazi Üniversitesi / Bilgisayar Bilimleri	Devam ediyor
Lisans	O.islamic university/ Bilgisayar Bilimleri	2014
Lise	Ombada girls school	2010

İş Deneyimi

Yıl	Yer	Görev
2014-Halen	O.islamic university	Araştırma Görevlisi
2016-2017	Al ribat university	Programmer

Yabancı Dil

İngilizce

Yayınlar

Fedal, N., Dursun. M. (2021). Covid-19 Detection By X-Ray Images and Deep Learning. International Conference on Mathematics and Artificial Intelligence (ICMAI). March ,2021. Chengdu, China.



GAZİLİ OLMAK AYRICALIKTIR..

