



T.C.

**BURSA TEKNİK ÜNİVERSİTESİ
MÜHENDİSLİK VE DOĞA BİLİMLERİ FAKÜLTESİ
MEKATRONİK MÜHENDİSLİĞİ**

BİTKİ YAPRAKLARINDA DERİN ÖĞRENME İLE HASTALIK TESPİTİ

LİSANS TEZİ

Semih KUŞCU

MEKATRONİK MÜHENDİSLİĞİ

HAZİRAN

2021

“Bu bitirme alışmasındaki tüm bilgilerin akademik kurallara ve etik ilkelere uygun olarak elde edildiğini ve sunulduğunu; ayrıca bu kuralların ve ilkelerin gerektirdiğı şekilde, bu alışmadan kaynaklanmayan bütün atıfları yaptığımı beyan ederim

Semih KUŞCU

ÖZET

Bitirme Çalışması

BİTKİ YAPRAKLARINDA DERİN ÖĞRENME İLE HASTALIK TESPİTİ

Semih KUŞCU

Bursa Teknik Üniversitesi

Mühendislik ve Doğa Bilimleri Fakültesi

Mekatronik Mühendisliği

Bitirme Çalışması Danışmanı:

Dr.Öğr.Üyesi Ahmet Remzi Özcan

Haziran 2021

Yapay zeka ve makine öğrenmesi çalışmalarının gün geçtikçe ilerlemesiyle çeşitli uygulama alanları ortaya çıkmıştır. Uygulanan derin öğrenme metotları ile bitkilerde bulunan hastalıklar başarıyla tespit edilebilmiştir. Bu tez çalışmasında derin öğrenme yapılarından biri olan Evrişimli sinir ağları kullanılarak en baştan bir model oluşturulmuştur. Veri seti olarak çeşitli bitki yaprakları bulunan 13 hastalık çeşidine sahip bir veri seti kullanılmıştır. Resimlerin boyutları 224x224 pikselidir ve renkli formattadır. Modelin ilk katmanında 5x5 boyutunda bir filtre matrisi ve 64 filtre kullanılmıştır. Filtre sayısı ilerleyen katmanlarda arttırılmasına karşın filtrenin matris boyutları 3x3'e düşürülmüştür. Aktivasyon fonksiyonu olarak ReLU tercih edilmiştir. Maksimum havuzlama katmanı ile resimlerin boyutları küçültülüp sonraki evrişim katmanına aktarılmıştır. Tam bağlı katmanda olasılıklar ortaya çıktıktan sonra eğitime geçilmiştir. Eğitim aşamasında optimizasyon fonksiyonu olarak Adam kullanılmıştır. Eğitim aşaması bittikten sonra modelin test aşamasına geçilmiştir. Test aşamasında ortaya çıkan değerler değerlendirilip bu tez hazırlanmıştır.

Anahtar Kelimeler : Derin Öğrenme, Yapay Sinir Ağları, Evrişimli Sinir Ağları

TEŞEKKÜR

Proje ve tez aşamasında değerli fikirlerini ve yardımlarını esirgemeyerek bana zamanını ve sabrını ayıran, karşılaştığım sorunlar karşısında deneyimleri ve bilgisiyle bana yol gösteren Sayın Dr. Öğr. Ahmet Remzi ÖZCAN' a teşekkürlerimi sunarım.

Her şeyden öte, eğitim hayatım boyunca maddi ve manevi desteğini esirgemeyen sevgili aileme, hayatımda bana yol gösteren öğretmenlerime ve arkadaşlarıma en derin saygılarımı ve sevgilerimi sunarım.

Semih KUŞCU

İÇİNDEKİLER

	<u>Satır</u>
ÖZET	iii
TEŞEKKÜR	iv
İÇİNDEKİLER.....	v
ŞEKİLLER DİZİNİ	vii
ÇİZELGELER DİZİNİ	ix
KISALTMALAR	xi
1. GİRİŞ	1
1.1 PROJENİN AMACI	2
1.2 LİTERATÜR TARAMASI	2
2. TEMEL BİLGİLER.....	5
2.1 ÖĞRENMENİN TANIMI	5
2.2 YAPAY ZEKA.....	5
2.3 MAKİNE ÖĞRENMESİ	6
2.4 YAPAY SİNİR AĞLARI	7
2.5 DERİN ÖĞRENME	10
2.6 EVRİŞİMLİ SİNİR AĞLARI	12
2.6.1 Evrişim Katmanı.....	14
2.6.2 Aktivasyon Katmanı	15
2.6.3 Havuzlama Katmanı	16
2.6.4 Düzleştirme İşlemi.....	17
2.6.5 Tam Bağlı Katman	17
2.6.6 Dropout Katmanı	18

3. MATERYAL VE YÖNTEM	19
3.1 KULLANILAN YAZILIMLAR	19
3.1.1 Python Programlama Dili	19
3.1.2 Google Colaboratory	19
3.1.3 Anaconda	19
3.2 KULLANILAN KÜTÜPHANELER	20
3.3 KULLANILAN VERİ SETİ	21
3.4 KULLANILAN YÖNTEM	22
3.4.1 Verilerin İçerik Aktarılması ve görüntülerin işlenmesi	22
3.4.2 Modelin Oluşturulması	23
3.4.2.1 Önemli parametrelerin ayarlanması	25
3.4.3 Modelin Görselleştirilmesi	27
4. TEST VE DEĞERLENDİRME	30
4.1 TEST EDİLEN RESİMLERİN DEĞERLENDİRİLMESİ	30
4.2 TEST EDİLEN HER BİR RESİM İÇİN ANALİZ DEĞERLERİ	32
5. SONUÇ VE ÖNERİLER	37
6. KAYNAKÇA	39
7. ÖZGEÇMİŞ	42

ŞEKİLLER DİZİNİ

<u>No</u>	<u>Sayfa</u>
Şekil 1. (A) İnsan nöronu, (B) Yapay nöron, (C) Biyolojik Sinaps ,(D) YSA Sinapsları	7
Şekil 2. Yapay Sinir Hücresinin gösterimi.	8
Şekil 3. İleri Beslemeli Yapay Sinir Ağı.	9
Şekil 4. Geri Beslemeli Yapay Sinir Ağı.....	10
Şekil 5. Derin Öğrenme ile Eski Öğrenme Algoritmaları arasındaki ilişki.	10
Şekil 6. Evrişimli Sinir Ağlarının algoritmasının çalışma prensibi.	11
Şekil 7. Evrişimli Sinir Ağlarının temel katmanları	12
Şekil 8. Evrişimsel Sinir Ağlarının genel mimarisi	13
Şekil 9. Evrişim işleminin gösterimi	14
Şekil 10. Evrişim katmanında dolgu işleminin gösterimi.....	15
Şekil 11. ReLU aktivasyon fonksiyonu.....	16
Şekil 12. Maksimum havuzlama işleminin uygulanması.	17
Şekil 13. Düzleştirme işleminin gösterimi.	17
Şekil 14. Evrişimli Sinir Ağlarında Dropout Yapısı.....	18
Şekil 15. Veri setinde bulunan örnek bitki yaprakları	21
Şekil 16. Veri setini işlemek için kullanılan kütüphaneler	22
Şekil 17. Keras ile görüntülerin işlenmesi	23
Şekil 18. Keras ile görüntülerin boyutlandırılması.....	23
Şekil 19. Modelin birinci katmanının oluşturulması	23
Şekil 20. Modelin genel gösterimi.....	24
Şekil 21. Modelin derlenmesi ve eğitilmesi	25
Şekil 22. Oluşturulan modelin her katmandaki parametre sayıları.....	26
Şekil 23. Eğitim sonucunda modelin görselleştirilmesi.	27
Şekil 24. Modelin genel kurgusu.....	28
Şekil 25. Eğitim sonrasında oluşan doğruluk grafiği.....	29
Şekil 26. Eğitim sonucunda oluşan kayıpların grafiği.....	29
Şekil 27. Test edilen bitki örneği.....	32
Şekil 28. Test edilen bir domates bitkisi örneği.	34
Şekil 29. Yanlış sınıflandırılan bir mısır örneği.	35

ÇİZELGELER DİZİNİ

<u>No</u>	<u>Sayfa</u>
Çizelge 1.	Biyolojik Sinir Sistemin yapay sinir sistemindeki karşılıklar..... 8
Çizelge 2.	Kullanılan veri setinin bilgileri.....21
Çizelge 3.	Test edilen sınıfların ayrıntılı olarak gösterimi.....30
Çizelge 4.	Sınıflandırmanın tahmin edilen ve gerçekleşen değer matrisi31
Çizelge 5.	Çilek bitkisinin hastalık oranı.....33
Çizelge 6.	Domates bitkisinin analiz değerleri.....34
Çizelge 7.	Mısır bitkisinin analiz değerleri.....35
Çizelge 8.	Test edilen bitkilerinin analiz tablosu36

KISALTMALAR

AI	: Artificial Intelligence
CNN	: Convolutional Neural Network
ESA	: Evriřimli Sinir Ađları
GPU	: Grafik Process Unit
İHA	: İnsansız Hava Aracı
R-CNN	: Region Convolutional Neural Network
ReLU	: Rectified Linear Unit
RGB	: Red (Kırmızı), Green (Yeřil), Blue (Mavi)
SDD	: Single Shot MultiBox Detector
YSA	: Yapay Sinir Ađları

1. GİRİŞ

Tarım günümüze kadar insan ve diğer canlıların hayatında önemli yere sahiptir. Geçim kaynağı bitkilerden elde edilecek ürünler olan ülkelerin, kalkınması tarımsal kaynaklara bağlıdır. Ürünlerin ithal ya da ihraç edilmesi ülke ekonomisinin gelişmesini sağlar. Dünyada yaşanan olumsuzluklar tarımı ve doğrudan canlıları da etkilemektedir.

Dünya nüfusunun artması, yaşanan kuraklıklar ve kıtlıklar tarımsal gıdalara olan ihtiyacı giderek arttırmıştır. Bu zamanlarda topraktan alınacak verim ve kalitenin önemi bir kat daha değerlendirilmiştir. Ayrıca alınacak verim ve kalite zamana bağlıdır. Bu açıdan ürünlerin verimli olabilmesi ve istenen kaliteye ulaşması bitkisel hastalıklardan korunması ile mümkündür. Bitkide oluşan hastalığın zamanında tespiti ve tedavi süreci tarımsal süreçlerde önemlidir.

Bitkilerdeki hastalıkların tespit edilmesi üreticiler için zaman ve maliyet açısından sorun çıkartmaktadır. Üreticilerin zarardan kurtulması ve zaman kazanması için hızlı, güvenilir, hesaplı çözümler gerekmektedir. Günümüzde artan teknolojik gelişmeler ile bitkide hastalık olup olmadığı eğer hastalık var ise elde edilen görüntü üzerinden sınıflandırma, nesne tespiti ya da görüntü işleme algoritmaları ile yüksek başarıda sonuçlar elde etmek mümkündür. İnsanların algılayamadığı ya da yorumlayamadığı durumlarda yapay zeka ve görüntü işleme algoritmaları teknolojik cihazlar yardım ile sonuca kolayca ulaşabilmektedir.

Yapay zeka kavramına 1956 yılında Dartmouth konferansında J.McCarthy tarafından ortaya atıldı. Yapay zeka ses tanıma , görüntü işleme, doğal dil işleme, karar verme gibi alanlarda kullanılabilmektedir. İnsan hayatında yapay zeka, sesli asistanlar, dil çevirileri, e-Ticaret, siber güvenlik, otonom araçlar, oyunlar gibi farklı birçok alanda kullanılmaktadır. [1]

1.1 PROJENİN AMACI

Yapay zeka uygulamaları son zamanlarda artan bilimsel gelişmeler ile yaygın olarak kullanılmaya başlandı. Teknolojinin günlük hayata uygulanması ve başarılı performans sergilemesi ile yapay zeka insan hayatının birçok alanında yer edinmeye başladı. Bu tez kapsamında derin öğrenme mimarilerinden olan ESA(Evrişimli Sinir Ağları) ile hastalık sınıflandırma işlemi otomatik olarak yapılmış ve dosya olarak kullanıcıya sonuçlar verilmiştir. Deneysel çalışmalar için hazır olarak alınan elma, mısır, patates, çilek, domates, şeftali yaprakları ile oluşturulan veri seti kullanılmıştır. ESA tabanlı yaklaşım ile verilerden hastalık tespiti için kurulan model ile özellik çıkarma işlemi yapılmıştır. Girdi resimleri RGB kanallı 256x256 boyutundadır. Oluşturulan modelde evrişim katmanları, havuzlama katmanları son aşamada tam bağlı katmanlara girdi olarak verilmiştir. Test aşamasında ortaya çıkan sonuçlar -csv formatında kaydedilmiştir.

Ortaya çıkan sonuçlara göre üretici tespiti hızlı bir şekilde görebilecektir ve hastalığın tedavisine odaklanabilecektir. Veri setindeki bitkiler ayrı ayrı olarak da test edilebilir ya da test dosyasında birden fazla bitki de test edilebilir. İstenen sonuca uygun olarak model kurgulanabilir. Ayrıca veri setinde olmayan bitkiler de eklenip model tekrardan ayarlanabilir ve test edilebilir. Böylelikle model farklı durumlar için geliştirme imkânı sunmaktadır.

Yapay zeka ve makine öğrenmesi alanında gerçekleşen çalışmalar ile uygulama alanları giderek artmıştır. Bu gelişmeler ışığında insan hayatına faydalı olabilecek uygulamalar yapılmaya başlanmıştır. Bu çalışmada üreticiler için yardımcı olabilecek bir yapay sinir ağı uygulaması yapılmaya çalışıldı.

1.1 LİTERATÜR TARAMASI

Arivazhagan ve arkadaşları 2013 yılında, bitki yapraklarındaki hastalıkların tespit edilmesi ve sınıflandırılmasını amaçlamışlardır. Hastalıklı bitki yapraklarının bölgesini tespit edilmesi ve doku özellikleri ile hastalığın sınıflandırmışlardır. Önce görüntü işleme yöntemleri bitkinin yaprağını özellikleri çıkartılmıştır. Görüntü işlemeden çıkan giriş resimlerini minimum uzaklık kriteri uygulanır ve elde edilen sonuçlar destek vektör makinelerine aktarılıp sınıflandırılmıştır. [2]

Sladojevic tarafından 2016 yılında yapılan çalışmada evrişimli sinir ağları kullanılarak bitkilerin yapraklarındaki hastalıklar tespit edilmiştir. Geliştirilen model sağlıklı bitki yapraklarını diğer hastalıklı 13 bitki yaprağından ayırt edebilmektedir. Veri büyütme işlemleri uygulanarak veri seti geliştirilmiştir. Deney sonuçları sınıflar arasında %91- %98 arasında değişen başarı sağlamıştır [3].

Mohanty tarafından 2016 yılında yapılan çalışmada 54,306 resim kullanılarak 14 çeşit bitki türünün 26 farklı hastalığı derin öğrenme kullanılarak tespit edilmiştir. Derin öğrenme ile eğitilen model %99,3 başarı oranına ulaşmıştır [4].

Fuentes 2017 yılında domates bitkisinin yaprakları üzerindeki hastalıkların tespiti için çalışma yapmıştır. Çalışmada CNN (Convolutional Neural Network), R-CNN (Region Convolutional Neural Network) ve SDD (Single Multibox Detector) modelleri kullanılıp performansları karşılaştırmıştır. VGGNet ve ResNet gibi özellik çıkarma yapıları kullanılmış ve başarı ile tespit etmiştir [5].

Türkoğlu tarafından 2018' de yapılan çalışmada derin öğrenme algoritmalarının başarımları oranları karşılaştırılmıştır. Özellik çıkarma ve ön işlemler evrişimli sinir ağları ile yapılmıştır. Kullanılan derin öğrenme algoritmalarından AlexNet, VGG16, VGG19, ResNet50 ve GoogleNet gibi yapılar kullanılmıştır [6].

Walleign ve arkadaşları 2018 yılında soya bitkisinin hastalığının evrişimli sinir ağları kullanarak tespit etmeye çalışmışlardır. Kullanılan veri seti 12,673 resimden ve 4 hastalık çeşidinden oluşmaktadır. 3 katmanlı bir mimari oluşturmuştur Model evrişim katmanları ve havuzlama katmanlarından oluşmaktadır. Yüksek test başarısı elde etmişlerdir[7].

Saleem tarafından 2019 yılında yapılan çalışmada derin öğrenme yöntemleri ile çeşitli görüntü işleme algoritmaları kullanılarak bitkilerin hastalıklarının tespit edilmesi amaçlanmıştır. Çeşitli yapılar kurulup test edilmiştir [8].

Doğan tarafından 2019 yılında yapılan çalışma da domates ve elma bitkisinin hastalıkları derin öğrenme ile tespit etmiştir. Evrişimli sinir ağları ve Öğrenmeli vektör kuantalama kullanmıştır [9].

Hanbay 2019 yılında kayısı bitkisinin hastalıklarını tespiti ve sınıflandırılması için sinir ağı kurgulamıştır. Derin evrişimsel sinir ağı modeli geliştirilip dört tür hastalık tespit etmiştir. İHA ve fotoğraf makinesi kullanılarak kayısı ağaçları görüntülenmiştir. Model evrişim katmanı , havuzlama katmanı, normalizasyon katmanı , Relu (Rectified Linear Unit) ve tam bağlı katmanlardan oluşmaktadır. Kayısı hastalıkları %98,2 oranında başarı ile sınıflandırmıştır [10].

Aslan tarafından 2021 yılında derin öğrenme algoritmalarından AlexNet kullanılarak daha önceden eğitilmiş model transfer öğrenimi ile hastalık tespitinde kullanmıştır. Yapılan çalışmada %99,3 doğruluk oranına ulaşmıştır [11].

2. TEMEL BİLGİLER

2.1 ÖĞRENMENİN TANIMI

Öğrenme , canlıların tecrübelerinden ortaya çıkan devamlı ve kalıcı süreçtir. Olaylara karşı cevap vermek ve gerektiği durumlarda bu tepkileri değiştirip yeni beceriler kazanma durumu olarak açıklanabilir. Bir bilginin ya da yeteneğin , öğrenme olarak kabul edilmesi için bireye yeni anlam kazandırması gerekmektedir. Yani öğrenme anlık gelişen bir durum değil, adım adım ilerleyen bir süreçtir. Öğrenmenin doğrudan gözlemlenmesi mümkün değildir. Gözlemlenebilen şey canlının göstermiş olduğu davranışlardır.[12]

2.2 YAPAY ZEKA

En karmaşık ve gizemli organımız olan beyin; düşünme, kavrama, nesneleri algılama ve sonuç çıkarma gibi becerilerin gerçekleştiği yerdir. Bu yeteneklerin tümüne zeka denir. Bilim insanları zekanın dijitalleşmesi için birçok çalışma yapmıştır.

Yapay zekanın doğuşu Dartmouth College’ de düzenlenen konferans kabul edilir. Kurucuları ise yine aynı konferansa katılan M. Minsky, A. Newell ve H.Simon adlı bilim insanlarıdır. Yapay zeka terimi ise ilk defa John McCarthy tarafından kullanılmıştır. [13]

Yapay zeka bir bilgisayarın insan zekasına ait özellikleri taklit etmesi olarak tanımlanabilir. Yaptığı yapay zeka çalışmalarıyla bilinen Google Mühendislik Müdürü Ray Kurzweil yapay zekayı şu şekilde tanımlamaktadır; “Yapay Zeka, insanlar tarafından gerçekleştirildiğinde zeka gerektiren işlevleri yerine getiren makineler yaratma sanatıdır”. Bilginin ve teknolojinin gelişmesiyle genel olarak AI (Artificial Intelligence) adlandırılan yapay zekaya ilgi giderek artmaktadır [14].

2.3 MAKİNE ÖĞRENMESİ

Makine öğrenmesi yapay zekanın alt bilim alanıdır. Kökleri matematiğe ve istatistiğe dayanmaktadır. Yapısal işlev olarak öğrenebilen ve öğrendiği şeyler üzerinden tahmin yapabilen algoritmaların çalışma ve inşalarını araştıran bir sistemdir. Bilgisayar ortamında ya da elektronik ortamda sorunların çözümü için belirli işlem adımları vardır. Algoritma, bir işin yapılmasını sağlayan talimatların sırası olarak tanımlanabilir. Bir sorun için farklı algoritmalar oluşturulabilir. Farklı yöntemler makinenin daha da öğrenmesini ve gelişmesini sağlar.

Bazı sorunların çözümü için bir algoritma olmayabilir. Böyle durumlarda sorunun çözümü için elimizde bulunan verileri kullanılabilir. Bir işi insan eliyle tek tek ayırarak yapmak yerine bilgisayara öğrenilmesi istenen şeyleri öğretip otomatik sonuçlar alınması beklenebilir. Yani istenen iş adımlarının bilgisayar ile oluşturması istenmektedir. Makine öğrenmesi ile geçmişteki veriler kullanılarak yeni veri için ideal model oluşturulmaya çalışılır. Makine öğrenmesi çeşitli öğrenme metotlarına sahiptir.

Gözetimli öğrenme, giriş değerleri ile istenen çıkış değerini eşleştiren bir fonksiyon oluşturulmaktadır. Bu fonksiyon sınıflandırma ya da eğri uydurma algoritmaları ile belirlenebilir.

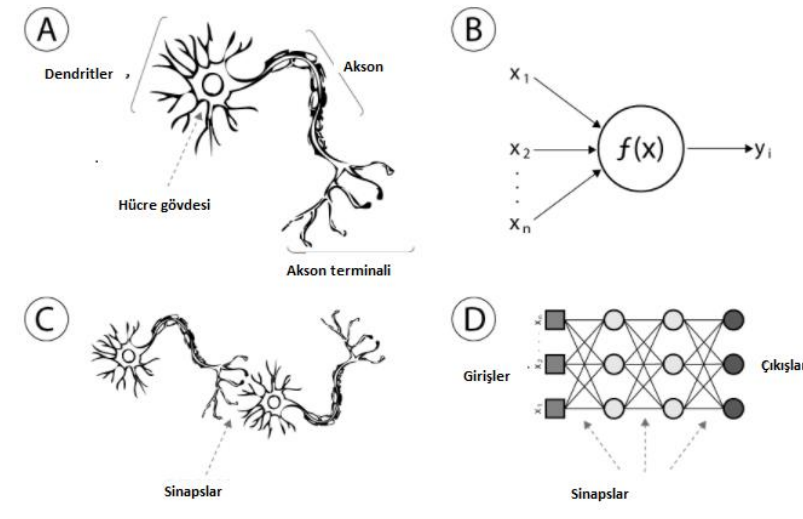
Gözetimsiz öğrenme, işaretlenmemiş veri üzerinden bilinmeyen bir tahmin yapıyı tahmin etmek için uygulanan makine öğrenmesi algoritmasıdır. Giriş verisinin hangi sınıfa ait olduğu belirsizdir [15].

Öğrenme sürecinde veriler algoritmaya iletilir. Ardından özellik çıkarımı, model oluşturma, değerlendirme ve tahmin gibi süreçler gerçekleşir. Değerlendirme aşamasında ise doğruluk, kesinlik, belirginlik gibi testler yapılmaktadır. Yapılan testler sonucunda model başarısız olduğu düşünülürse model düzenlenebilir ya da yeniden oluşturulabilir. Süreç ideal model oluşturulana kadar devam etmektedir.

2.4 YAPAY SİNİR AĞLARI

Yapay sinir ağları , insan beynin çalışma algoritmasını temel alarak beynin hatırlama, öğrenme, algılama, genelleme gibi ana işlevlerini herhangi bir yardım almadan gerçekleştirmek amacıyla geliştirilen bilgisayar sistemidir. YSA(Yapay Sinir Ağları), biyolojik sinir ağları taklit eden sentetik ağlardır [16].

Yapay sinir ağları ; insan beyni esas alınarak ,öğrenme sürecinin matematiksel olarak modellenmesi ile meydana gelmiştir. İlk yapay sinir ağı modeli sinir hekimi Warren McCulloch ve matematikçi Walter Pitts tarafından 1943 yılında “Sinir Aktivitesinde Düşüncelere Ait Bir Mantıksal Hesap (A Logical Calculus of Ideas Immanent in Nervous Activity)” başlıklı makale ile ortaya çıkarılmıştır [17].



Şekil 1. (A) İnsan nöronu, (B) Yapay nöron, (C) Biyolojik Sinaps ,(D) YSA Sinapsları [18]

Yapay sinir ağları bilgi işleme özelliği olan nöronlardan (sinir hücreleri) oluşmaktadır. Şekil 1’ de insandaki sinir hücresinin yapay sinir ağı benzetimi gösterilmiştir. Çizelge 1’de biyolojik sinir sisteminin kısımlarının yapay sinir sistemindeki karşılıkları gösterilmiştir.

Çizelge 1. Biyolojik Sinir Sistemin yapay sinir sistemindeki karşılıkları [19].

Biyolojik Sinir Sistemi	Yapay Sinir Sistemi
Nöron	İşlem Elemanı
Dentrit	Toplama Fonksiyonu
Hücre Gövdesi	Aktivasyon Fonksiyonu
Aksonlar	Yapay Nöron Çıkışı
Sinapslar	Ağırlıklar

Biyolojik sinir sisteminde, sensörler gibi çalışan dentritler kendilerine iletilen sinyalleri çekirdeğe iletmektedir. Çekirdek dentritten aldığı sinyaller değerlendirilmekte ve ağırlandırılarak aksonlara iletmektedir. İletilen sinyaller aksonlardan sinapslara iletilmektedir. Sinapslar gelen sinyalleri sinir hücrelerine iletmektedir. Bu şekilde sinir hücreleri arasında iletim sağlanmış olur.

Bir yapay sinir hücresi beş bölümden oluşmaktadır. Şekil 2’de yapay sinir hücresinin yapısı gösterilmiştir.

Girdiler: Yapay sinir ağlarına eğilmek için gelen verilerdir.

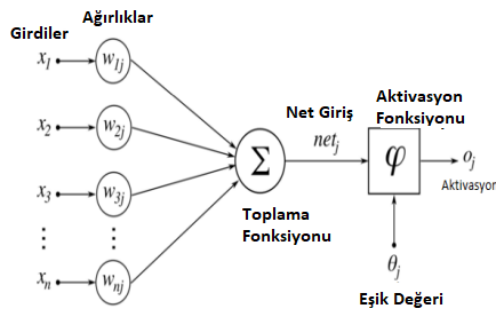
Ağırlıklar: Yapay sinir hücresine gelen veriler ağırlıklarla çarpılarak çekirdeğe iletilir.

Böylelikle giriş verilerinin çıkış verileri üzerindeki etkisi ayarlanır.

Toplama Fonksiyonu: Hücrenin net girdisinin hesaplandığı yerdir.

Aktivasyon Fonksiyonu: Katmanlar arası geçişi sağlayan fonksiyondur.

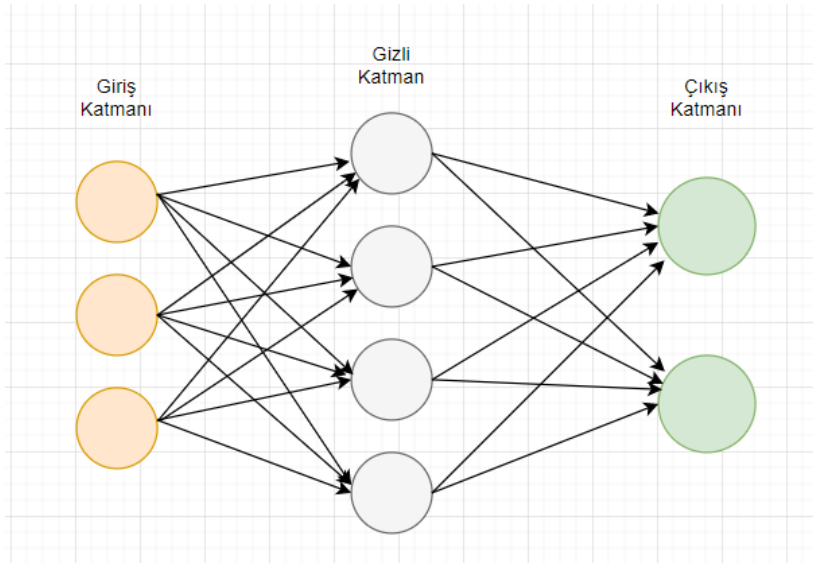
Çıktılar: Girdilere uygun bir çıktı üretilmektedir. Birden fazla girişin tek bir çıkışı vardır.



Şekil 2 Yapay Sinir Hücresinin gösterimi[20].

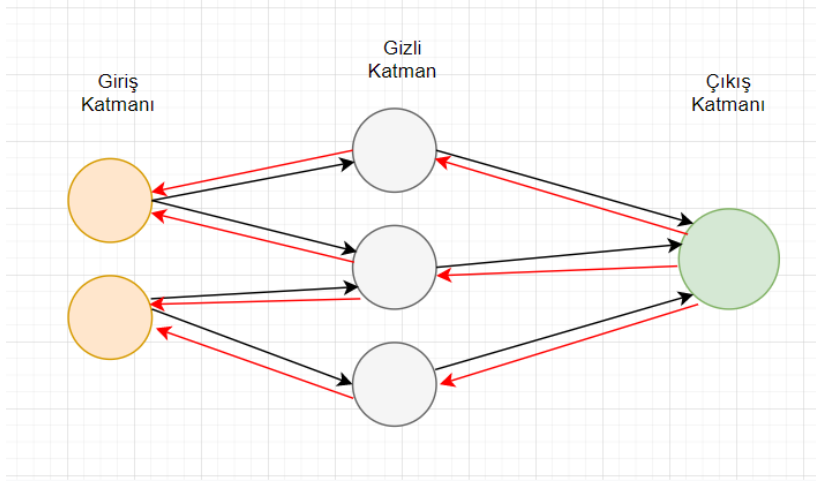
Yapay sinir ağılar ileri beslemeli ağı yapısı ve geri beslemeli ağı yapısı olarak iki yapıya ayrılmaktadır. Şekil 3’te ileri beslemeli yapay sinir ağı yapısı gösterilmiştir.

İleri beslemeli modelde sinir hücreleri giriş katmanından çıkış katmanına doğru tek yönlü ve düzenli bir şekilde akar. Modelde bir önceki sinir hücresine geri dönüş mümkün değildir. Katmanlarda bulunan sinir hücrelerini birbiri ile bağlantıları yoktur yani veri akışı birbirine bağlı katmanlar arasında olur. Giriş katmanı ve çıkış katmanı arasında bulunan gizli katman sayısı değişebilir. Katman sayısı değişeceği için nöron sayısı da değişir yani modelin göstereceği performansta değişiklikler olur.



Şekil 3. İleri Beslemeli Yapay Sinir Ağı.

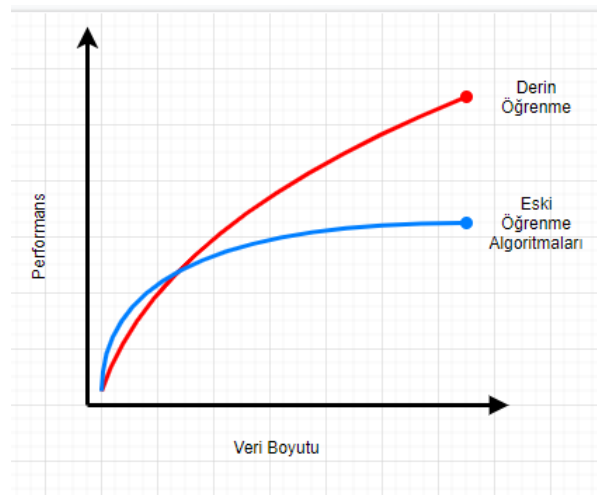
Şekil 4’te gösterilen geri beslemeli yapay sinir ağında ise girişler ileri ve geri yönde aktarılabilir. Çıkış verileri sadece kendinden sonra gelen katmana verilmez aynı zamanda kendinden önceki ya da kendi bulunduğu katmana girdi olarak verilebilir. Yapısı itibariyle doğrusal olmayan dinamik hafızaları vardır. Geri beslemeli sinir ağları tahmin uygulamaları için uygundur. Geri beslemeli sinir ağı yapısı, zincir kuralı kullanılarak, ağı üzerindeki tüm ağırlıklara hata fonksiyonunun ağırlıklarını yaymaktır [21].



Şekil 4. Geri Beslemeli Yapay Sinir Ağı.

2.5 DERİN ÖĞRENME

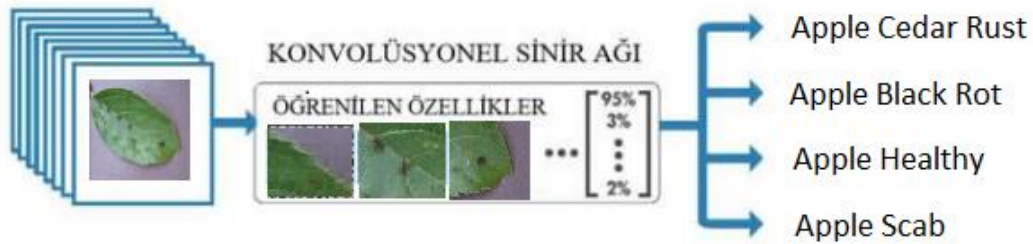
Derin öğrenme, özellik ve görevlerini veriden öğrenerek ilerleyen makine öğrenimi alanıdır. Yapısında üç veya daha fazla katmana sahip sinir ağı olan derin öğrenme ,yapay zeka ve makine öğreniminin alt alanıdır. Boyutları büyük olan verileri işleyerek daha küçük boyutlu veriler elde etmek amacıyla derin mimarileri kullanan algoritmalarından oluşmaktadır. Şekil 5'te derin öğrenme ile eski öğrenmenin performansları karşılaştırılmaktadır.



Şekil 5. Derin Öğrenme ile Eski Öğrenme Algoritmaları arasındaki ilişki.

Derin öğrenmenin temelinde genellikle Yapay sinir ağları vardır. YSA'nın gizli katmanlarla daha derinleştirilmiş halidir. Derin sinir ağları, her tahmini veya sınıflandırmayı iyileştirmek ve işlemek için birbirine bağlı birçok katmandan oluşur. Derin sinir ağlarının giriş ve çıkışına görünür katman denir. Giriş katmanı verileri aldığı yerdir ve çıkış katmanında ise tahmin ya da sınıflandırmanın yapıldığı kısımdır.

Derin öğrenme algoritmaları , öğrenme işlemi için giriş verisi olarak resim alındığında tüm pikselleri giriş verisi olarak işler. Giriş verileri , gri ya da renkli olabilir. Renkli bir giriş verisi üç farklı renk katmanından oluşacağı için mevcut pikselin üç katı kadar veri kullanılır. Gri tonlu bir resimde ise piksel sayısı kadar giriş verisi kullanılır. İlk olarak evrişim katmanları , havuzlama katmanları uygulanarak öznitelik haritası çıkartılır ve tam bağlı katmanlara aktarılır. Tam bağlı katmanlarda her pikselin her sınıf için olasılık değerleri ortaya çıkar. Bu değerler her pikselin hangi sınıfa ait olduğunu belirler. Şekil 6'da olasılık değerleri ortaya çıkan bir görsel yer almaktadır.



Şekil 6. Evrişimli Sinir Ağlarının algoritmasının çalışma prensibi [22].

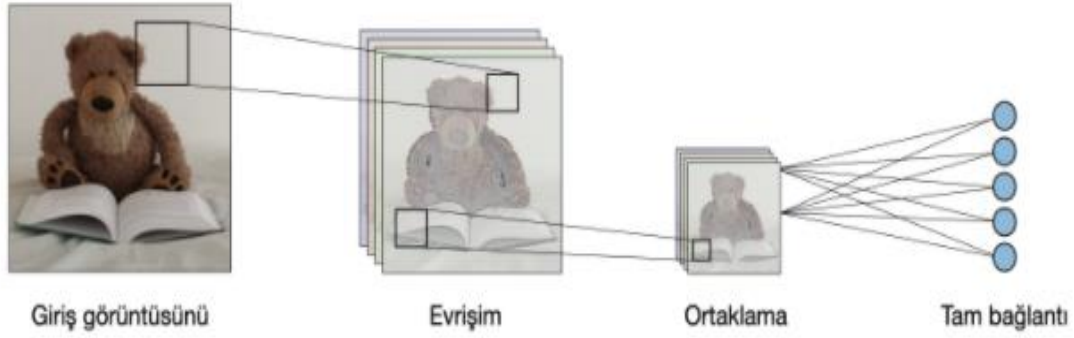
Derin öğrenme algoritmalarını, YSA dan ya da makine öğrenmesi algoritmalarından ayıran en önemli fark , kendi içerisinde yer alan katmanların öznitelik haritası çıkarabilmesidir. Bu yapılar sayesinde resimlere farklı filtreler uygulanarak öznitelik haritaları evrişimsel sinir ağları tarafından analiz edilir.

Derin öğrenmenin gelişmesinde en önemli etken teknolojik gelişmelerdir. Artan işlemci gücü ve yetenekleri ile eğitimde kullanılacak veri sayısı ve modellerde kurgulanan katman sayıları artmıştır. Daha büyük sinir ağları oluşturdukça ve daha fazla veriyi eğittikçe performansları da gelişmeye başlar. Böylelikle karmaşık problemler üzerine çözüm üretmek daha kolay olmuştur.

2.6 EVRİŞİMLİ SİNİR AĞLARI

ESA (Evrışimli Sinir Ağları), genellikle görüntü tanıma, nesne sınıflandırma gibi uygulamaların modellerinin oluşturulmasında kullanılan derin yapay sinir ağlardan oluşmaktadır. ESA ile insan ya da canlının yüzünü tanıma, canlı ya da cansız varlıları sınıflandırma, hastalıkları tespit etme, farklılıkları ayırt etme gibi çok çeşitli işlemler yapılabilmektedir. Evrışimli sinir ağlarına olan ilgi görüntü işleme ve ses işlemedeki başarısının yüksek olması nedeniyle giderek artmıştır.

ESA , Şekil 7 de gösterildiği üzere evrişim katmanı, havuzlama katmanı, aktivasyon katmanı ve tam bağlı katman olmak üzere dört ana katmandan meydana gelmektedir.



Şekil 7. Evrişimli Sinir Ağlarının temel katmanları [23].

Giriş olarak verilen görüntüler ilk olarak evrişim katmanında işlenmektedir. Evrişim katmanında çeşitli filtreler uygulanarak görüntünün özellik haritası çıkarılmaktadır. Filtreler belirli adım sayısında ilerleyerek giriş görüntüsünün üzerinde gezmektedir. Giriş resminin renk kanal sayısı ve formatına göre işlemin yoğunluğu değişmektedir. Evrişim katmanından çıkan veriler havuzlama katmanına geçmektedir.

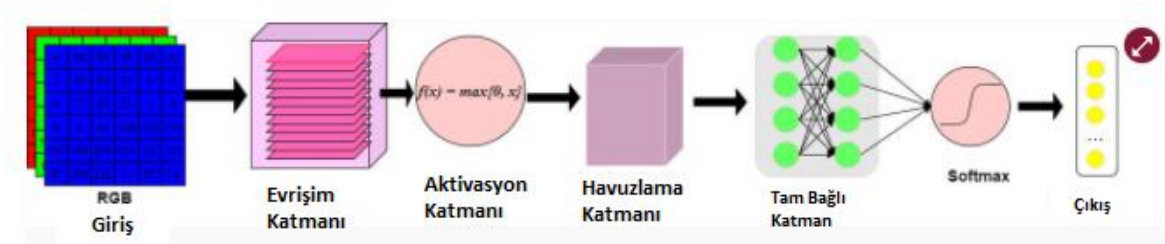
Havuzlama katmanında öğrenilen parametre yoktur. Giriş matrisinin kanal sayısı sabit kalacak şekilde resmin boyutları küçültülür. Hesaplamalardaki karmaşıklığı gidermek için kullanılan bir yöntemdir. Ancak bazı bilgiler kaybolabileceği için başarıdan ödün vermek gerekebilir. Veriler son olarak tam bağlı katmana iletilir ve eğitim işlemi gerçekleştirilir.

Tam bağı katman, her giriş verisinin tüm sinir hücrelerine bağı olduğu bir giriş üzerinde çalışmaktadır. Tam bağı katmanın amacı, sınıf olasılıklarını hesaplamaktır. Evrişim katmanında çıkan verilerin düzleştirilmiş vektörlerini girdi olarak alır.

Giriş verisi olarak kullanılacak resim veya video ham şekildedir. Yani verinin çözünürlüğü, boyutu, renk formatı modelin yapısını dolayısıyla başarısını etkilemektedir. Eğer görüntü yüksek boyut ve çözünürlükte seçilirse eğitim süresi ona bağı olarak artmaktadır ve buna bağı olarak bellek ihtiyacı ortaya çıkmaktadır. Giriş verisi küçük boyutlu seçilirse iş yükü azalır fakat buna bağı olarak performans düşmektedir.

Mimari olarak Şekil 8’deki gibidir.

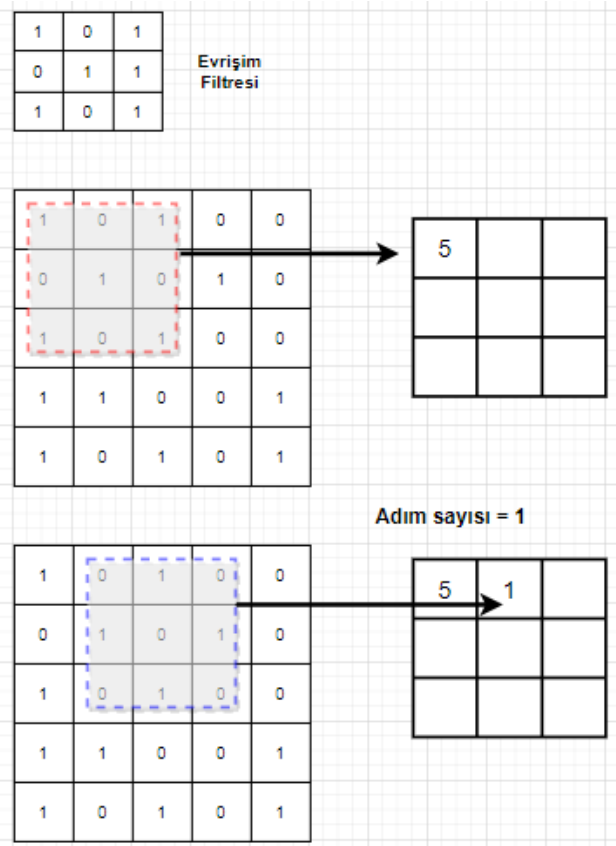
- Giriş Katmanı: Genişlik x Yükseklik x Derinlik, derinlik renk kanallarını temsil eder.
- Evrişim Katmanı : Filtrelerin uygulandığı, nitelik haritasının çıkarıldığı katmandır.
- Aktivasyon Katmanı : Normalizasyon işlemini gerçekleştirdiği katmandır.
- Havuzlama Katmanı : Ağ içindeki parametrelere ve hesaplama sayısının azaltılması için kullanılır.
- Tam Bağı Katman : Sınıf olasılıklarının hesaplandığı katmandır.



Şekil 8. Evrişimsel Sinir Ağlarının genel mimarisi [24].

2.6.1 Evrişim Katmanı

Evrişim katmanları, ESA'ların yapı taşlarını oluşturur. Bu katmanda, giriş verisinin konvolüsyon uygulanarak nitelik haritası çıkarılmaktadır. Evrişimli katmanlar filtre adı verilen ağırlık matrislerinden oluşur. Belirli bir filtre tüm resim üzerinde gezdirilerek ortaya yeni bir matris çıkartılır. Kullanılan filtrelerin katsayıları her adımda yinelenir. Böylelikle resim üzerindeki önemli yerler tespit edilmiş olur. Bir filtrenin derinliği, evrişim katmanındaki filtre sayısına eşittir ve her bir girdide farklı özellik aramaktadır.

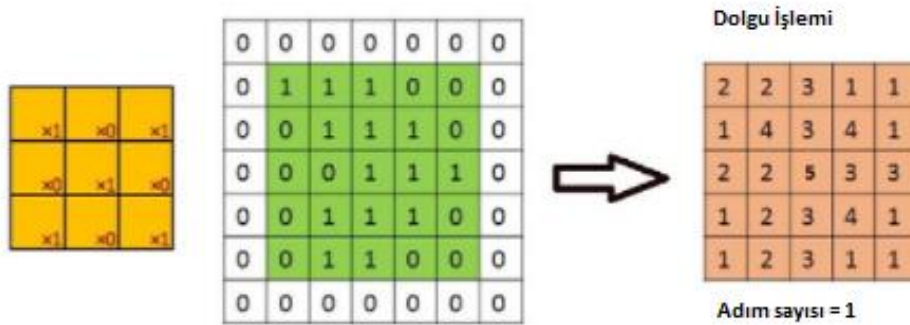


Şekil 9. Evrişim işleminin gösterimi

Evrişim katmanında, filtre verilen giriş matrisinin üzerinde sol üstten başlayarak verilen adım sayısında kayarak sağ alt noktaya ulaşmaktadır. Her adımda filtredeki pikseller görüntüde karşılık gelen piksel değeri ile çarpılır ve sonuçlar toplanır. Şekil 9'da yapılan işlem gösterilmektedir.

Filtrenin kaç piksel kaydırılacağı adım sayısı ile sağlanır. Adım sayısının iki olması durumunda, matris her seferinde iki piksel kayarak işlem yapacaktır. Adım sayısı evrişim işlemi sonunda ortaya çıkan matrisin boyutunu etkilemektedir. Eğer adım sayısı büyük olursa veri sayı az olacaktır ancak verilerden elde edilen kayıplar fazla olacaktır.

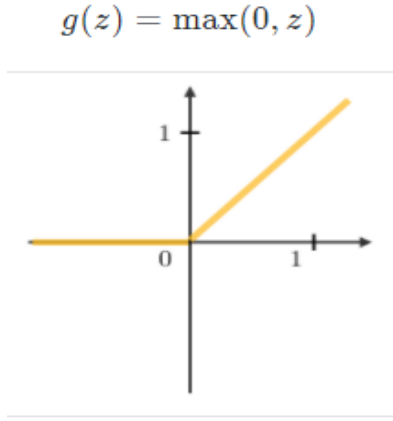
İlk filtre uygulanınca bir özellik haritası oluşmaktadır ve bir özellik tespit edilmektedir. Ardından ikinci bir filtre uygulanarak başka bir özellik tespit edilir. Ortaya çıkan özellik haritası orijinal resimden küçük olduğu için boyutunu ayarlamak için dolgu işlemi uygulanır. Şekil 10'da zero padding işleminin nasıl yapıldığı ve filtrenin matris üzerinde nasıl gezindiği gösterilmektedir.



Şekil 10. Evrişim katmanında dolgu işleminin gösterimi [25].

2.6.2 Aktivasyon Katmanı

Aktivasyon katmanında, evrişim katmanında oluşan matrisler eğitime uygun hale getirilmektedir. Yapay sinir ağlarında doğrusal olmayan vektörleri kullanmak için aktivasyon fonksiyonlarına ihtiyaç duyulur. Aktivasyon kullanılmayan bir sinir ağı sınırlı öğrenme gücüne sahip bir doğrusal bağlanım gibi davranacaktır. Kurgulanan sinir ağının doğrusal olamayan durumları da öğrenmesi gerekir. İşlenen görüntülerden anlamlı özellikler çıkarmak için doğrusal olmayan durumlara ihtiyaç duyarız. Doğrusal olamayan aktivasyon fonksiyonları ile ağın öğrenimi hızlanmış olur. ReLU aktivasyon fonksiyonu doğrusal olan yapıyı doğrusal olmayan yapıya dönüştürmek için kullanılan en sık fonksiyonlardan biridir. Şekil 11' de ReLU fonksiyonun grafiği ve parçalı fonksiyon olarak ifadesi gösterilmektedir.



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

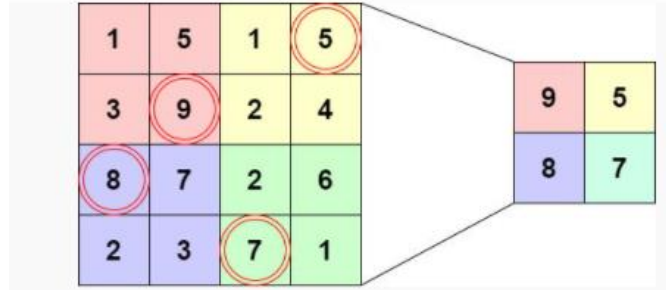
Şekil 11. ReLU aktivasyon fonksiyonu [26].

Aktivasyon işlemlerinin eğitim sürecinde tek seferde binlerce düğüme uygulanması gerekir. Sigmoid ya da hiperbolik tanjant gibi fonksiyonlar da kullanılabilmektedir fakat ReLU az hata oranı ve hızı sebebiyle tercih edilmektedir. ReLU hesaplama açısından verimli ve basittir. [27]

2.6.3 Havuzlama Katmanı

Evrişim katmanının ardından genel olarak havuzlama katmanı eklenmektedir. Havuzlama katmanı ile giriş boyutu azaltılır böylelikle parametre ve hesaplama miktarı azalmaktadır. Ağda oluşabilecek uyumsuzluklar kontrol edilmiş olur. Aşırı öğrenme durumunun önüne geçmek için kullanılan yöntemlerden biridir.

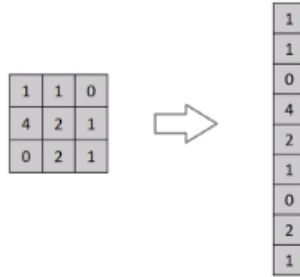
Evrişim katmanında olduğu gibi belirli bir filtre ile görüntü üzerinde kaydırılarak yapılır. Sol üstten başlayarak adım adım sağ alta doğru ilerler. Genellikle filtre boyutu olarak 2 x 2 boyutunda matris kullanılır. Filtreler genellikle maksimum değeri ya da değerlerin ortalamasını alarak bu işlemi yapar. Maksimum havuzlama iyi sonuçlar verdiği için daha çok tercih edilir. Maksimum havuzlama işleminde filtre resim üzerinde kayar ve bulunduğu piksellerdeki en büyük değeri alır. Filtre kaydıkça maksimum değerlerden yeni bir matris ortaya çıkmaktadır. Havuzlama katmanında yükseklik ve genişlik azalmaktadır ancak derinlik korunmaktadır. Şekil 12’de maksimum havuzlama işlemi sonucunda ortaya çıkan yeni matris ifade edilmektedir.



Şekil 12. Maksimum havuzlama işleminin uygulanması [28].

2.6.4 Düzleştirme İşlemi

Düzleştirme işleminin amacı , son ve en önemli katman olan tam bağlı katmana giriş verilerini hazırlamaktır. Evrişim ve havuzlama katmanından çıkan verilerin tek boyutlu diziye çevirir ve tam bağlı katmana aktarır. Şekil 13'te 3x3'lük bir matrisin düzleştirme işlemi sonucunda ortaya çıkan yeni matrisi gösterilmiştir.



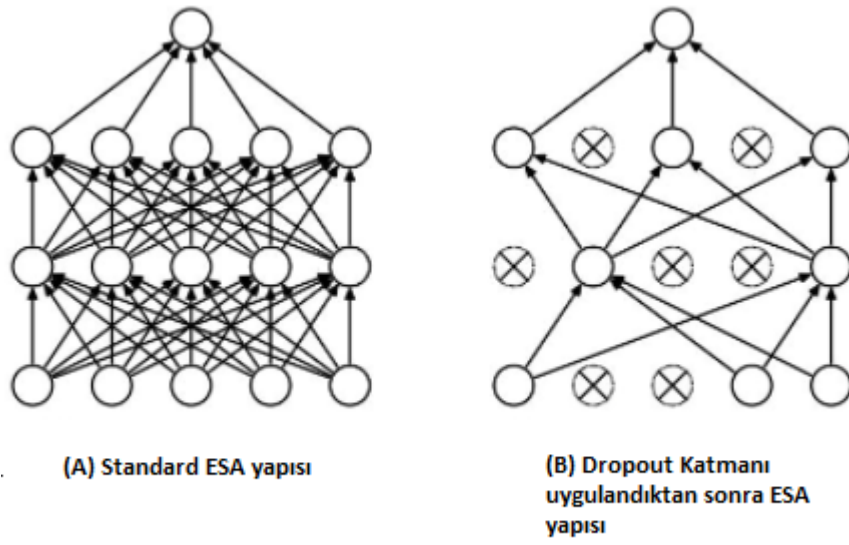
Şekil 13. Düzleştirme işleminin gösterimi [29].

2.6.5 Tam Bağlı Katman

Evrişimli sinir ağlarında, evrişim katmanı, aktivasyon katmanı, havuzlama katmanından sonra tam bağlı katman gelmektedir. Tam bağlı katman bir önceki katmanlarda etkinleştirilen düzleştirilmiş vektörleri girdi olarak alır. Bu katmanda veriler YSA ile eğitilmektedir. Tam bağlı katmanın sonunda istenilen sınıf sayısı kadar sınıf oluşturulmaktadır. Sınıflandırma işlemleri bu katmanın sonunda gerçekleşir. Yapay sinir ağı modeli olarak genellikle geri yayımlı algoritma kullanılmaktadır. Aktivasyon fonksiyonu olarak genellikle softmax tercih edilir. Softmax, çıktı katmanındaki değerlerin toplamının 1 olmasını sağlar ve sınıflandırma uygulamaları için kullanılır.

2.6.6 Dropout Katmanı

Evrişimli sinir ağlarında eğitim aşamasında bazı durumlarda ezberleme olayı meydana gelmektedir. Dropout katmanı ile ezberlemenin önüne geçilmeye çalışılır. Bazı düğümler kaldırılarak eğitim performansı artırılır. Eğitim esnasında belirli miktardaki bağlantı rastgele kapatılarak yapılır. Şekil 14'te dropout katmanı uygulandıktan sonra bağlantı koparılan ESA yapısı gösterilmektedir.



Şekil 14. Evrişimli Sinir Ağlarında Dropout Yapısı [30]

3. MATERYAL VE YÖNTEM

3.1 KULLANILAN YAZILIMLAR

Evrişimli sinir ağlarını kodlamak ve sonuçları değerlendirmek için kullanılan yazılımlar ve yazılım dilleri tanıtılmıştır.

3.1.1 Python Programlama Dili

Python 90'lı yıllarda geliştirilmeye başlanan programlama dilidir. Python nesne yönelimli, yorumsal, modüler ve etkileşimli, yüksek seviyeli bir dildir. Makine mantığına yatkın olması sebebiyle hızlı çalışmaktadır. Python basit bir söz dizimine sahiptir. Böylelikle hem program yazmak daha basit ve rahat hale gelir, hem de başkalarının yazdığı kodlar daha rahat anlaşılabilir. Python az kod ile çok şey yapmayı sağlar.

3.1.2 Google Colaboratory

Python programlama dilindeki kodları yazmayı ve çalıştırmayı sağlar. GPU ve TPU desteğini ücretsiz verir. Bu sayede yapay zeka uygulamaları için zaman ve performans kazanılmış olur.

3.1.3 Anaconda

Anaconda, veri bilimi ve yapay zeka uygulamaları için python kullanmak için hazırlanmış uygulamadır. İçinde Jupiter Notebook, Spyder, Orange ve PyCharm gibi geliştirme ortamlarını barındırmaktadır. Kullanılan kütüphaneler kolayca kurulup kullanılabilir.

3.2 KULLANILAN KÜTÜPHANELER

- **Tensorflow**

Bir dizi görev arasında veri akışı ve türetilenebilir programlama için kullanılan ücretsiz ve açık kaynaklı bir kütüphanedir. Yapay zeka ve makine öğrenme uygulamalarında çokça tercih edilmektedir.

- **Keras**

Derin öğrenme modellerini tanımlamak ve anlamlı hale getirip, eğitmek için kullanılan kütüphanedir. Yapay sinir ağları ile hızlı çalışan, modüler ve ücretsiz bir sinir ağı kütüphanesidir. Kolay ve hızlı bir şekilde model oluşturulabilir.

- **Matplotlib**

Veri bilimi çalışmalarında matematiksel hesaplamaların 2 veya 3 boyutlu görsel çıktıların alınmasını sağlayan veri görselleştirme kütüphanesidir. Kullanılması ve uygulaması basit ve hızlıdır.

- **Pandas**

Veri işleme ve analizi için kullanılmaktadır. Verileri işleyerek analizlerin daha hızlı ve kolay yapılmasını sağlar.

- **NumPy**

Veri bilimi çalışmalarında büyük boyutlu matematiksel hesaplamaların kolay ve hızlı şekilde gerçekleştirilmesini sağlayan kütüphanedir. Çok boyutlu matrisler ve dizilerde çalışmaya olanak sağlamaktadır. [31]

3.3 KULLANILAN VERİ SETİ

Veri seti oluşturulurken birçok veri seti incelenmiştir. Bitkiler ve hastalıklar göz önüne alınarak oluşturulmuştur. Çizelge 2’de veri setinin detayları verilmiştir.

Çizelge 2. Kullanılan veri setinin bilgileri

Hastalıklar	Train	Validation	Test
Apple_Black_Rot	350	125	100
Apple_Cedar_Rust	350	125	100
Apple_Healthy	175	75	100
Apple_Scab	350	125	100
Corn_Common_Rust	350	125	100
Corn_Gray_Leaf_Spot	350	125	100
Corn_Healthy	350	100	100
Corn_Northern_Leaf_Blight	350	100	100
Peach_Bacterial_Spot	350	125	100
Peach_Healthy	175	100	100
Potato_Early_Blight	350	120	100
Potato_Healthy	175	100	100
Potato_Late_Blight	350	125	100
Strawberry_Healthy	175	100	100
Strawberry_Leaf_Scorch	350	125	100
Tomato_Bacterial_Spot	350	125	100
Tomato_Early_Blight	350	125	100
Tomato_Healthy	350	125	100
Tomato_Late_Blight	350	125	100
Toplam	5950	2220	1900

Çizelge 2 de gösterildiği gibi veri seti 19 sınıftan oluşmaktadır. Veri setinde 13 hastalıklı ve 6 sağlıklı bitki yaprağı bulunmaktadır. Veri seti eğitim, doğrulama, test olmak üzere 3’e ayrılmıştır. Veri setindeki resimler RGB formatındadır. Şekil 15’te bazı yapraklar gösterilmiştir.



Şekil 15. Veri setinde bulunan örnek bitki yaprakları

3.4 KULLANILAN YÖNTEM

Verileri işlemek ve sınıflandırmak için evrişimli sinir ağları kullanılmıştır. Şekilde 16'da kodlamaya başlamadan önce kullanılan kütüphaneler verilmiştir.

```
import tensorflow as tf

from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D
from tensorflow.keras.layers import Dense
from tensorflow.keras.layers import MaxPool2D
from tensorflow.keras.layers import Dropout
from tensorflow.keras.layers import Activation
from tensorflow.keras.layers import Flatten

from tensorflow.keras.optimizers import Adam
from tensorflow.keras.preprocessing import image

import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import pandas as pd

from tensorflow.keras.callbacks import ModelCheckpoint
```

Şekil 16. Veri setini işlemek için kullanılan kütüphaneler

3.4.1 Verilerin İçerik Aktarılması ve görüntülerin işlenmesi

Modelin işlenmeden önce giriş resimlerinin normalleştirilmesi gerekmektedir. Ön işlem bölümünde `-rescale` ile tüm sayısal değerler 0 ile 1 aralığında bir değer ile ölçeklendirilmiştir. Resimlerde oluşabilecek piksel farklılıklarının önüne geçmek için yapılmaktadır. Böylelikle model eğilirken ve ağırlıklar güncellenirken tüm görüntülerin eşit olarak değerlendirilmesi sağlanır. Bazı veri setlerinde görüntülerin genelleştirilmeleri gerekebilir. Çeşitli metotlar ile örneğin rastgele döndürme, parlaklık artırma ya da azaltma, konumlarını yatay ya da dikey olarak kaydırma gibi çeşitli yöntemler kullanılabilir. Fakat kullanılan veri setinde bu işlemlere gerek görülmemiştir. Keras'ın 'ImageDataGenerator' sınıfı kullanılarak normalleştirme yapılmıştır. Yapılan işlem Şekil 17'de gösterilmiştir.

```
train_datagen = ImageDataGenerator(rescale=1./255,
width_shift_range=0.0,
height_shift_range=0.0,
fill_mode='nearest')
```

Şekil 17. Keras ile görüntülerin işlenmesi

Yine Keras'ın sağladığı yöntemlerden biri olan `flow_from_directory` ile girdi görüntülerinin yeniden boyutlandırılması sağlanır. Yapının güçlü oluşması için girdi boyutları 224x224 boyutlarında seçilmiştir. Ayrıca kullanılan görüntüler RGB formatındadır. Şekil 18'de kullanılan fonksiyonun kod yapısı verilmiştir.

```
training_set=train_datagen.flow_from_directory(file_path+'/train',
target_size=(224,224),
batch_size=batch_size,
class_mode='categorical')
```

Şekil 18. Keras ile görüntülerin boyutlandırılması.

3.4.2 Modelin Oluşturulması

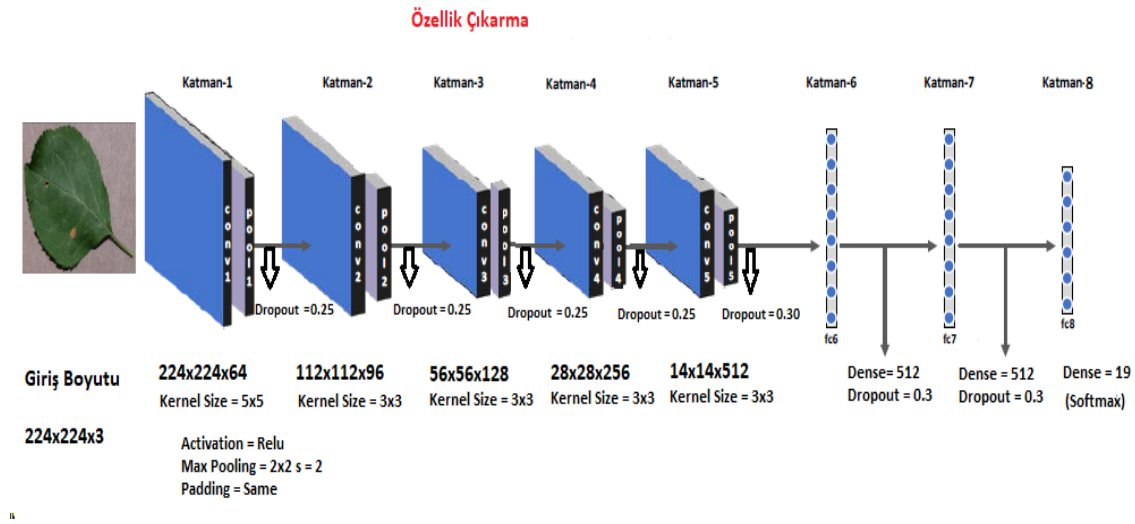
Model sorunu çözmeye yönelik şekilde baştan oluşturulmuştur. `-Sequential` ile bir Keras modeli oluşturulmaya başlanmıştır. `-Sequential` metodu ile oluşturulan katmanların ardışık olacağı anlaşılmaktadır.

```
model = Sequential() # modelim katmanlar şeklinde olmasını sağlar.
#-----1-----
model.add(Conv2D(filters = 64, # 5x5 boyutunda 64 adet Filtre kullanılır.
kernel_size = (5,5),
padding = 'Same', #bilgi kaybetmemek için matris çevresine 0' lar ile çerçeve yapılır.
activation = 'relu', #[0, + ∞ )
input_shape = (224,224,3) # giriş resimlerinin boyutları
))

model.add(MaxPool2D(pool_size=(2,2), strides=(2,2))) # 2x2 boyutlu havuz katmanı adım sayısı = 2
model.add(Dropout(0.25)) #Her seferinde nöronların %25'i atılır.
```

Şekil 19. Modelin birinci katmanının oluşturulması

Şekil 19 da modelin ardışık metot ile başladıktan sonra oluşturulan birinci katmanı yer almaktadır. Modelin birinci katmanında filtre sayısı ilk olarak 64 olarak belirlenmiştir ve ilerleyen katmanlarda arttırılmıştır. Kullanılan filtrenin matris boyutu 5x5' tir yani görüntü üzerinde 5x5 boyutunda matris dolandırılarak özellik haritası çıkarılmıştır. Verilen matris boyutu ilerleyen katmanlar da 3x3'e düşürülmüştür. Bilgi kayıplarını önlemek için 'padding' metodunu 'same' seçilmiştir bu sayede veri kaybı en aza düşürülmeye çalışılmıştır. Oluşturulan katmanın aktivasyon fonksiyonu ReLU olarak seçilmiştir. Evrişim katmanının son olarak boyutları Keras dili kullanıldığı için 224x224x3 seçilmiştir. Verilen boyutlar genişlik, yükseklik, renk formatını belirtmektedir. Keras kütüphanesi kullanıldığı için giriş boyutu verilirken renk formatı verilmesi gerekmektedir. Evrişim katmanının ardından havuzlama katmanı gelmektedir. Havuzlama katmanı olarak maksimum havuzlama uygulaması tercih edilmiştir. 2x2 boyutunda matris görüntü üzerinde her adımda 2 piksel kayarak ilerlemektedir.



Şekil 20. Modelin genel gösterimi.

Model, Şekil 20'de ifade edildiği gibi 5 evrişim katmanı, 5 havuzlama katmanı ve 3 tam bağlı katmandan oluşmaktadır. Evrişim katmanının tamamında, tam bağlı katmanın ise ilk 2 katmanın ReLU aktivasyon fonksiyonu tercih edilmiştir. Tam bağlı katmanın son katmanında Softmax aktivasyon fonksiyonu ile sınıflandırılma yapılmıştır.

3.4.2.1 Önemli parametrelerin ayarlanması

Optimizasyon metodu olarak Adam seçilmiştir. Sınıflandırma uygulamalarında sıkça tercih edilmektedir. Öğrenme katsayısı oluşturulan derin öğrenme modellerini etkilemektedir. Katsayı olarak 0,0001 gibi küçük bir değer kullanılmıştır.

Hata hesaplamaları modelin başarısını ölçen fonksiyonlardan biridir. Kayıp fonksiyonu modelin yaptığı tahminin, gerçek değer değerden ne kadar saptığını göstermektedir. Kayıp fonksiyonu olarak -categorical_crossentropy tercih edilmiştir.

Mini-Batch sayısı modelin eğitim aşamasında hızını, adım sayısını, hata hesaplarını etkileyen önemli paramterelerden biridir. Tüm verilerin eğitilmesi yerine verileri parçalara bölerek, parçalar üzerinden eğitim yapılmasını sağlamaktadır. Modelin eğitimi esnasında bu değer 32 olarak seçilmiştir. Genel olarak 2'nin katları kullanılmaktadır.

Eğitimde tur sayısı modelin başarısını ciddi şekilde etkilemektedir. İlk veri demeti eğitildikten sonra modelin başarı oranı ortaya çıkmaktadır oluşan başarıma göre geri yayılım ağırlıkları güncellenmektedir. Her adımdan sonra veri seti eğitilmekte ve ağırlıklar güncellenmektedir. Şekil 21'de modelin derlenmesi ve eğitimi sırasında kullanılan kod yapısı verilmiştir.

```
optimizer = Adam(lr=0.0001, beta_1=0.9, beta_2=0.999)
model.compile(optimizer = optimizer ,
              loss = "categorical_crossentropy",
              metrics=["accuracy"])

model_fit=model.fit(
    training_set,
    steps_per_epoch=train_num//batch_size,
    validation_data=valid_set,
    epochs=20,
    validation_steps=valid_num//batch_size,
    callbacks=[weights])
```

Şekil 21. Modelin derlenmesi ve eğitilmesi

Model: "sequential_4"

Layer (type)	Output Shape	Param #
conv2d_20 (Conv2D)	(None, 224, 224, 64)	4864
max_pooling2d_20 (MaxPooling)	(None, 112, 112, 64)	0
dropout_28 (Dropout)	(None, 112, 112, 64)	0
conv2d_21 (Conv2D)	(None, 112, 112, 96)	55392
max_pooling2d_21 (MaxPooling)	(None, 56, 56, 96)	0
dropout_29 (Dropout)	(None, 56, 56, 96)	0
conv2d_22 (Conv2D)	(None, 56, 56, 128)	110720
max_pooling2d_22 (MaxPooling)	(None, 28, 28, 128)	0
dropout_30 (Dropout)	(None, 28, 28, 128)	0
conv2d_23 (Conv2D)	(None, 28, 28, 256)	295168
max_pooling2d_23 (MaxPooling)	(None, 14, 14, 256)	0
dropout_31 (Dropout)	(None, 14, 14, 256)	0
conv2d_24 (Conv2D)	(None, 14, 14, 512)	1180160
max_pooling2d_24 (MaxPooling)	(None, 7, 7, 512)	0
dropout_32 (Dropout)	(None, 7, 7, 512)	0
flatten_4 (Flatten)	(None, 25088)	0
dense_12 (Dense)	(None, 512)	12845568
dropout_33 (Dropout)	(None, 512)	0
dense_13 (Dense)	(None, 512)	262656
dropout_34 (Dropout)	(None, 512)	0
dense_14 (Dense)	(None, 19)	9747
Total params: 14,764,275		
Trainable params: 14,764,275		
Non-trainable params: 0		

Şekil 22. Oluşturulan modelin her katmandaki parametre sayıları

Şekil 22’de model oluştuktan sonraki parametreler gösterilmektedir. Havuzlama ve dropout katmanının parametrelere doğrudan bir etkisi yoktur. Havuzlama katmanı ile resim boyutları küçülmekte, dropout ile nöronlar rastgele atılmaktadır. Böylelikle aşırı öğrenme durumunun önüne geçilmesi amaçlanmaktadır.

3.4.3 Modelin Görselleştirilmesi

Derin öğrenme uygulamalarında tahmin kısmından önce genellikle eğitilen veri görselleştirilmektedir. Ortaya çıkan grafikleri üzerinden doğruluk ve kayıplar görülebilmektedir. Veri görselleştirme kütüphanesi olarak Matplotlib kullanılmıştır. Eğitim sonucunda ortaya çıkan accuracy ve loss değerleri ilgili fonksiyonların yardımıyla grafiğe aktarılmıştır. Şekil 23’de matplotlib kullanılarak hazırlanan kod yapısı verilmiştir.

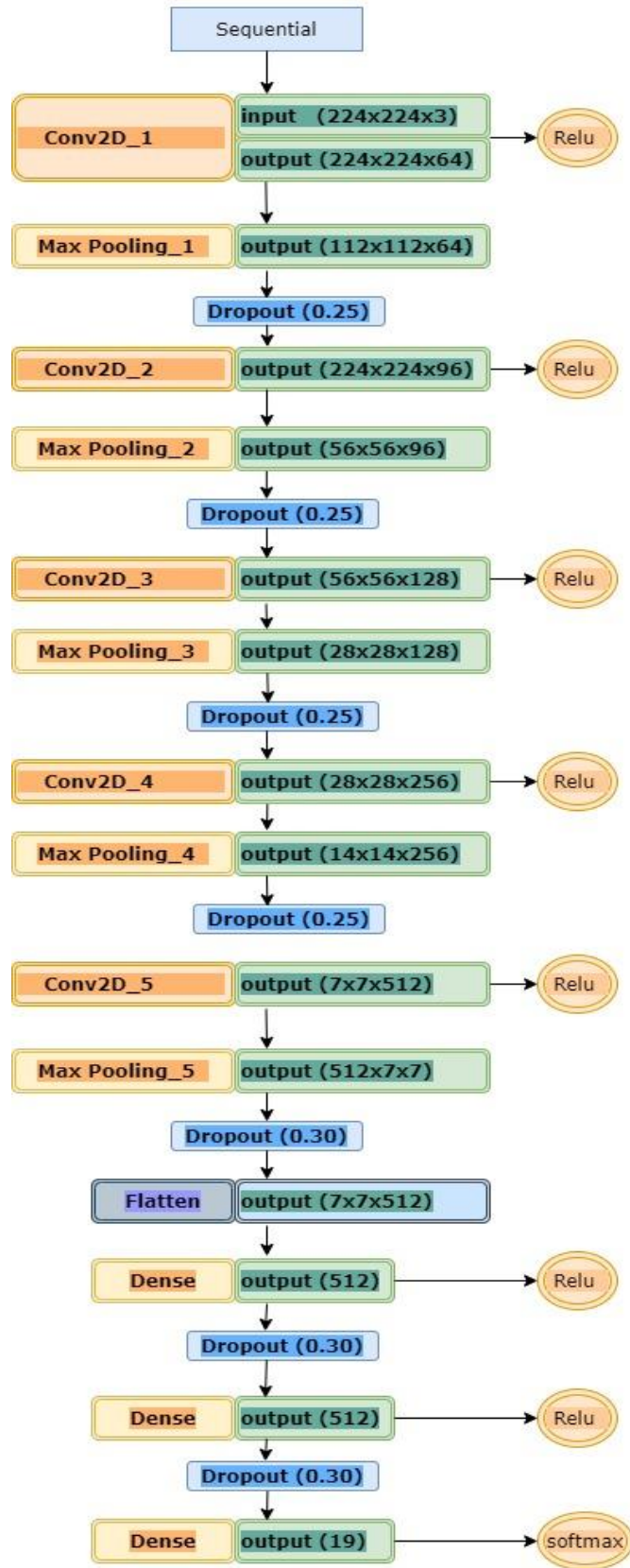
```
accuracy = model_fit.history['accuracy']
val_accuracy = model_fit.history['val_accuracy']
loss = model_fit.history['loss']
val_loss = model_fit.history['val_loss']
epochs = range(1, len(loss) + 1)

#accuracy plot
plt.plot(epochs, accuracy, color='green', label='Training Accuracy')
plt.plot(epochs, val_accuracy, color='blue', label='Validation Accuracy')
plt.title('Training and Validation Accuracy')
plt.ylabel('Accuracy')
plt.xlabel('Epoch')
plt.legend()

plt.figure()
#loss plot
plt.plot(epochs, loss, color='green', label='Training Loss')
plt.plot(epochs, val_loss, color='red', label='Validation Loss')
plt.title('Training and Validation Loss')
plt.xlabel('Epoch')
plt.ylabel('Loss')
plt.legend()

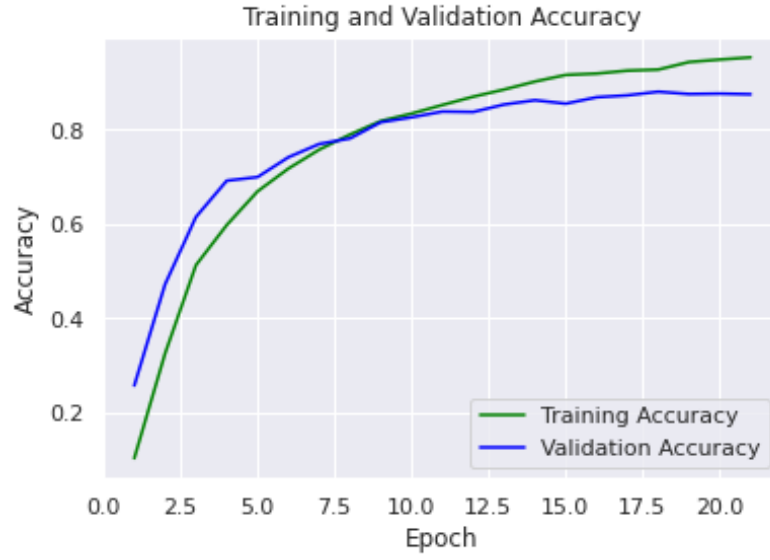
plt.show()
```

Şekil 23. Eğitim sonucunda modelin görselleştirilmesi.



Şekil 24. Modelin genel kurgusu

Şekil 24'te modelin kurgusunun daha rahat anlaşılabilmesi için hazırlanan görsel yer almaktadır. Eğitim sonrasında model %93,95 başarı göstermiştir. Şekil 25 ve Şekil 26 da modelin doğruluk ve kayıp grafiği verilmiştir. . Bu aşamadan sonra bir sonraki başlıkta oluşturulan modelin uygulanması ve test aşmasına geçilecektir.



Şekil 25. Eğitim sonrasında oluşan doğruluk grafiği.



Şekil 26. Eğitim sonucunda oluşan kayıpların grafiği.

4. TEST VE DEĞERLENDİRME

4.1 TEST EDİLEN RESİMLERİN DEĞERLENDİRİLMESİ

Test aşaması için ayrı bir yapı kurgulanmıştır. Test resimleri her sınıf için 100 adet, toplamda 1900 adettir. Resimlerin boyutları 224x224 pikseldir ve renkli formattadır. Yapılan test sonucunda test başarımları oranı %89,89 olarak tespit edilmiştir. Her sınıfın yüzdesel olarak doğruluk, kesinlik, duyarlılık oranları Çizelge 3’de gösterilmiştir.

Çizelge 3. Test edilen sınıfların ayrıntılı olarak gösterimi

	Kesinlik	Geri Çağırma	F-Skoru
Apple_Black_Rot	0.862385	0.94	0.899522
Apple_Cedar_Rust	0.942308	0.98	0.960784
Apple_Healthy	0.869565	0.80	0.833333
Apple_Scab	0.920000	0.92	0.920000
Corn_Common_Rust	0.961165	0.99	0.975369
Corn_Gray_Leaf_Spot	0.987654	0.80	0.883978
Corn_Healthy	1.000000	1.00	1.000000
Corn_Northern_Leaf_Blight	0.857143	0.96	0.905660
Peach_Bacterial_Spot	0.946809	0.89	0.917526
Peach_Healthy	0.938144	0.91	0.923858
Potato_Early_Blight	0.968421	0.92	0.943590
Potato_Healthy	0.964706	0.82	0.886486
Potato_Late_Blight	0.714286	0.80	0.754717
Strawberry_Healthy	0.969072	0.94	0.954315
Strawberry_Leaf_Scorch	0.980198	0.99	0.985075
Tomato_Bacterial_Spot	0.968085	0.91	0.938144
Tomato_Early_Blight	0.637681	0.88	0.734965
Tomato_Healthy	0.938776	0.92	0.929293
Tomato_Late_Blight	0.806818	0.71	0.755319

Oluşturulan modellerde sadece doğruluk oranı ile değerlendirilemez. Doğruluk %99 olduğu durumlarda dahi modelin başarısından emin olunmayan durumlar ortaya çıkmaktadır.

Çizelge 4. Sınıflandırmanın tahmin edilen ve gerçekleşen değer matrisi

	Tahmin Edilen	
	Gerçekleşen	
Gerçekleşen	True_Positives	False_Negatives
	False_Positives	True_Negatives

Çizelge 4’de True_Positive ve True_Negative olan indisler modelin doğru olarak tahmin ettiği, False_Positive ve False_Negative ise modelin yanlış olarak tahmin ettiği alanlardır. Örnek olarak Apple_Black_Rot’ u ele alırsak ;

- True_Positive: Apple_Black_Rot olarak tahmin edildi ve bu doğrudur.
- True_Negative: Apple_Healthy olarak tahmin edildi fakat bu doğru ise ,
- False_Positive: Apple_Black_Rot olarak tahmin edildi fakat bu yanlış ise,
- False_Negative: Apple_Healthy olarak tahmin edildi fakat bu yanlış ise,

Apple_Healthy sadece kıyaslamak amacıyla seçilmiştir. Veri setinde Apple_Black_Rot dışında 18 sınıf daha bulunmaktadır yani başka sınıflar ile karıştırması mümkündür.

Kesinlik (precision) değeri pozitif olarak tahmin edilmesi gereken değerlerin gerçekten kaç adedinin pozitif olarak sınıflandırıldığını göstermektedir. Duyarlılık (recall) ise pozitif olarak tahmin edilmesi gereken değerlerin ne kadarının pozitif olduğunu göstermektedir. F1-score değeri ise bize kesinlik ve duyarlılığın harmonik ortalamasını vermektedir. Bunun sebebi uç durumların göz ardı edilmesini engellemektir.

Model bazı hastalıkların görüntülerini tanımlarken iyi performans göstermektedir. Model özellikle tomato_early_blight ve potato_late_blight isimli hastalıkların sınıflandırmasında orta düzeyde başarı elde etmiştir.

Bunun sebebi modelin kurgusundan, veri setinin düzgün dağılmamasından, gerçek hayatta da özellikle domates ve patates yapraklarının birbirine benzemesinden dolayı olabilir. Yaprakların birbirine benzemesi öznitelik haritası çıkarılırken sorun oluşturmıştır. Bu yüzden değerler düşük çıkmıştır.

4.2 TEST EDİLEN HER BİR RESİM İÇİN ANALİZ DEĞERLERİ

Şekil 27 de bulunan çilek resminin analiz değerleri Çizelge 5'te verilmiştir. Tablo %98 oranında çileğin sağlıklı olduğunu belirtmektedir. Çizelge 5'te bulunan tablo test edilen her bitki için ortaya çıkmaktadır böylelikle test edilen bitkilerin analiz değerleri de görülebilmektedir. Fakat test edilen resim sayısı çok fazla olduğu için sınırlı sayıda resim ve analiz değeri paylaşılmıştır. Test edilen bitkilerin analiz değerleri -csv olarak kaydedilmektedir bu sayede gerektiği durumlarda test edilen bitkinin analiz değerleri incelenebilmektedir. Farklı bir bitki resmi için Şekil 28'de gösterilmiştir.



Şekil 27. Test edilen bitki örneği.

Çizelge 5. Çilek bitkisinin hastalık oranı

Test Edilen Resim	Strawberry_Healthy2.JPG
Apple_Black_Rot	0.0021
Apple_Cedar_Rust	0
Apple_Healthy	0
Apple_Scab	1.9931
Corn_Common_Rust	0
Corn_Gray_Leaf_Spot	0
Corn_Healthy	0
Corn_Northern_Leaf_Blight	0
Peach_Bacterial_Spot	0
Peach_Healthy	0
Potato_Early_Blight	0
Potato_Healthy	0
Potato_Late_Blight	0
Strawberry_Healthy	98.0046
Strawberry_Leaf_Scorch	0.0001
Tomato_Bacterial_Spot	0
Tomato_Early_Blight	0
Tomato_Healthy	0
Tomato_Late_Blight	0
Sonucunuz	Strawberry_Healthy

Test edilen resim Strawberry_Healthy2.JPG için analiz değerler incelendiğinde %98 oranında Strawberry_Healthy olarak görülmüştür. Eser miktarda diğer diğer sınıflardan olasılıklar meydana geldiği saptanmıştır.

İlerleyen sayfalarda çeşitli örneklerde farklı bitki türleri için tahmin ve analiz değerleri verilecektir. Bu analiz değerleri her bitki için ortaya çıkmaktadır fakat bu tez de sınıflı sayı da bitki için gösterilebilmektedir.

Şekil 28’de verilmiş olan Tomato_Bacterial_Spot(5).JPG isimli resmin analiz değerleri Çizelge 6’da gösterilmiştir. Tahmin edilen sınıf ile gerçek resim eşleşmektedir. Şekil 29’ da ise mısır örneği verilmiştir fakat tahmin edilen ve gerçek birbiriyle eşleşmemektedir. İlgili analiz değeri Çizelge 7’de incelendiğinde başka bir hastalık türünün olasılığını daha fazla olduğu görülmektedir.

Tahmin: Tomato_Bacterial_Spot
Gerçek : Tomato_Bacterial_Spot



Şekil 28. Test edilen bir domates bitkisi örneği.

Çizelge 6. Domates bitkisinin analiz değerleri

Test Edilen Resim	Tomato_Bacterial_Spot(5).JPG
Apple_Black_Rot	0.0032
Apple_Cedar_Rust	0.0180
Apple_Healthy	0.0017
Apple_Scab	0
Corn_Common_Rust	0
Corn_Gray_Leaf_Spot	0
Corn_Healthy	0
Corn_Northern_Leaf_Blight	0
Peach_Bacterial_Spot	0
Peach_Healthy	0
Potato_Early_Blight	0.0001
Potato_Healthy	0.0019
Potato_Late_Blight	0.1566
Strawberry_Healthy	0
Strawberry_Leaf_Scorch	0.0001
Tomato_Bacterial_Spot	99.1699
Tomato_Early_Blight	0.1461
Tomato_Healthy	0.0089
Tomato_Late_Blight	0.017
Sonucunuz	Tomato_Bacterial_Spot

Tahmin: Corn_Northern_Leaf_Blight
Gerçek : Corn_Gray_leaf_spot



Şekil 29. Yanlış sınıflandırılan bir mısır örneği.

Çizelge 7. Mısır bitkisinin analiz değerleri

Test Edilen Resim	Corn_Gray_leaf_spot4.JPG
Apple_Black_Rot	0.0056
Apple_Cedar_Rust	0.0127
Apple_Healthy	0.0003
Apple_Scab	0.0001
Corn_Common_Rust	0.0068
Corn_Gray_Leaf_Spot	8.3984
Corn_Healthy	0.0230
Corn_Northern_Leaf_Blight	90.7226
Peach_Bacterial_Spot	0
Peach_Healthy	0
Potato_Early_Blight	0.0001
Potato_Healthy	0.0015
Potato_Late_Blight	0.003
Strawberry_Healthy	0
Strawberry_Leaf_Scorch	0.79
Tomato_Bacterial_Spot	0.0022
Tomato_Early_Blight	0.0017
Tomato_Healthy	0.0013
Tomato_Late_Blight	0.0035
Sonucunuz	Corn_Northern_Leaf_Blight

Çizelge 8. Test edilen bitkilerinin analiz tablosu

	Apple_Black_Rot1	Apple_Black_Rot2	Apple_Black_Rot3
Apple_Black_Rot	0.9999	0.9998	0.9955
Apple_Cedar_Rust	0	0	0
Apple_Healthy	0	0	0
Apple_Scab	0	0	0
Corn_Common_Rust	0	0	0
Corn_Gray_Leaf_Spot	0	0	0
Corn_Healthy	0	0	0
Corn_Northern_Leaf_Blight	0	0	0
Peach_Bacterial_Spot	0	0	0
Peach_Healthy	0	0	0
Potato_Early_Blight	0	0	0
Potato_Healthy	0	0	0
Potato_Late_Blight	0	0	0
Strawberry_Healthy	0	0	0
Strawberry_Leaf_Scorch	0	0	0
Tomato_Bacterial_Spot	0	0	0
Tomato_Early_Blight	0	0	0
Tomato_Healthy	0	0	0
Tomato_Late_Blight	0	0	0

Test edilen her bitki analiz değerleri çıktıktan sonra Çizelge 8'deki gibi eklenmektedir. Böylelikle verilerin saklanması ve istenilen zamanda incelenmesi amaçlanmıştır. Tablodaki satırlar hastalık sınıflarını göstermektedir. Sütunlar ise test edilen bitkileri temsil etmektedir. Test edilen resim sayısı 1900 olduğu için tez aşamasında yalnızca Çizelge 7'de gösterilen kısım koyulabilmektedir. Çizelgede her sınıf için 100 resim bulunmaktadır. Tahmin edilen değerler dışında kalan olasılıkların 0 olmasının sebebi eser miktarda bulunmalarından dolayıdır.

5. SONUÇ VE ÖNERİLER

Yapay zeka ve makine öğrenmesi çalışmalarının ilerlemesiyle çeşitli uygulama alanları ortaya çıkmıştır. Yapılan derin öğrenme yaklaşımları ile bitkilerde bulunan hastalıklar başarılı ve hızlı bir şekilde sınıflandırılmıştır.

Bu tez çalışmasında bitki yapraklarındaki hastalıkların derin öğrenme tabanlı bir yaklaşım ile sınıflandırılması gerçekleştirilmiştir. Veri seti oluşturulurken çeşitli veri setleri incelenmiş ve amaca uygun şekilde düzenlenip birleştirilmiştir. Veri setini eğitirken hazır model kullanmak yerine en baştan ESA tabanlı bir model hazırlanmıştır. Hazırlanan modelde evrişim katmanı, aktivasyon katmanı, havuzlama katmanı ve tam bağlı katman bulunmaktadır. Evrişim katmanında filtreler katman sayısı arttıkça artmaktadır. Filtrelerin matris boyutu ise ilk katmandan itibaren azaltılmıştır. Havuzlama katmanında ise maksimum havuzlama seçilmiştir. Dropout katmanı eklenerek aşırı öğrenme durumunun önüne geçilmesi amaçlanmıştır. Aktivasyon fonksiyonu olarak ReLU tercih edilmiştir. Kurgulanan model önce küçük veri setlerinde uygulanmıştır. Ortaya çıkan değerler ışığında model incelenip eksiklikleri giderilmiştir. Adım adım işlem uygulanarak veri seti ve model son halini almıştır. Eğitim başarı oranı %93,95 ‘tir.

Test aşamasında her sınıf için oluşturulan test resimleri içeri aktarılıp boyutları düzenlenmiştir. Yapılan testin başarı oranı %89,89’ dur. Her test resmi için analiz değerleri bulunmuştur. Bulunan analiz değerleri ile bir tablo oluşturulmuştur.

Yapılan çalışmanın proje kısmında sistem yetersizliğinden dolayı küçük bir veri seti kullanılmıştır. Yapay zeka çalışmalarına uygun bilgisayarlar kullanılarak daha büyük veri setleri eğitilebilir ya da daha büyük modeller kurgulanabilir. Çeşitli bitki modelleri ekleyerek oluşturulan modelin geliştirilmesi mümkündür. Modelin katmanları değiştirilip yeni modeller oluşturulabilir.

Sonraki alıřmalarda hazırlanan modelin ağırlıkları kullanarak farklı veri setleri transfer ğrenimi ile eğitilebilir. Böylelikle zaman ve performans açısından ciddi bir kazanç sağlanabilir. Transfer ğrenimi için hazır modeller mevcuttur. Uygun model seçilerek sorunun özümüne uygun şekilde hazırlanabilir.

Veri setini oluşturulurken yüksek özünürlük için daha gelişmiş kamera ve İHA' lar kullanılabilir. Yapılan alıřma için mobil uygulama yapılabilir. Yapay zeka uygulamaları akıllı telefonlarla veya tabletler ile alışabilmektedir. Alınan görüntüler anlık olarak iletilebilir ve değeriendirilebilir. Bilgisayara gerek kalmadan herkesin kullanabileceğı bir sistem geliştirilebilir.

6. KAYNAKÇA

- [1] **Pirim, H. ,** (2006), *TEKNOLOJİ BAĞIMLI YAŞAMIN MATEMATİKSEL DESENLERİ-I*, Journal of Yasar University,81
- [2] **Arivazhagan, S., Shebiah, R. N., Ananthi, S., & Varthini, S. V.** (2013). Detection of unhealthy region of plant leaves and classification of plant leaf diseases using texture features. *Agricultural Engineering International: CIGR Journal*, 15(1), 211-217.
- [3] **Sladojevic, S.,** (2016). Deep neural networks based recognition of plant diseases by leaf image classification. *Computational intelligence and neuroscience*
- [4] **Mohanty, S. ,** (2016). Using deep learning for image-based plant disease detection. *Frontiers in plant science*, 7, 1419.
- [5] **Fuentes, A., Yoon, S., Kim, S., & Park, D.** (2017). A robust deep-learning-based detector for real-time tomato plant diseases and pests recognition. *Sensors*, 17(9),
- [6] **Turkoğlu, İ. ,** (2018).Derin Öğrenme Modelleri ve Uygulama Alanlarına İlişkin Bir Derleme,*DÜMF Mühendislik Dergisi*
- [7] **Walleign S., Polceanu M., Buche C.**(2018). Soybean Plant Disease Identification Using Convolutional Neural Network. In *The Thirty-First International Flairs Conference*.
- [8] **Saleem M. , Potgieter J . & Arif, K.** (2019). Plant Disease Detection and Classification by Deep Learning, *Mdpi*
- [9] **Doğan M .,** (2019), Bitkilerde görülen hastalıkların derin öğrenme yöntemleriyle tespiti ve sınıflandırılması , Yüksek Lisans Tezi, Yalova Üniversitesi, Fen Bilimleri Enstitüsü, Bilgisayar Mühendisliği Anabilim Dalı
- [10] **Hanbay D.** 2019. Plant disease and pest detection using deep learning-based features. *Turkish Journal of Electrical Engineering and Computer Sciences* 27, 2019.

[11] **Aslan, M.** , (2021).Derin Öğrenme ile Şeftali Hastalıklarının Tespiti. *Avrupa Bilim ve Teknoloji Dergisi*, 2021 , 540-546.

[12] **Öğrenme hakkında** : <https://bilgezone.com/ogrenme-nedir/>

[13] **Yapay zeka tarihçesi** : <https://www.derinogrenme.com/2017/03/04/yapay-sinir-aglari/>

[14] **Kurzweil R.**, (1992),The age of intelligent machines, *Cambridge, MA: MIT Press*.

[15] **Makine öğrenmesi ile ilgili olarak** : <https://medium.com/t%C3%BCrkiye/makine-%C3%B6%C4%9Frenmesi-nedir-20dee450b56e>

[16] **Eğrioğlu, E. , Günay, S.** , (2007), Tek Değişkenli Zaman Serileri Analizine Giriş; Hacettepe Üniversitesi Yayınları.

[17] **Mc Culloch, W. , Pitts, W.** , (1943) , A logical calculus of the ideas immanent in nervous activity, 115-133.

[18] **Maltarollo, V. G., Honório, K. M., & Da Silva, A. B. F.** (2013). Applications of Artificial Neural Networks in Chemical Problems. London: Intechopen.

[19] **Öztürk, K. , Şahin, M.E.** , A General View of Artificial Neural Networks and Artificial Intelligence , Takvimi-Vekayi, 28.

[20] **Yapay sinir hücresi** : <http://kod5.org/yapay-sinir-aglari-ysa-nedir/>

[21] **Yüksek, A.G., 2007**, SİVAS İLİNDE YAPAY SİNİR AĞLARI İLE HAVA KALİTESİ MODELİNİN OLUŞTURULMASI ÜZERİNE BİR UYGULAMA, C.Ü. İktisadi ve İdari Bilimler Dergisi, Cilt 8, Sayı 1, 2.

[22] **Tan Z.**, (2019), Derin öğrenme yardımıyla araç sınıflandırma, Yüksek Lisans Tezi ,Fırat Üniversitesi Fen Bilimleri Enstitüsü

[23] **Geleneksel mimarisi** : <https://stanford.edu/~shervine/l/tr/teaching/cs-230/cheatsheet-convolutional-neural-networks>

- [24] Genel mimarisi : <https://www.learndatasci.com/tutorials/convolutional-neural-networks-image-classification/#EvaluatingOurNetwork>
- [25] Dolgu işlemi : <http://www.ashokrahulgade.com/skills/DL/CNN/Introduction.html>
- [26] ReLU grafiği : <https://stanford.edu/~shervine/l/tr/teaching/cs-230/cheatsheet-convolutional-neural-networks>
- [27] Aktivasyon Fonskiyonları: <https://ayyucekizrak.medium.com/derin-%C3%B6%C4%9Frenme-i%C3%A7in-aktivasyon-fonksiyonlar%C4%B1n%C4%B1n-kar%C5%9F%C4%B1la%C5%9Ft%C4%B1r%C4%B1lmas%C4%B1-cee17fd1d9cd>,
- [28] Havuzlama Katmanı : <https://www.learndatasci.com/tutorials/convolutional-neural-networks-image-classification/#EvaluatingOurNetwork>
- [29] Düzleştirme işlemi : <https://medium.com/@tuncerergin/convolutional-neural-network-convnet-yada-cnn-nedir-nasil-calisir-97a0f5d34cad>
- [30] Dropout Katmanı : <http://derindelimavi.blogspot.com/2018/01/bir-derin-ogrenme-deneyi-dropout-ve.html>
- [31] Kütüphaneler Hakkında : <https://miracozturk.com/python-kutuphaneleri-ve-ozellikleri/>

7. ÖZGEÇMİŞ

Ad Soyad : Semih KUŞCU
Doğum Tarihi : 18/12/1995
Doğum Yeri : LÜLEBURGAZ

ÖĞRENİM HAYATI :

İlkokul : Ahmetbey Atatürk İlköğretim Okulu
Lise : Lüleburgaz Lisesi
Üniversite : Bursa Teknik Üniversitesi, Mekatronik Mühendisliği, 2014-2021
Staj : Asparat Ar-Ge , Bursa

ADRES BİLGİLERİ :

Adres : Karşıyaka Mahallesi, Öğretmen Mehmet Ali Yolagelmez Sokak No :5
Ahmetbey/Lüleburgaz/Kırklareli

Telefon : 5389487440

Email : semihkusc17@gmail.com