T.C.
VAN YÜZÜNCÜ YIL ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
YAPAY ZEKA VE ROBOTİK ANABİLİM DALI

ÇİLEK MEYVESİ GELİŞİMİNİN DERİN ÖĞRENME METOTLARIYLA KARŞILAŞTIRMALI İNCELENMESİ

YÜKSEK LİSANS TEZİ

Levent DALGIN
Danışman: Doç. Dr. Murat CANAYAZ

VAN – 2023

T.C.
VAN YÜZÜNCÜ YIL ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
YAPAY ZEKA VE ROBOTİK ANABİLİM DALI

ÇİLEK MEYVESİ GELİŞİMİNİN DERİN ÖĞRENME METOTLARIYLA KARŞILAŞTIRMALI İNCELENMESİ

YÜKSEK LİSANS TEZİ

Levent DALGIN

VAN – 2023

KABUL VE ONAY SAYFASI

Van Yüzüncü Yıl Üniversitesi Fen Bilimleri Enstitüsü, Yapay Zeka ve Robotik Anabilim Dalı'nda Doç. Dr. Murat CANAYAZ danışmanlığında, Levent DALGIN tarafından sunulan "Çilek Meyvesi Gelişiminin Yapay Zeka Metodları İle Karşılaştırmalı İncelenmesi" başlıklı bu çalışma Lisansüstü Eğitim ve Öğretim Yönetmeliği'nin ilgili hükümleri gereğince ….. /..... /..... tarihinde aşağıdaki jüri tarafından oy birliği / oy çokluğu ile başarılı bulunmuş ve yüksek lisans tezi olarak kabul edilmiştir.

Başkan: İmza:

Üye: İmza:

Üye: İmza:

Üye: İmza:

Üye: İmza:

Fen Bilimleri Enstitüsü Yönetim Kurulu'nun …. /.... /…... tarih ve ..... sayılı kararı ile onaylanmıştır.

İmza

……

Enstitü Müdürü

ETİK BEYAN

Tez içindeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edilerek sunulduğunu, ayrıca tez yazım kurallarına uygun olarak hazırlanan bu çalışmada bana ait olmayan her türlü ifade ve bilginin kaynağına eksiksiz atıf yapıldığını bildiririm.

İmza
Levent DALGIN

ÖZET

ÇİLEK MEYVESİ GELİŞİMİNİN DERİN ÖĞRENME METOTLARIYLA KARŞILAŞTIRMALI İNCELENMESİ

DALGIN, Levent
Yüksek Lisans Tezi, Yapay Zeka ve Robotik Anabilim Dalı
Danışman: Doç. Dr. Murat CANAYAZ
Haziran 2023, 93 sayfa

Tarımda yapay zeka uygulamalarının kullanımı her geçen gün artmakta, buna paralel olarak bilgisayarlı görü dünyasında geliştirilen modeller ile ürün tespit ve takibi daha doğru sonuçlar vermektedir.

Ülkemizde Ege, Akdeniz ve Marmara bölgelerinde yoğun yetiştirilen, ayrıca seralarda da çok geniş yetişme alanı bulunan çilek meyvesi dar bir hasat zamanına sahiptir. Hasat (derim) için çileğin hasat zamanının geciktirilmesi çileğin yumuşamasına, erken hasadı ise verimin düşmesine neden olmaktadır. Çilek gelişim evrelerinin yapay zeka destekli görüntü işleme teknikleriyle takip edilmesi, olgunlaşan çileklerin tespit edilerek bunların zamanında hasat edilmesini kolaylaştıracaktır.

Çilek meyvesinden sabit zaman aralıklarında alınan görüntülerin Yolo, FasterCNN ve SSD görüntü işleme algoritmalarıyla olgunlaşma tespiti yapılarak hangi algoritmanın daha hızlı ve doğru sonuçlar verdiği bilimsel veriler ışığında grafiksel olarak karşılaştırılmıştır.

Cep telefonundan farklı zaman ve benzer açılardan alınan görüntüler kullanılarak nesne algılama çalışmaları için veri seti oluşturulmuştur. Evrişimsel sinir modelleri dünyası sürekli artan ihtiyaçlara binaen daha hızlı ve sağlıklı sonuçlar üreten sistemler geliştirmektedir. Evrişimsel sinir ağları modellerinde başarının artması tarımda yapay zeka uygulamalarını destekleyecek ve insansız tarım, hasat robotları gibi teknolojilerin daha doğru çalışmasını sağlayacaktır. Hazırlanan bu tez evrişimsel sinir ağları modellerinin başarı analizlerini matematiksel olarak hesaplayıp çıkan sonuçları grafik ve tablolarla anlaşılır bir şekilde karşılaştırmıştır. Sıkça kullanılan sinir ağı modellerinin deneysel analizlerle elde edilen sonuçlarının karşılaştırmalı grafikleri bu alanda yapılan çalışmalara katkı sağlayacaktır.

Anahtar kelimeler: Bitki gelişimi, FasterCNN, Görüntü işleme, OpenCv, Python, SSD, Yolo, Yapay zeka

ABSTRACT

COMPARISON OF DEVELOPMENT STRAWBERRY FRUIT WITH DEEP LEARNING METHODS

DALGIN, Levent
M.Sc. Thesis, Department of Artificial Intelligence and Robotics
Supervisor: Assoc. Prof. Dr. Murat CANAYAZ
June 2023, 93 pages

The use of artificial intelligence applications in agriculture is increasing every single day, and in parallel with this, the models developed in the world of computer vision provide more accurate results in product detection and tracking.

Strawberry fruit, which is grown intensively in the Aegean, Mediterranean and Marmara regions of our country, and also has a wide growing area in greenhouses, has a narrow harvest time. Delaying the harvest time of the strawberry for harvest (my skin) causes the strawberry to soften, and the early harvest causes the efficieny to decrease. Following the strawberry development stages with artificial intelligence supported image processing techniques will facilitate the detection of ripe strawberries and their timely harvest.

The ripening detection of the images taken from the strawberry fruit at fixed time intervals was made with Yolo, FasterCNN and SSD image processing algorithms, and which algorithm gave faster and more accurate results was compared graphically in the light of scientific data. A data set was created for object detection studies by using images taken from a mobile phone at different times and from similar angles The world of convolutional neural models develops systems that produce faster and healthier results based on ever-increasing needs. Increasing success in convolutional neural network models will support artificial intelligence applications in agriculture and will enable technologies such as unmanned agriculture and harvesting robots to work more accurately. This thesis prepared mathematically calculates the success analysis of convolutional neural network models and compares the results with graphs and tables in an understandable way. Comparative graphics of the results obtained by experimental analysis of frequently used neural network models will contribute to the studies in this field.

Keywords: Artificial intelligence, Development, FasterCNN, Image processing, OpenCv, Python, Plant SSD, Yolo

TEŞEKKÜR

Tez çalışmam boyunca bilgi ve birikimiyle her zaman olumlu, yol gösterici ve motive edici katkılarından dolayı değerli danışmanım Doç Dr. Murat CANAYAZ'a teşekkür ederim.

Yüksek lisans eğitimim süresince beni her zaman destekleyen başta sevgili eşim Yağmur DALGIN'a çocuklarım Hamza DALGIN ve Nur Sinem DALGIN olmak üzere tüm aileme özellikle tez çalışması süresince desteklerinden ötürü meslektaşım Muhammed Necip İLBAY'a teşekkür ederim.

2023

Levent DALGIN

İÇİNDEKİLER

Sayfa

ÇİZELGELER LİSTESİ

ŞEKİLLER LİSTESİ

SİMGELER VE KISALTMALAR

Bu çalışmada kullanılmış bazı simgeler ve kısaltmalar, açıklamalarıyla aşağıda sunulmuştur.

Kısaltmalar	Açıklama
ANN	Artificial Neural Network
BSS	Biyolojik Sinir Ağı
CNN	Convolutional Neural Network
CRNN	Convolutional Recurrent Neural Network
DNN	Deep Neural Network
FCNN	Fully Convolutional Neural Network
FP	False Positive
GAN	Generative Adversarial Network
HMM	Hidden Markov Model
IoT	Internet of Things
KNN	K-Nearest Neighbors
LMT	Logistic Model Tree
LSTM	Long Short-Term Memory
RNN	Recurrent Neural Network
SSD	Single Shot MultiBox Dedector
SVM	Support-Vector Machine
TN	True Negative
TP	True Positive
YOLO	You Only Look Once
YSA	Yapay Sinir Ağı

1. GİRİŞ

Tarımda yapay zeka uygulamaları gün geçtikçe yaygınlık kazanmaktadır. Ülkemizde tarımda yapay zeka uygulamalarının gelişimine bakıldığında ise özellikle son yıllarda giderek öneminin artığı görülmektedir. Ülkemiz su kaynakları ve coğrafi olarak düşünüldüğünde istenen seviyede bir tarım hacminin olmadığı görülmektedir. Tarımsal sorunların giderilmesi için güncel çözümlerin üretilmesi gerekmektedir. Bu noktada dünyada gün geçtikçe yaygınlık kazanan tarımda yapay zeka teknolojilerin verimli ve bilimsel verilerle kullanılması önem arz etmektedir (TBD, 2019).

İnsan gözünün gördüğü ve beyinde bir obje halinde algıladığı görüntülerin makinelerin anlayacağı şekilde manipüle edilmesine görüntü işleme denmektedir (Castelman, 1996). Günümüzde görüntü işleme güncel hayatta birçok farklı alanda kullanılmaktadır. Bunlar, savunma, tarım, tıp alanları, sanayinin farklı kolları vb. olarak karşımıza çıkmaktadır. Tarım alanlarında ise gün geçtikçe daha çok yaygınlık kazanmakla birlikte, bitki hastalıkları ve zararlıların tespitinde, dekolte hesaplamalarında, kalite ölçümlerinde kullanılmaktadır (Keefe, 1992).

İnsanlar tarafından yapılan bitki hastalığı tespiti veya meyve olgunluğu tespit işlemlerinde hata payı %5-10 arasında değişmekte iken yapay zeka algoritmaları tarafından yapılan tespit işlemlerinde hata payı ilk yıllarda %11'lerde iken bu oran günümüzde insanlar tarafından yapılan tespit hata oranlarının altına düşmüştür.

Tarım uygulamaları içerisinde önemi gün geçtikçe yaygınlaşan örtü altı tarım uygulamaları, üretimde verimliliği arttırdığı gibi, iklimsel değişimlerden etkilenmemesiyle ülke ekonomilerine de daha fazla katkı sağlamaktadır.

Örtü altı tarım uygulamalarında en yaygın üretilen türlerden biri olan çilek meyvesi, çabuk yetişmesi, aynı yıl art arda hasat alınabilmesi, tüm coğrafyalarda yetişme alanı bulması vb. sebeplerle en çok tercih edilen türler arasında yer almaktadır. Çileğin örtü altı tarımda yetiştirilmesinin yaygınlık kazanmasıyla birlikte meyve olgunluğu tespitinin zamanında ve yüksek doğrulukla yapılması da önem arz etmektedir. Literatür taraması yapıldığında derin öğrenme ile bitki gelişimi konulu çalışmaların yetersiz olmakla birlikte özellikle son yıllarda birçok çalışmanın yapılamaya başlandığı görülmektedir.

Bu çalışma dünyada yetiştirilme alanı çok geniş olan, örtü altı tarımda da çokça tercih edilip dört mevsim hasadı yapılan çilek meyvesine ait gelişim dönemlerinin farklı yapay zeka modelleriyle karşılaştırması yapılmıştır. Çilek meyvesine ait anlaşmalı bir seradan topladığımız görüntülerden oluşturduğumuz veri seti ile çilek olgunluğunun tespitinde farklı algoritmaların hız ve doğruluk karşılaştırmaları yapılmıştır. Çalışmada literatüre katkı sunması için çilek meyvesi gelişimi baz alınarak Yolov5, FasterCNN ve SSD dedection algoritmalarının karşılaştırmaları deneysel sonuçlarla tespit edilmiştir.

2. KAYNAK BİLDİRİŞLERİ

Çilek, dünyada çok çeşitli ekolojik koşullarda yetiştirilen meyve türlerinden biridir. Çilek üretimi miktarı dünya da her geçen yıl artmaktadır. Çilek yetiştiriciliği genellikle geleneksel tarımla toprakta yapılmaktadır. Geleneksel tarımla yapılan üretim birçok hastalığa neden olmaktadır. Geleneksel tarımda üretim verimliliği düşük seviyelerde olmaktadır. Seralarda topraksız tarımda yapılan üretimlerde verim çok yüksek seviyelere çıkmaktadır. Çilek bitkileri seralarda yetiştirilirken büyümesi çok hızlı olmaktadır. Buda çilek bitkilerinin ihtiyaçlarının çok hızlı değişmesi anlamına gelmektedir. Bu durumun tespiti ve verilecek tepki ne kadar hızlı olursa çileğin verimi o kadar yüksek olmaktadır. Bu durum insan gücü yerine teknolojik imkanların daha fazla kullanılmasını ve bu yolla elde edilecek verimin daha fazla arttırılmasını gerekli kılmaktadır. Derin öğrenme metotları kullanılarak ihtiyaçları çok hızlı değişen çilek türü meyvelerin olgunluk, hastalık vb. tespiti daha hızlı ve yüksek doğrulukla yapılabilmektedir. Literatürde tarımda derin öğrenme ile yapılan birçok çalışma mevcuttur.

Fujita vd. (2016) salatalık yaprakları üzerine çalışmışlardır. Bu çalışmada ise amaç yapraklardaki hastalık tespitidir. Çalışmada veri setinde kullanılmak üzere 7520 tane örneklem ele alınmıştır. Örneklemede kullanılan fotoğraflarının yarısı kötü koşullarda çekilmiştir. Çalışma sonucunda evrişimli sinir ağları ile %82,3 doğruluk oranı elde etmişlerdir (Fujita vd., 2016).

DeChan ve Wiesner-Hanks ise (2017) tarlada yetişen bitkilerin görüntülerinde ortaya çıkan sınırlı verilerin ve sayısız düzensizliğin zorluklarını ele alan, evrişimli sinir ağlarını (CNN'ler) kullanarak mısır bitkisinde görülen ve kuzey yaprak küfü denilen hastalığının tespiti üzerine çalışma yapmışlardır. Yapılan çalışmada kullanılan veri seti 1796 görüntüden oluşmaktadır. Bu veri setiyle yapılan deneysel çalışma sonucunda %96.7'lik bir doğruluk oranına ulaşmışlardır (DeChan ve Wiesner-Hanks 2017).

Lu vd. (2017) ise pirinç hastalıkları üzerine yoğunlaşmışlardır. Yapılan çalışma yaygın 10tane pirinç hastalığının tespiti üzerinedir. Kullanılan veri seti 500 görüntüden oluşmaktadır. CNN mimari modeli ile %95.48'lik bir doğruluk oranına ulaşmışlardır (Lu vd., 2017).

Mohanty ve Hughes (2016) ise 54306 resimden oluşan bir veri seti kullanmışlardır. Yapılan çalışmada 26 farklı hastalığı tespit etmeye çalışmışlardır. Evrişimli sinir ağlarının kullanıldığı çalışma sonucunda, uzatılmış bir test setinde %99.35 doğruluk oranına ulaşmışlardır (Mohanty ve Hughes 2016).

Brahimi vd. (2018) genel bitki hastalıklarının tespiti üzerine çalışmışlardır. Çalışmalarında evrişimli sinir ağlarından AlexNet ve GoogleNet mimarilerini kullanmışlardır. Çalışmalarında 14828 görüntüden oluşan bir veri seti kullanmışlardır. Çalışmaları sonucunda, AlexNet ile %97.35 ve GoogleNet ile %97.71 doğruluk oranlarına ulaşmışlardır (Brahimi vd., 2018).

Fang vd. (2019) ise elma meyvesine dair hastalık tespiti çalışmalarında CNN modelini kullanmışlardır. Sınıflandırmaları doğru bir şekilde yapmak için CNN modelinde çokça başvurulan merkez kaybı fonksiyonunu kullanmışlardır.

> *"Veri kümesi olarak antraknoz yaprak yanıklığı, sedir pas, yaprak pas, gri nokta, siyah nokta ve siyah çürük hastalığına sahip olan 5373 hastalıklı yaprak görüntüsü ve 1683 sağlıklı yaprak görüntüsü kullanılmıştır. Çalışmada sonuç olarak önerilen VGG16 modeli ile %95- %99,70 arasında doğruluk elde edilmiştir" (Fang vd., 2019).*

Baranwal vd. (2019) tarafından elma yaprağı hastalığı tespiti için gerçekleştirilen çalışmada CNN modeline ait GoogLeNet mimarisi kullanılmıştır.

> *"22 katmandan oluşan bu mimari, PlantVillage veri kümesinin alt kümesi üzerinde doğrulanmıştır. Veri seti olarak siyah çürük, uyuz ve pas hastalığına sahip 1526 hastalıklı yaprak görüntüleri ve sağlıklı yaprak görüntüleri kullanılmıştır. Çalışmada sonuç olarak önerilen GoogLeNet modelinin ortalama %98,42 doğruluk oranına sahip olduğu görülmüştür" (Baranwal vd., 2019).*

Alruwaili vd. (2019) ise yaptıkları bitki hastalığı tespiti çalışmasında yine CNN modelinin barındırdığı AlexNet mimarisinden faydalanmışlardır.

> *"Çalışmada, 14 farklı bitki türü ve 26 farklıhastalığın toplam 54.306 görüntüsü bulunan PlantVillage veri setini kullanmışlardır. Sonuç olarak*

önerilen AlexNet modeli ile %99.11 doğruluk, %99.49 hassasiyet, %99.29 oranında F1 skoru elde ederek AlexNet mimarisinin farklı bitki hastalıklarını etkin bir şekilde saptadığını ortaya koymuşlardır" (Alruwaili vd., 2019).

Bitki hastalığı teşhisinde farklı bir çalışma ise Ferentinos (2019) tarafından geliştirilen derin öğrenme modelleri ile yapılmıştır. Çalışmada 87.000'den fazla görüntüden oluşan veri seti eğitim ve test şeklinde ayrılıp CNN modeli kullanılarak eğitilmiştir.

"Bu görüntüler 25 farklı bitki türünün 58 farklı hastalığını içermektedir. Çalışmada, AlexNet, AlexNetOWTBn, GoogLeNet, Overfeat, VGG modelleri kullanarak en yüksek başarı oranı VGG ile %99.53 olarak elde edilmiştir" (Ferentinos vd., 2019).

Khitthuk vd. (2018) ise üzüm yapraklarındaki hastalıkların tespiti üzerine çalışmışlardır. Çalışmada ARTMAP adlı denetimsiz sinir ağı modeli, oluşturulan veri setindeki hastalık sınıflandırması için kullanılmıştır.

"Veri seti olarak pas, uyuz, tüylü küf hastalığı olan ve hastalıksız toplam 346 üzüm yaprağı görüntüsü kullanılmıştır. Çalışmada sonuç olarak, %90'nın üzerinde doğruluk elde etmişlerdir" (Khitthuk vd., 2018).

Singh ve Misra (2017) ise çalışmalarında üç farklı sorun üzerinde durmuşlardır. Gül ve fasulye yaprakları, limon yaprağı ve muz yapraklarından oluşan üç farklı bitki türüne ait bakteriel hastalıklar, güneş yanığı ve erken yanık hastalığı gibi üç farklı etmen üzerinden tespitini yapmışlardır.

"Çalışmada, önerilen algoritma ile SVM kullanıldığında genel doğruluk %95.71 oranında elde edilmiştir" (Singh ve Misra, 2017).

CNN modeli kullanılarak yapılan farklı bir çalışma ise Sladoyeviç vd. (2016) tarafından yapılmıştır. Kullanılan model ile 13 değişik bitki hastalığı ve sağlıklı bitkiler tespit edilebilmiştir. Veri seti oluşturulurken eğitimlerin sağlıklı ve uygulanabilir

sonuçlar üretmesi için 13 farklı bitki hastalığına ait görüntüler, sağlıklı bitkiler ve arka plan görüntüleri kullanılmıştır.

> *"Veri setinde bulunan 30880 görüntü eğitim, 2589 görüntü test için kullanılmıştır. CNN' in geliştirilmesi için 8 öğrenme katmanı, 5 evrişimsel ve 3 tam bağlı katman içeren Caffe çerçevesi kullanılmıştır. Sonuç olarak model, %91 ile %98 arasında hassasiyet ve %96,3 doğruluk elde etmiştir" (Sladojevic vd., 2016).*

Bu alanda yapılan farklı bir çalışma ise elma ağacına dair yapısal bozukluklarla ilgilidir. Nachtigall vd. (2016) tarafından yapılan çalışmada yine CNN modeli kullanılarak maxigala, pink ve fuji suprema türlerine ait elma ağaçlarına dair yapısal bozukluklar otonom olarak tespit edilmiştir. Çalışmada bu üç türe ait uyuz zararı, magnezyum ve potasyum mineral eksikliği, glomeralla lekesi ile herbisit zararı olmak üzere beş farklı fiziki hasar üzerinde durulmuştur.

> "Veri kümesi olarak Brezilya'nın güney kesimindeki meyve bahçelerinden 1970 hastalıklı yaprak görüntüsü, 569 sağlıklı yaprak görüntüsü kullanmışlardır. CNN ile gerçekleştirilen çalışmada %97.3 doğruluk oranında sonuç elde edilmiştir" *(Nachtigall vd., 2016).*

An vd. (2022) ise yaptıkları çalışmada çilek meyvesine ait gelişim yüzdelerini YOLO modelinin farklı versiyonlarını karşılaştırarak yapmışlardır. Elde ettikleri deneysel Sonuçlar, YOLO modelinde hassasiyetinin, doğruluğunun ve geri çağırmasının farklı versiyonlarda sırasıyla %94.26, %93.15 ve %90.72 olduğunu ve bu hesaplamalardan yola çıkarak ulaşılan sonuca göre kesinlik değerinin doğruluktan yüzde olarak sırasıyla %4.08, %3.64 ve %2.04 daha yüksek çıktığını görülmektedir.

Alanda yapılan farklı bir çalışmada ise Aksoy vd. (2020) bitkilerin kök ve yapraklarından gıda almalarına mâni olup pas diye adlandırılan bir bakteri çeşidinin yapay zeka metotlarıyla tespiti konusunda çalışmalar yürütmüştür.

> *"Çalışmasında zambak çiçeği yaprak görüntülerinde pas hastalığı olup olmadığına dair GLCM ve gabor dalgacık dönüşümü (ing. Gabor Wavelet Transform–GWT) tabanlı sınıflandırıcıları öznitelik çıkarma metodu olarak kullanılarak bir sistem*

tasarlanmıştır. Çalışmada veri seti olarak 32 sağlıklı görüntü, 21 hastalıklı görüntü üzerinde k-NN, çok katmanlı algılayıcı ve en küçük kareler SVM kullanmıştır. Sonuç olarak çalışmasında, GLCM tabanlı k-NNve çok katmanlı YSA ile %88,9 oranında başarı elde ederek bitkideki pas hastalığını doğru tespit etmiştir" (Aksoy vd., 2020).

3. MATERYAL VE YÖNTEM

Makine öğrenme ve bilgisayarlı görü alanında yapılan çalışmalar yapay zeka temelli olduğu için yapay zeka konusunun öncelikle açıklığa kavuşturulması gerekmektedir.

Makine öğrenmesinde başvurulan algoritmalar ve mimarilerin tümü yapay zeka kapsamındadır. Tezin bu bölümünde yapay zeka kapsamında kullanılan mimariler, modeller ve alt bileşenlerine dair bilgilendirmeler bulunmaktadır.

3.1 Yapay Zeka

İnsanoğlu kendisine bahşedilen akıl ve zeka sayesinde etrafında gelişen olayları değerlendirir ve bu olaylara karşı tavır geliştirir. Makineler ise donatıldıkları sensörler ve kameralar ile etraflarındaki nesnelere karşı, mesafe, sıcaklık, görüntü işleme vb. girdiler yardımıyla tepki verme eğilimindedir. Bu şekilde bir tanımlama yapılırsa makinelerinde insanlara kıyasla ilkel bir zekaya sahip olduğu söylenebilir.

Günümüzde makineler IoT (Internet of Things - Nesnelerin interneti) teknolojisi için tanımlanan sensörler ve kameralar yardımıyla etraflarında gelişen olayları tıpkı insan duyuları gibi algılar, veriler toplar, bu verileri işler ve insan faydasına olacak sonuçlar üretirler. İnsanlar içinde bulundukları dünyada kendini savunma, metabolizmanın varlığını devam ettirme, sanat yapma, duygusal bağ kurma yönleriyle günümüzde makinelerin çok ötesinde olsalar da, hesaplama yapabilme, verileri işleyip sonuç çıkarma, hızlıca analizler yapma, çok büyük verileri doğru bir şekilde depolayıp ihtiyaç olduğunda hızlıca ulaşma vb. alanlarda makinelerin yardımına ihtiyaç duymaktadırlar.

Tezin konusu olan görüntü işleme tekniklerinde de aynı şekilde kameralar yardımıyla elde edilen görüntüler toplanmaktadır. Elde edilen görüntülerden oluşan veri setleri tıpkı deneysel çalışmalardaki kontrol ve deney grupları gibi eğitim ve test şeklinde kategorize edilmektedir. Toplanan veriler işlenmekte ve işlenen veri setleri ile makinelerin sonuç üretmesi yani karar vermesi sağlanmaktadır.

1950'li yıllarda matematikçi ve bir bilim adamı olan Turing ile başlayan yapay zeka serüveni, Turing testleriyle düşünebilen, karar verebilen makinelerin üretilmesi

fikri ile devam etmiştir. O yıllardan beri popülerlik kazanan yapay zeka kavramı, günümüzde konuşma, görme, karar verme, bilgi işleme amaçlı çalışmalarla hız kazanmıştır (Dawson, 2007).

İnsanoğlu gelişen bir olay karşısında karar verirken önceki edindiği tecrübelerden faydalanır. Bir nesneye dokunurken onun sıcak veya soğuk olduğuna önceki tecrübelerine dayanarak karar verir. Yine insan ön öğrenmeler yardımıyla hayatını dizayn eder. Öğrenciler okullarda derslere girer ve hazır bulunurluk seviyelerine göre sınavlara tabi tutulur. Yani bir soruya cevap verirken o konuyla ilgili ön öğrenmelerinden faydalanır. Yapay zekada aynı mantıkla işler, önceden işlediği verilere göre karar verir.

Aşağıdaki Şekil 3.1'de yapay zeka ve kapsamındaki alt bileşenler hiyerarşik bir şekilde şematize edilmiştir. İnsanların düşünme, fikir yürütme ve karar verme süreçlerinin makineler tarafından yapılabilmesi çalışmaları genel olarak yapay zeka olarak tanımlanmaktadır.

> *"Ancak makine öğrenmesi çalışmaları ve algoritmaları daha alt alanı kapsamaktadır. Makine öğrenmesinin en dikkat çekici uygulamalarını içeren evrişimsel sinir ağları ise derin öğrenme alt alanında yer alan çalışmaları oluşturur" (Piccinini 2004).*

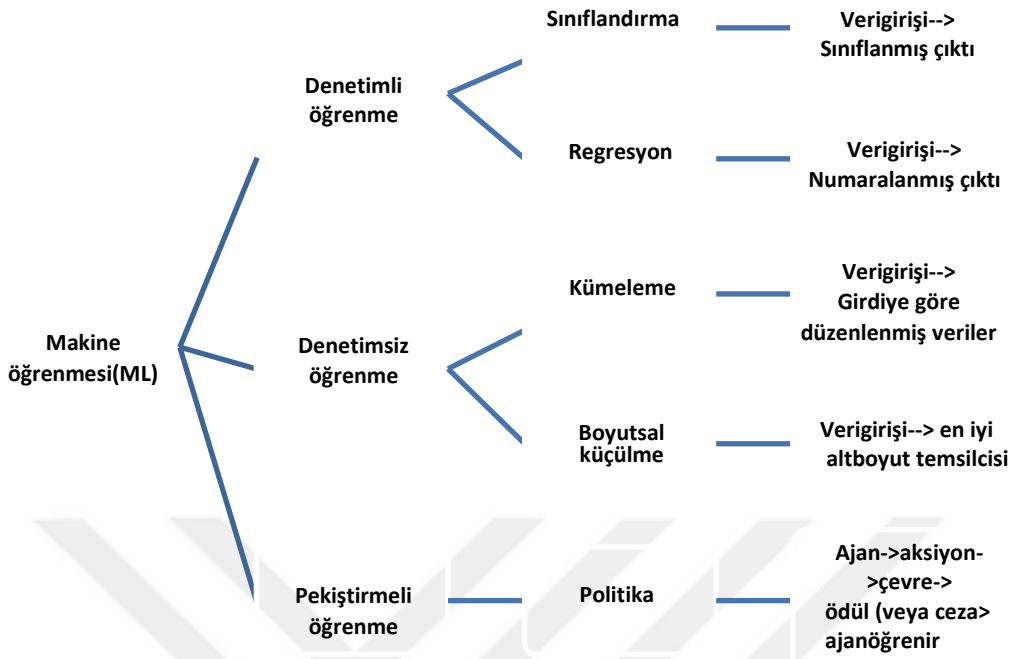
Şekil 3.1 Yapay zeka ve alt bileşenleri (Anonim, 2023a)

3.2 Makine Öğrenmesi

Yapay zekanın bir alt işlem basamağı olan makine öğrenmesi, yapay zekanın tanıma, öğrenme gibi işlemlerini yerine getirmek için başvurduğu uygulamalar olarak tanımlanmaktadır. Makine öğrenmesi insanların ön bilgilerine dayalı karar verme mekanizmaları örneğinde olduğu gibi bilgiyi işler ve farklı algoritmalar yardımıyla karşılaştırma yaparak en doğru sonuca ulaşmaya çalışmaktadır.

Yapay zeka algoritmaları alışık olduğumuz otonom sistemlerdeki sıralı görevleri yerine getirmekten çok oluşturulan veri setinden edindiği tecrübe ile karşılaştığı yeni ham verileri tanıma, bunlarla işlem yapma, öğrendiklerini yeni durumlarda kullanabilme şeklinde tanımlanmaktadır. Kullanılan algoritmaların geliştirilerek insansı öğrenme işlevinin makineler tarafından uygulanması makine öğrenmesi olarak tanımlanmaktadır. (Yılmaz, 2021).

Makine öğrenmesi için insan zekasını taklit etmek amacıyla tasarlanmış matematiksel algoritmalar (Costa vd., 2015), ile bir sorunun toplanan verilere uygun bir yapay zeka algoritması ile modellenmesi şeklinde tanımlanabilir (Atalay ve Çelik, 2017). Big Data denilen büyük veri havuzu beraberinde verilere hızlı ve sağlıklı ulaşım gibi büyük problemleri de getirmiştir (Suthaharan, 2014). Classification denilen sınıflandırma tekniği ile büyük veriler kategorize edilmekte ve verilere ulaşım bu şekilde basitleştirilmektedir. Makine öğrenmesine ait alt algoritmaların gösterimi Şekil 3.2'de yapılmıştır.



Şekil 3.2 Makine öğrenmesi algoritmaları hiyerarşisi (Rashidi vd., 2019)

3.2.1 Denetimli Öğrenme

Makine öğrenmelerinin verilerden insan müdahelesine gerek kalmadan öğrenebilen ve yeni deneyimlere yeni çözümler gerçekleştirebilen programlar olduğu belirtilmişti. Makine öğrenmelerinden denetimli makine öğrenimi, belirsizlik dâhilinde eldeki mevcut verilere yani kanıtlara dayalı tahminler yapan bir model oluşturur. Denetimli öğrenme algoritmalarında bilinen bir girdi verisi seti ve bu verilere yine bilinen yanıtları alır, ardından yeni verilere yanıt için makul tahminler oluşturmak üzere bir model eğitilir. Tahmin etmeye çalıştığınız çıktı için bilinen verilerin olduğu durumlarda denetimli öğrenme kullanılabilmektedir (Nizam ve Akın, 2014).

3.2.2 Sınıflandırma Teknikleri

Sınıflandırma, girdi verilerinin kategorize edilmesidir (Tecim vd., 2022). Örneğin yaptığımız çalışmada çilek bitkisine dair resimlerden oluşan veri seti sonuçlandırılırken olgun, yarı olgun, tam olgun ve olgunlaşmamış şeklinde bir sınıflandırmaya yani etiketlemeye gidilir. Algoritmalar verileri bu sınıflandırmaya göre

eğitir ve makine bu sınıflandırmayı temel alarak öğrenir. Eğitim sonrası yeni veriler yine bu sınıflandırmaya göre kategorize edilmiştir.

Sınıflandırma tekniğinin kullanılabilmesi için verilerin kategorize edilebilmesi gerekmektedir. Buna el yazısı tanıma uygulamaları örnek gösterilebilir. El yazısı tanıma algoritmalarında harf, sembol ve sayıları tanımak için sınıflandırma tekniği yaygın olarak kullanılmaktadır.

> *"Sınıflandırma yapmak için yaygın algoritmalar, destek vektör makinesi (SVM), karar ağaçları, k-en yakın komşu, Naive Bayes, diskriminant analizi, lojistik regresyon lojistik regresyon ve sinir ağlarını içermektedir" (Harman, 2021).*

3.2.3 Regresyon Teknikleri

Regresyonun bağımlı değişkenle bağımsız değişken arasındaki ilişkilerin tahmin edilmesine yönelik kullanılan istatistiksel bir araç olduğu bilinmektedir.

Regresyon yöntemlerinin kullandığı temel teknikler mevcut yöntemlerde bazı farklılıklar olmakla birlikte klasik optimizasyon (en iyi değeri bulma) teorisidir.

> *"Bununla birlikte, sayısal sinyal işleme ve diğer sayısal merkezli disiplinlerde karşılaşılan çoğu pratik regresyon problemi, doğrusal denklem sistemlerinin hesabı için en uygun çözümün bulunmasını gerektirir. Dolayısıyla, araştırmacılar doğrusal bir denklem sisteminin veri özelliklerinin tanımlanması için doğrusal regresyon yöntemlerinin daha çok teorik kısımlarını kullanırlar" (Cadzow, 2002).*

Basit doğrusal regresyon ve çoklu doğrusal regresyon, doğrusal regresyonun alt bileşenleridir.

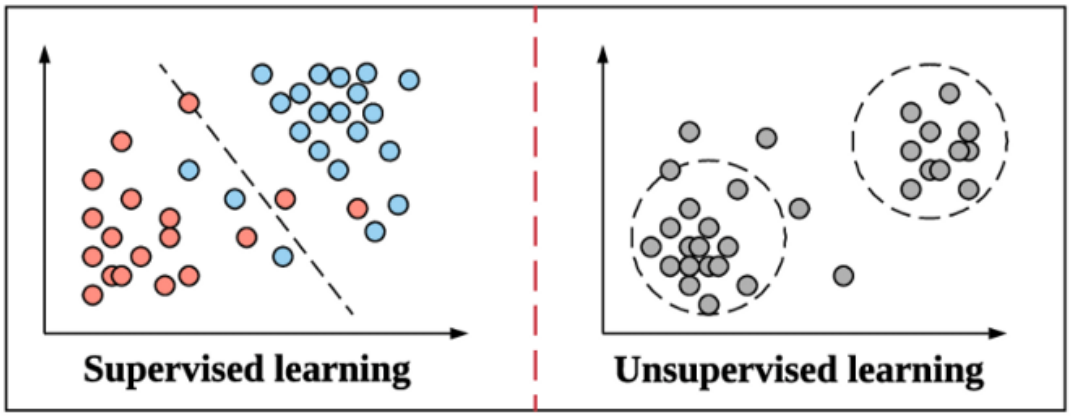
> *"Basit regresyon analizi, yanıt değişkeni ile tek bir açıklayıcı değişken arasındaki doğrusal ilişkiyi açıklar. Eğer tek bir yanıt değişkeni ve birden fazla açıklayıcı değişken arasındaki doğrusal veya eğrisel bir ilişki tanımlanmak istenirse, ilişki çoklu doğrusal regresyon analizi ile incelenir" (Okur, 2009).*

3.2.4 Denetimsiz Öğrenme

Denetimsiz (Unsupervized) öğrenme, oluşturulan verilerin kaç bölüme ayrıldığı, girdi aşamasındaki verilere karşılık gelen çıktı aşamasındaki verilerden anlamsız (etiketsiz) yani tanımsız verilerden anlamlı sonuçlar çıkarma işlemidir. Bu veriler sınıflara yani ortak özelliklere göre tanımlanır. Kümeleme (clustering) yöntemi denetimsiz öğrenme yöntemlerinden biridir (Bulut vd., 2014).

3.2.5 Kümeleme

Kümeleme yöntemi denetimsiz öğrenme söz konusu olduğunda kullanılan yöntemlerden biridir. Sınıflara ayrılmamış, kategorize edilmemiş verilerden bir model veya yapı bulmayla ilgilenir. Bu algoritmalar verilerinizi işleyerek ve veriler üzerinde doğal kümeler (clusters) oluşturur. Ayrıca sınıf sayısı veya veri farklılıklarına göre küme sayısı da değişkenlik arz etmektedir. Oluşan grupların ayrıntı seviyelerinin düzenlenmesine de imkân tanınmaktadır (Atalay ve Çelik, 2017).



Şekil 3.3 Denetimli ve denetimsiz kümeleme modelleri (Hazır, 2021)

3.2.6 K-Kümeleme

K, en yüksek değeri bulmamıza yardımcı olmak için her bir yineleme için başvurulan yinelemeli bir kümeleme algoritması demektir. Başlangıçta, istenilen sayıda küme seçilmektedir. Bu kümeleme yönteminde, veri noktalarının k gruplarına

kümelenmesi gerekmektedir. Daha büyük bir k, aynı şekilde daha fazla ayrıntıya sahip daha küçük gruplar anlamına gelmektedir (Yukselturk vd., 20014).

3.2.7 Hiyerarşik Kümeleme

Hiyerarşik kümelemede ise adından da anlaşılacağı üzere veriler küme ağacı şeklinde sınıflara ayrılmaktadır (Karakaya, 2021).

Yöntem olarak tek, orta ve ortalama bağlantı yöntemlerini kullanan hiyerarşik kümeleme yöntemi yine farklı bir tanım olarak veri noktalarının üst ve alt kümeler halinde kümelenmesidir.

Hiyerarşik kümeleme yönteminde oluşturulan veri etindeki her bir eleman veya veri benzer özellikleri ve veriler arası mesafe düzeyleri dikkate alınmak suretiyle kümeleme sürecinde değişik aşamalarda bir araya gelerek kategorize edilmektedir (Yüceşahin, 2011).

3.2.8 Olasılıksal Kümeleme

İhtimaller üzerinden oluşturulan bir kümeleme yöntemidir (Gaussianların karışımı- mix of gaussians). Veri noktaları olasılıklı bir ölçekte kümeler halinde kümelenir.

3.2.9 Pekiştirmeli Öğrenme

Makine öğrenmesi veya yapay öğrenme veriye dayalı olarak öğrenmeyi mümkün kılan algoritmalara verilen genel bir isimdir. Kendi içerisinde gözetimli (İng. supervised), gözetimsiz (İng. unsupervised) ve pekiştirmeli (İng. reinforcement) öğrenme gibi alt dallara ayrılmaktadır.

Pekiştirmeli öğrenme bir temsilciyi, faydalı faaliyeti ve faydasız faaliyeti tanımlayan net parametrelerin ve ulaşılması gereken kapsayıcı bir oyun sonu olan bir ortama yerleştirir. Süreç içerisinde kullanılan algoritmalara net hedefler belirlenerek ödül ve ceza yönüyle denetimli öğrenmeye benzer bir süreç işler. Bu, gereken açık programlama seviyesinin denetimsiz öğrenmeye göre daha yüksek olduğu anlamına

gelir. Ancak, bu parametreler bir kez ayarlandıktan sonra, algoritma kendi kendine çalışır ve denetimli öğrenme algoritmalarından çok daha kendi kendini yönetir. Bu nedenle, insanlar bazen takviyeli öğrenmeye yarı denetimli öğrenmenin bir dalı olarak atıfta bulunur, ancak gerçekte, çoğu zaman kendi makine öğrenimi türü olarak kabul edilmektedir.

Farklı bir tanım olarak makine öğrenme yaklaşımının daha çok amaca yönelik ne yapılması gerektiğini öğrenme şeklidir. Pekiştirmeli öğrenmede öğrenme arttıkça öğrenen makine pekiştireç sinyali alır ve devamında öğrenen makine daha çok ödül için öğrenme puanını maksimuma çıkarmaya çalışır.

> *"Bu şekilde çalışan deneme yanılma yöntemi bazlı öğrenme, pekiştirmeli öğrenmenin en ayırt edici özelliğidir" (Waslander vd., 2005).*

Pekiştirmeli öğrenme sürecinde Markov denilen bir karar verme modeli kullanılmaktadır.

> *"Markov karar süreçlerinin en önemli 3 özelliği; algılama (sensation), eylem (action) ve hedeftir (goal)" (Sutton ve Barto, 1998).*

Özetle pekiştirmeli öğrenme doğal yaşamdaki canlı iç güdülerinden esinlenir. Yaşam döngüsünün ilk aşamasında hiçbir tecrübesi yokken, sensörlerle donatılmış olan canlılar herhangi bir eğitmen desteği olmadan, yürüme, koşma, tutma, emme, yüzme vb. komplike davranışları öğrenebilmekte ve bunları belleğe alarak ihtiyaç duyduğunda anında uygulayabilmektedir (Sutton ve Barto, 1998).

3.3 Derin Öğrenme

Derin öğrenme sinir ağları veya yapay sinir ağları, insan beynini veri girişleri, ağırlıklar ve yanlılığın bir kombinasyonu aracılığıyla taklit etmeye çalışmaktadır. Bu yapılar veri gruplarındaki nesneleri gerektiği şekilde teşhis etmek, tanımlamak ve kategorize etmek amacıyla entegre çalışırlar.

Derin sinir ağları, tahmin veya sınıflandırmayı iyileştirmek ve optimize etmek için her biri bir önceki katmanın üzerine inşa edilen, birbirine bağlı birden çok düğüm katmanından oluşmaktadır. Hesaplamaların ağ üzerinden bu ilerlemesine ileri yayılım denmektedir. Derin bir sinir ağının giriş ve çıkış katmanlarına görünür *katmanlar* denmektedir. Girdi katmanı, derin öğrenme modelinin verileri işlemek için aldığı yer olarak ve çıktı katmanı, son tahminin veya sınıflandırmanın yapıldığı yer olarak tanımlanmaktadır.

Makine öğrenimi (Machine learning) daha basit yöntemler kullanırken, derin öğrenme, insanların nasıl düşündüğünü ve öğrendiğini taklit etmek için tasarlanmış yapay sinir ağlarıyla çalışmaktadır. Çalışmalar bu kadar ileri seviyeye gelmeden önce sinir ağları bilgi işlem gücüyle sınırlıydı ve bu nedenle karmaşıklığı da sınırlıydı. Bununla birlikte, büyük veri analitiğindeki ilerlemeler, bilgisayarların insanlardan daha hızlı karmaşık durumları gözlemlemesine, öğrenmesine ve bunlara tepki vermesine izin vererek daha büyük, karmaşık sinir ağlarına izin vermiştir. Derin öğrenme, görüntü sınıflandırmasına yardımcı olmaktadır. Bununla birlikte dil çevirisi, konuşma tanıma gibi alanlarda da kullanılmaktadır. Otonom olarak herhangi bir örüntü tanıma sorununu çözmek için ve insan müdahalesi olmadan kullanılabilmektedir.

Birçok katmandan oluşan yapay sinir ağları, derin öğrenmeyi desteklemektedir. Derin Sinir Ağları (DNN'ler), her katmanın görüntü, ses ve metni anlamlandıran temsil ve soyutlama gibi karmaşık işlemleri gerçekleştirebildiği bu tür ağlardır.

Bilgisayarların hayatımıza girdiği andan itibaren işlerimizi kolaylaştırdığı, hesaplanması uzun zaman alan işlemlerin çok kısa sürede hesaplandığı bilinen bir gerçektir. Ancak bilgisayarın ilk zamanlarından beri en çok merak edilen ve üzerinde çalışılan konulardan biri yordama yapabilen, yüz tanıma, nesne algılama ve takip etme, kendi kendine öğrenme işlemleri gerçekleştirebilen yapay zeka tabanlı makineler olmuştur. İnsanlar daha önce öğrendikleri birçok kavramdan yeni karşılaştıkları kavramlar için çıkarsamada bulunabilirler. Yapay zeka işleme algoritmaları da aynı mantıkla çalışarak işlenen verilerden elde ettiği işlenmiş veriyi yeni ham verileri tanımak için kullanmaktadır.

Kavramlar hiyerarşisi mantık olarak bilgisayarlarda karmaşık kavramların daha basit bir şekilde örüntü kurarak tespit edilmesine olanak sağlar. Kullanılan bu

kavramlardan katmanlı yapılar yoluyla oluşan mimari derin öğrenme metodunu oluşturmuştur (Murat, 2021).

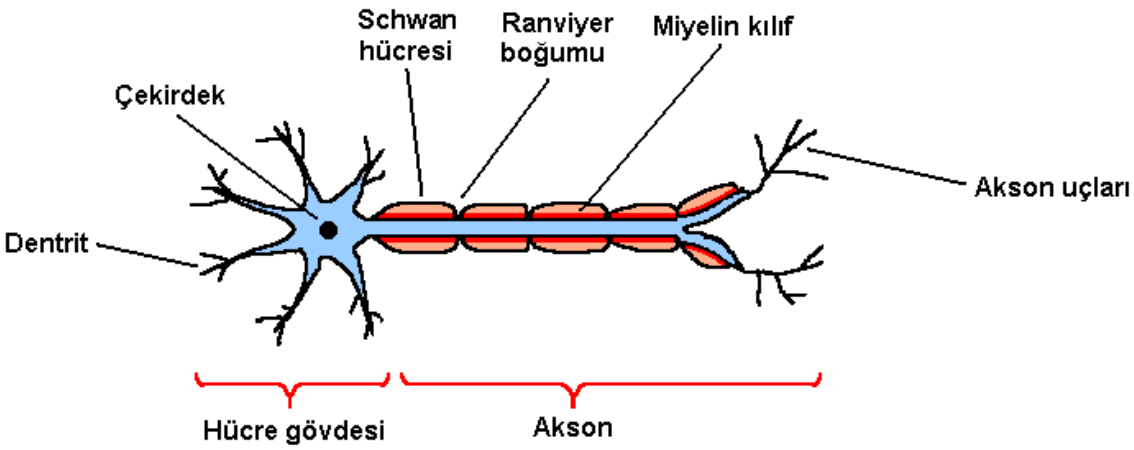
Görüntü işleme teknolojileri derin öğrenmeye dayalı yapılar ve alt bileşenler üzerinden gelişme kaydetmektedir. Bu sebeple bu tez çalışması boyunca makine öğrenmesi (machine learnining) ve yapay zeka (artificial intelligence) yerine çoğunlukla derin öğrenme (deep learning) konusu incelenmiştir.

3.3.1 Yapay Sinir Ağları

Yapay sinir ağlarının yapısını anlamanın ilk koşulu insan sinir hücrelerinin yapısı ve mantığını anlamaktan geçer, çünkü yapay sinir ağları tasarı ve mantığı tıpkı biyolojik sinir ağları mantığına benzemektedir.

Aşağıdaki şekilde bir insan hücresi şematize edilmiştir (Şekil 3.4). Şekilde de görüldüğü gibi bilgisayar sistemlerindeki input-output benzeri bir yapı dentrit-akson kısımları tarafından verilerin alınması ve işlem sonuçlarının çıkması sağlanmaktadır.

Akson uçları yardımıyla çevreden alınan veriler akson yolu boyunca iletilir ve hücre gövdesinde bulunan dentritler aracılığıyla diğer sinir hücrelerine aktarılır. Bu şekilde milyonlarca sinir hücresi bir araya gelerek görme, duyma, algılama vb. işlemlerin yerine getirilmesine yardımcı olmaktadırlar.

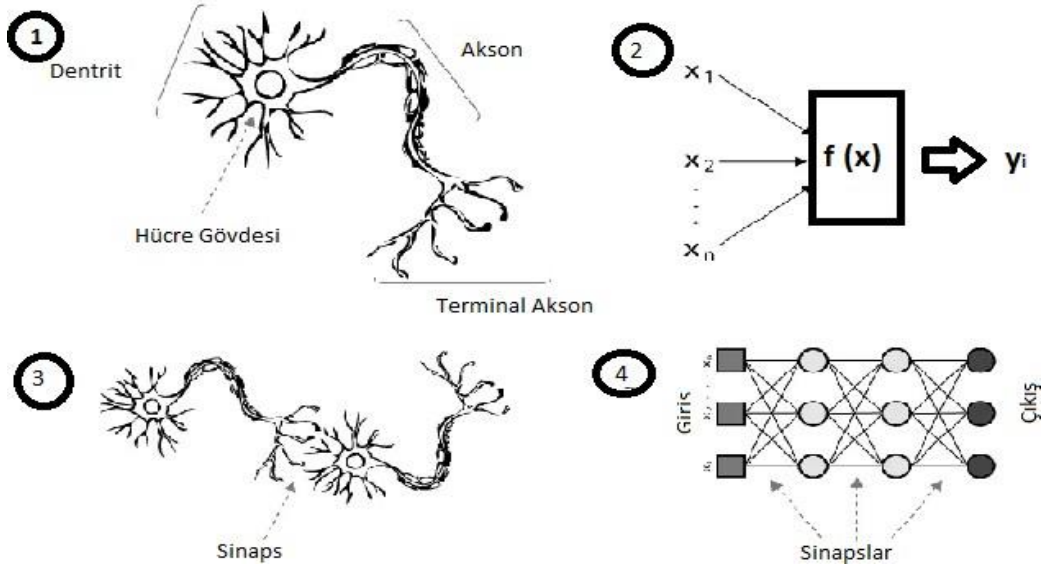


Şekil 3.4 İnsan sinir hücresinin yapısı (Anonim, 2023b)

Yapay sinir ağı (YSA) insan ve hayvan sinir hücreleri gibi biyolojik sinir hücrelerini çalışma mantığı yönüyle taklit etmektedir. Bu yönüyle incelendiğinde ilk yapay sinir ağı mimarisine 1943 yılında bir sinir hekimi Warren Mc Culloch ve çalışma arkadaşı ve aynı zamanda bir matematikçi olan Walter Pitts'in

> *"Sinir Aktivitesinde Düşüncelere Ait Bir Mantıksal Hesap (ALogical Calculus of Ideas Immanent in Nervous Activity)" adlı makalesinde rastlanmaktadır (Egrioğlu ve Aladağ, 2009).*

Biyolojik sinir ağlarının dentrit-gövde-akson şeklindeki basitleştirilmiş yapısının girdi-işlem-sonuç şeklinde şematize edilerek kullanımı günümüzde de devam etmektedir.



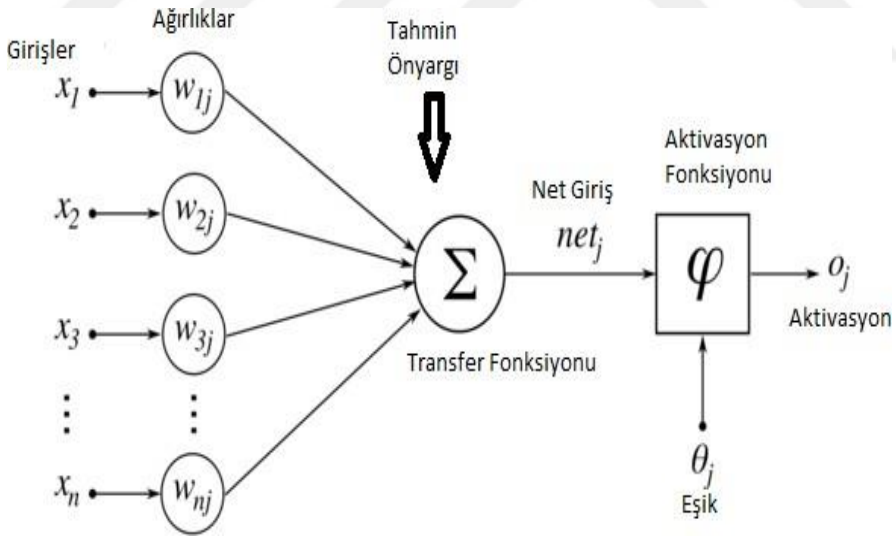
Şekil 3.5 Biyolojik sinir hücresi ve yapay sinir ağı (Maltarollo vd., 2013)

İnsan sinir hücresi ve yapay sinir ağı işlev olarak benzerlik göstermektedir denilmişti. Şekil 3.5'teki benzetimlerde bu benzerlik net bir şekilde görülmektedir. Bununla birlikte biyolojik sinir hücresine ait kısımların yapay sinir ağındaki işlevsel karşılıkları Çizelge 3.1'de karşılaştırmalı olarak sunulmuştur.

Çizelge 3.1 İşlev olarak BSS – YSA'nın yapısal karşılıkları

Biyolojik Sinir Sistemi	Yapay Sinir Ağı
Nöron	İşlemci Elemanı
Dentrit	Toplama Fonksiyonu
Hücre Gövdesi	Transfer Fonksiyonu
Aksonlar	Yapay Nöron Çıkışı
Sinapslar	Ağırlıklar

Aşağıdaki şekilde Şekil 3.6'da görüldüğü gibi biyolojik sinir sistemlerinin duyular gelen algılardan elde ettiği verilere benzer bir işlemin yapay sinir ağı modelinde de görüldüğü ve gelen her n tane verinin toplamda Xn ile ifade edildiği görülmektedir. Girdi şeklindeki her veri ağırlılarla çarpılır ve bu şekilde tüm girdi verileri toplanır. Toplanan girdi verilerine ön yargı (tahmini atamalar) tanımlanır ve bu şekilde elenerek net yargıya ulaşılır. Transfer fonksiyonunda net girdi işleme dahil edilerek bir veri çıktısı elde edilir.



Şekil 3.6 Yapay sinir hücresi (Anonim, 2018)

3.3.2 Tek Katmanlı Algılayıcılar

Çok katmanlı algılayıcıların temelini oluşturan tek katmanlı algılayıcılar, girdi-hesaplama-çıktı katmanları olmak üzere üç kısımdan oluşmaktadır. Giriş için tanımlanan değerler belirlenmiş ağırlıklar ile çarpılır ve işlem sonucu genel toplam üçüncü kısım olan çıktı katmanına iletilir.

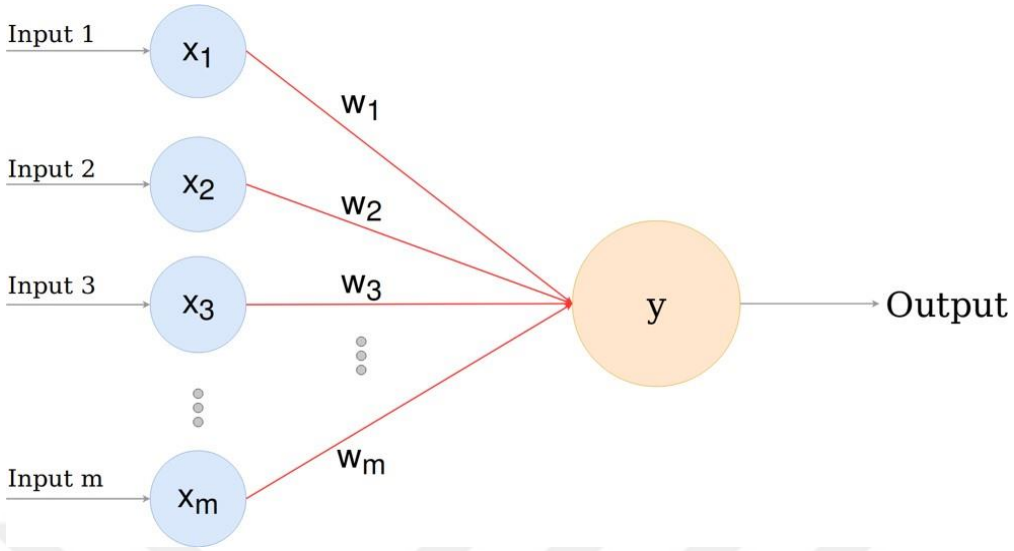
> *"Yapay sinir ağı denklemde x olarak gösterilen girdi öznitelikleri ile denklemde w ile gösterilen ağırlık değerlerini çarpma işlemine tabi tutar ve hesaplanan değer denklemde b ile ifade edilen yanlılık değeri ile toplanmaktadır". (Öğücü 2006)*

Çıkan sonuç aktivasyon fonksiyonundan geçirilerek işlem yapılır. Çıkan sonuç kapalı-açık devre şemalarında veya biyolojik sinir sistemlerindeki var-yok benzeri bir şekilde 1 veya 0 değeri elde edilir. Elde edilen bu 1-0 değerleri bu sinir ağı modelinin sınıflandırmada sağlıklı bir şekilde kullanılabileceğini belirtmektedir. Aşağıdaki eşitlikte (Eşitlik 3.1) bir yapay sinir ağı modelinin yaptığı matematiksel hesaplama fomülize edilmiştir.

$$f(x) = \begin{cases} 1, & w.x + b > 0 \; ise \\ 0, & değilse \end{cases} \tag{3.1}$$

Şekil 3.6'da görüldüğü gibi girdi sayısında herhangi bir kısıtlama bulunmamaktadır. Girdi vektörleri pozitif-negatif değeler alabilir. Bu girdi değerlerinin 1'den büyük veya 1'den küçük olmaları toplam performansı etkilememektedir. İşlemin üretmesini istediğimiz değerlere hedef değerler denilmektedir. Girdi vektörler için belirtilen kısıtlama bulunmaması, pozitif-negatif değerler alabilmesi, alınan değerlerin 1'den küçük veya büyük olması gibi durumlar hedef vektörler için geçerli olmamaktadır. Burada kullanılan aktivasyon fonksiyonu nedeniyle hedef vektörlerin içerebileceği değerler 1 ve 0 ile sınırlıdır (Öğücü, 2006).

Tek katmanlı sinir ağlarında (Şekil 3.7) giriş değerleri input1-input2-input3…input m eşitlik 3.2'de anlatılan formüle uygun olacak şekilde x1-x2-x3…xm şeklinde gösterilip ağa dahil olan her girdi değeri w1-w2-w3…wm olarak ifade edilen ağırlık değerleriyle işleme tabi tutularak bir çıktı değeri elde edilmektedir.

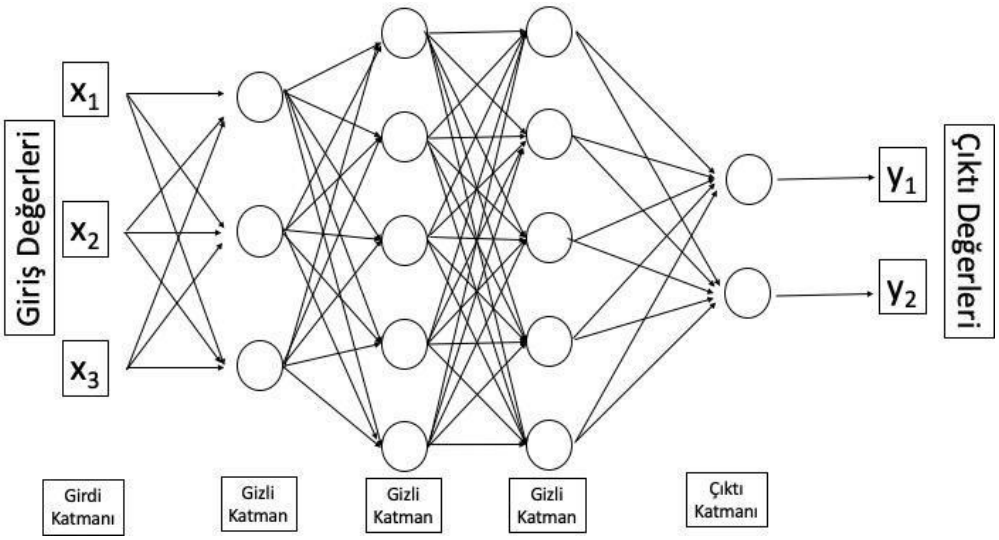


Şekil 3.7 Tek katmanlı yapay sinir ağı yapısı (Anonim, 2020)

3.3.3 Çok Katmanlı Algılayıcılar

Çok katmanlı algılayıcıların, tek katmanlı algılayıcıları temel aldığından daha önce bahsedilmişti. Derin ağlarda çok karmaşık problemlerin çözülmesi amacıyla tek katmanlı algılayıcılarda yapılan geliştirmeler sonucunda çok katmanlı algılayıcılar elde edilmiştir. Bu süreç, tek katmanlı algılayıcılarda bulunan girdi- çıktı katmanları arasına gizli katmanların eklenmesiyle elde edilmiştir.

Gizli katmandaki her bir nöron tek katmanlı algılayıcılarda da ifade edildiği gibi her bir input girdi değerinin, ağırlık değerleri ile çarpılarak elde edilen matematiksel değerin sonraki işlem olan aktivasyon fonksiyonuna tabi tutup sonraki katman içine tekrar girdi değeri olarak tanımlanmasını sağlamaktır. Bu tekrar edilen işlem çok katmanlı algılayıcıların her gizli katmanında tekrar edilmektedir. Aşağıdaki şekilde (Şekil 3.8) çok katmanlı bir yapay sinir ağı modeli şematize edilmiştir (Kabalcı, 2014).



Şekil 3.8 Çok katmanlı yapay sinir ağı yapısı (İslamoğlu, 2015)

Çok katmanlı ağlarda tek katmanlı birden fazla ağ gizli katmanlarda kullanılır. Bu şekilde yapay sinir ağlarında gerçekleştirilen işlemlerle daha fazla katman sayesinde daha derin ağların oluşturulması sağlanmaktadır. Girdi katmanından sağlanan her bilgi her veri gizli katmanlarda ayrı ayrı işleme tabi tutulmaktadır. Buradan elde edilen her veri sonraki katmanda yine tüm nöronlar tarafından işleme tabi tutulmaktadır. Bu işlem gizli katmanda bulunan katman sayısı kadar tekrar ederek ilerlemektedir. Girdi değerlerine uygun olan sonuçlar çıktı katmanında üretilmektedirler.

3.3.4 İleri Yönde Yayılım (Forward Propagation)

Yapay sinir ağlarında ileri yönde yayılım (Forward Pagation) en doğru sonuçla karşılaştırılabilecek tahminler elde edinceye kadar ağ mimarisinde soldan sağa doğru giriş verilerinin kullanılarak hesaplama işlemlerinin yapılması şeklinde tanımlanmaktadır. Bu işlem biyolojik sinir sitemiyle karşılaştırıldığında her nöronun girdi değerlerini, kendisiyle karşılaştırılabileceği ağırlık ve bias değerleriyle değerlendirilip buna göre bir tahmin üretmesi anlamına gelmektedir.

Yapay sinir ağlarında karşılaştırma işlemlerinde kullanılan bias değerleri istenmeyen sıfır değerlerinin oluşmaması için her bir nöronun aktivasyon fonksiyonlarını değiştirerek sıfır değerini almamasına yardımcı olan sayısal değerlerdir (Jibril vd., 2022).

Bilgisayar biliminde ekrandaki her sembolün veya resmin sayısal karşılıkları bulunmaktadır. Bilgisayarlı görü işlemlerinde fotoğraf tanımlayabilmek için gri veya renkli her girdi fotoğrafının piksel değerinin matematiksel karşılığıyla ifade edilmesi gerekmektedir. Ortaya çıkan öznitelikler random veya başka bir derin öğrenme modelinden aktarılan ağırlık değerleri olsun fark etmez, elde edilen bu ağırlık değerleri hesaplamaya tabi tutulmaktadır.

İleri yönde yayılım adından da anlaşılacağı gibi her işlem sonunda elde edilen değer ileri yönde bir sonraki katmana aktarılmaktadır. Elde edilen değerlerin bir anlam ifade edebilmesi ve doğrusal olmaması için aktivasyon fonksiyonlarıyla (çok katmanlı yapılarda bahsedildiği gibi) yeniden işleme tabi tutulması gerekmektedir. Yapılan hesaplamalar sonucunda şayet negatif piksel değerleri oluşursa sistem bu değerleri önemsiz kabul edip sıfıra eşitler. İşlemler bu şekilde devam edip her işlem ileri yönde bir sonraki katmana aktarılarak ileri yönde yayılım elde edilmektedir (Gurney, 1997).

İleri yönde yayılım sürecinde aktivasyon fonksiyonlarının görevi modelin doğrusallığını kırmaktır. Gerçek hayatta sınıflandırmalar doğrusal olmadığı için, aktivasyon fonksiyonları yapay sinir ağlarının gerçeğe yaklaşmasına yardımcı olmaktadırlar. Bu yönüyle aktivasyon fonksiyonlarının işleme dahil edilmesi oldukça önemlidir.

Her bir nöronun hesaplama işlemi eşitlik (3.2) ve eşitlik (3.3)'te ifade edilmiştir.

$$Z = X * W + b \tag{3.2}$$

$$Y = \sigma(Z) \tag{3.3}$$

Denklemde X giriş verilerini, W ağırlıkları, b bias değerini ve σ fonksiyonu ise aktivasyon fonksiyonunu ifade eder (Jibril vd., 2022).

3.3.5 Maliyet Fonksiyonunun Hesaplanması

Yapay sinir ağlarından çok katmanlı olanlarda girdiye ait özellikler ileri yayılımın ilk basamağında modele ait kurallara göre veya rastgele seçilerek ağırlık değerleriyle hesaplanıp matematiksel sonuçlar üretmektedir. Her girdi işlem gördüğü

katmandan bir sonraki katmana tekrar girdi olarak aktarılarak son katmana ulaşıncaya kadar defalarca işleme tabi tutulmaktadır. Bu şekilde ilerleyen girdiler son katmanda çıktı olarak üretilmektedir. Sonuç kısmında üretilen bu tahmin değerlerinin gerçek değerlere olan mesafesinin ve doğru tahmin etme yüzdesinin ölçülmesi gerekmektedir. Bu noktada kullanılan modele gerçeğe en yakın tahmini yaptıracak parametre değerlerinin bulunması amaçlanmaktadır. Maliyet fonksiyonu gerçek değerler ile yapay sinir ağlarındaki tahmin değerlerini nicel olarak hesaplayarak sayısal çıktılar olarak ifade etmektedirler.

Aşağıdaki eşitlik (Eşitlik 3.4) baz alındığında sinir ağındaki gerçek değer y ile, sinir ağı modelinin tahmin değeri $\hat{y}$ ile ifade edilmektedir.

$$L = -y.\log \hat{y} \tag{3.4}$$

Optimizasyon problemlerinde kayıp fonksiyonu daha düşük seviyededir ve tek bir örneğin veya bileşenin hata değerini belirlemede kullanılır, maliyet fonksiyonu ise daha yüksek bir seviyededir ve bir sistemin nasıl değerlendirildiğini açıklar. N boyutlu bir veri kümesi için çok sınıflı bir sınıflandırma kaybı maliyeti aşağıdaki denklem ile hesaplanır.

Çok katmanlı sınıflandırıcı modellerinde genellikle eşitlik 3.4 üzerinden maliyet fonksiyonları ortalaması hesaplanmaktadır. Maliyet fonksiyonlarındaki hata değerleri optimizasyon problemlerine göre daha yüksek seviyelerdedir. N boyutlu bir veri kümesinde çok katmanlı bir sınıflandırma da kayıp (loss) maliyeti eşitlik 3.5 ile gösterilmiştir (Jibril vd., 2022).

$$J = -1\ N \sum yi\ .\ \log(\hat{y}i)\ N\ i{=}1 \tag{3.5}$$

Bir ağda iterasyonların yani eğitimlerin ne zaman biteceğine karar veren yapılar çapraz doğrulama ve maliyet fonksiyonlarıdır (Golik vd., 2013).

Eşitlik 3.6'da ileri yayılım için n katmanlı bir yapay sinir ağında n. katmanda yapılacak hesaplama formülü gösterilmiştir. Bu hesaplama sonucunda elde edilen değer ŷ olarak ifade edilen tahmin değeridir. Bu tahmin değerinin gerçek değere olan uzaklığı maliyet ya da kayıp adı verilen fonksiyon ile hesaplanmaktadır.

Aşağıda gösterilen eşitlik 3.6'da n katmandan oluşan bir sinir ağı modelinin ileri yayılım için n katmanlı hesaplama formülü belirtilmiştir. Yapılan hesaplamalar sonucunda üretilen değerler ŷ sembolü ile ifade edilen tahmin sonuçlarıdır. Elde edilen tahmin değerinin gerçek değere olan uzaklığı ise daha öncede belirtildiği gibi kayıp (loss) fonksiyonu ile hesaplanabilmektedir.

$$Z[n]=W[n]*A[n-1]+b[n] \tag{3.6}$$

Eşitlik 3.7 ortalama mutlak hata formülünü göstermektedir. Formülde m ifadesi tahmin edilen örnek sayısını göstermektedir.

Maliyet fonksiyonu hesaplamalarında kullanılan iki yöntem bulunmaktadır, bunların ilki ortalama mutlak hata diğeri ise ortalama kare hatasıdır. Ortalama mutlak hata hesaplanırken tahmin gurubundaki hataların ortalama büyüklükleri yönler dikkate alınmadan ölçülmektedir. Aşağıdaki eşitlik (Eşitlik 3.7) ortalama mutlak hata formülize edilmiştir. Ortalama mutlak hata formülündeki *n* değeri tahmin edilen örnek sayısını ifade etmektedir.

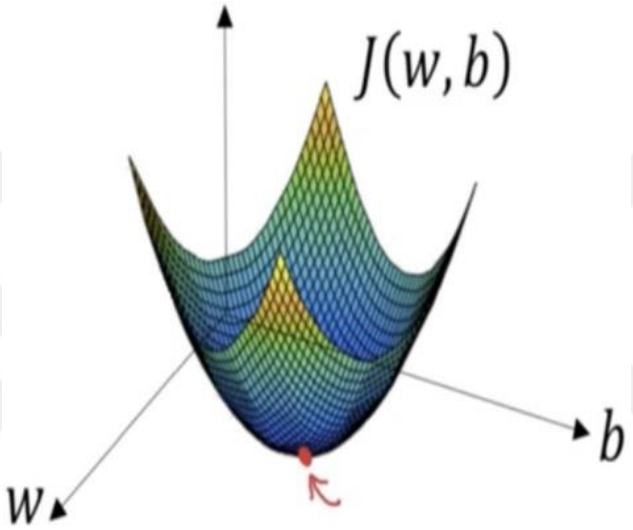
$$\text{Ortalama Mutlak Hata} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \tag{3.7}$$

Bir diğer maliyet fonksiyonu hesaplama yöntemi ortalama kare hatasında ise; tahmin edilen ve beklenen sonuçlar arasındaki ortalama kare hataları hesaplanır. Bu hata türünde farklılıkların mutlak değerinin alınması yerine kareleri alınmaktadır. Eşitlik 3.8 ortalama kare hatası formülünü göstermektedir.

Bir diğer maliyet hesaplama fonksiyonu ise ortalama kare hatası yöntemidir. Ortalama kare hatası yönteminde tahmin edilen ve beklenen çıktılar arasındaki hataların ortalama karesi hesaplanmaktadır. Bu denklemde ortaya çıkan farklılıkların mutlak değerleri değil kareleri alınmaktadır. Aşağıda eşitlik 3.8'de ortalama kare hatası fomülü gösterilmiştir.

$$\text{Ortalama Kare Hatası} = \frac{1}{2m}\sum_{i=1}^{m}(\hat{y}^{(i)} - y^{(i)})^2 \tag{3.8}$$

Aşağıda gösterilen dereceli alçalma fonksiyonu gösteriminde (Şekil 3.9) şeklin dış bükey olarak gösterilmesi maliyet fonksiyonuna ait ileri yönde yayılımın ilk basamaklarında grafiğin tepe noktası olarak tanımlanmaktadır. Burada amaç maliyet fonksiyonunun grafikte tabana doğru inmesini sağlamaya yönelik ağırlık ve yanlılık değerlerinin bulunmasını sağlamaktır. Grafikte okla gösterilen taban noktası ise maliyet fonksiyonunu en düşük değerlere ulaştıran ağırlık ve yanlılık değerleri olarak ifade edilmektedir.



Şekil 3.9 Dereceli alçalma fonksiyonu (Murat, 2021)

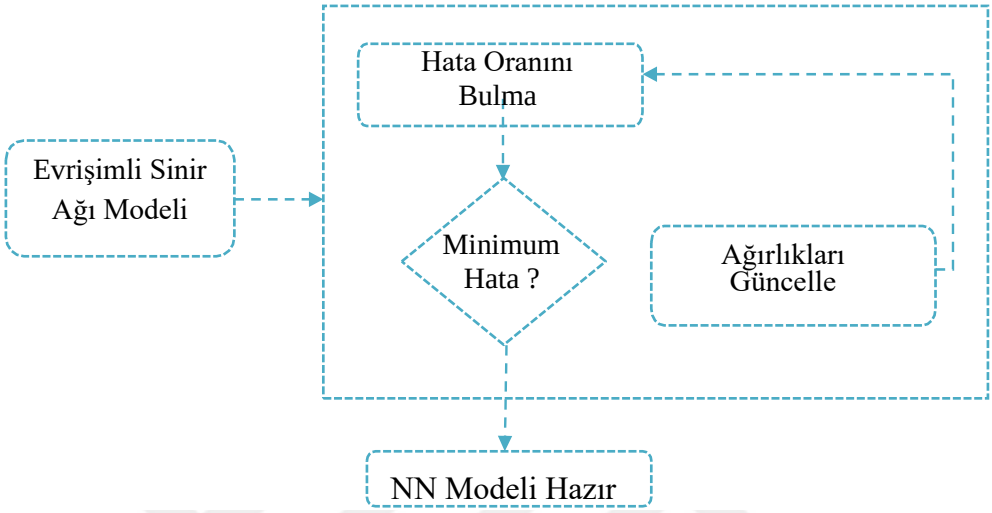
Eğitim sürecinde ağın hata değeri tek bir iterasyon ile hesaplanabilmektedir. Bulunan bu hata değeri hesaplanan maliyet fonksiyonu ile birlikte ağdaki ağırlıkların güncellenebilmesi için ağa geri gönderilmektedir. Sonraki her iterasyonda yapılan bu işlem tekrar edilmektedir, yani hata miktarının düşülmesi için her işlemde ağrılıklar güncellenmektedir.

> *"Kayıp fonksiyonun eğimi kullanılarak ağırlıkların güncellenmesi geri yayılım olarak ifade edilmektedir" (Kızılboğa, 2021).*

3.3.6 Geri Yönde Yayılım ve Hesaplama Aşamaları

Geri yönde ya da geri beslemeli yayılım tüm yapay sinir ağlarının bünyesinde bulunmaktadır. Verilerin eğitilmesi sürecinde tüm yapay sinir ağlarının ileride

uygulama kısmında da görüleceği gibi ince ayarlarının yapılarak hata oranlarının iyileştirilmesi gerekmektedir. Ağılıklar doğru ayarlandığında model daha güvenilir olacak ve daha düşük hata oranları üretecektir (Al-Masri, 2015).



Şekil 3.10 Geri yayılımın çalışma mekanizmasının akış diyagramı

Yukarıdaki Şekil 3.10'da geri yayılma süreci mekanizmasının günlük iş akışlarını gösterilmektedir. Geriye doğru yayılma sırasında, bu hesaplama faaliyetleri aşağıda belirtildiği gibi gerçekleşecektir.

• Hata Oranını Bul (Find error rate): Burada model çıktısını gerçek çıktıyla hesaplamamız gerekiyor.

• Minimum Hata (Error is min): Hatanın minimize edilip edilmediğini çapraz doğrulama.

• Ağırlıkları Güncelleyin (Update the Weights): Hata, kabul edilebilir aralıktan fazladır, ağırlıkları ve sapmaları güncelleyin. Bundan sonra, hatayı tekrar kontrol edin. Hata azalana kadar işlemi tekrarlayın.

• Sinir Ağı Modeli Hazır (NN Model is Ready): Hata oranı kabul edilebilir aralığa geldiğinde, model verileri tahmin etmek için kullanıma hazırdır (Guo vd., 2020).

Geri yönde yayılım algoritmaları, gözlenen değer ile üretilen sonuç değeri arasındaki farka bakılarak ağırlıkların güncellenmesi mantığıyla çalışmaktadır. Maliyet fonksiyonu hesaplanırken Şekil 3.9' da bahsedildiği gibi bu algoritma dereceli alçalma (Gradient Descent) tabanlı bir optimizasyon algoritmasıdır.

Bu algoritmanın dezavantajları aşağıda verilmiştir.

1. Alçalma tabanlı olması sebebiyle yerel minimum değerini geçemeyebilir ve bu sebeple en doğru sonuca ulaşması mümkün olmayabilir.

2. Alınan başlangıç değerleri ve algoritma parametreleri eğitim performansını istenmeyen yönde etkileyebilir.

> *"Bu zayıflıkları düzeltmek için literatürde çeşitli çalışmalar yapılmıştır. Özellikle yerel (local) tuzaklardan kaçınmak için sezgisel algoritmalardan (heuristic algorithms) yararlanan çalışmalar da bulunmaktadır" (Aladağ, 2009).*

3.3.7 Hiperparametreler

Yapay sinir ağlarının çok katmanlı mimari tasarımında regresyon ve sınıflandırma için gradyan artırma, rastgele orman ve sinir ağları gibi makine öğrenimi (ML) algoritmaları, bir dizi hiper parametre içermektedir. Bu parametrelerin çalıştırmadan önce ayarlanması gerekmektedir. Doğrudan, birinci düzey model parametrelerinin aksine, bunlar eğitim sırasında belirlenen bu ikinci seviye ayar parametrelerinin genellikle dikkatli bir şekilde ayarlanması gerekmektedir. Bu parametreler maksimum performans elde etmek için optimize edilmiştir (Eiben ve Smit, 2011).

Verimli bir ayarlama stratejisi seçmenin yanı sıra, ayarlanabilir hiperparametreler kümesi ve bunlara karşılık gelen aralıklar, ölçekler ve müteakip örnekleme için potansiyel önceki dağılımlar kullanıcı tarafından belirlenmelidir. Bazı hiperparametreler güvenle şu şekilde ayarlanabilmektedir, birçok farklı senaryoda iyi çalışırlarsa varsayılan değerler. Çünkü bu konularda yanlış alınan kararlar, ortaya çıkan modelin kalitesini veya en azından verimliliği engelleyebilmektedir.

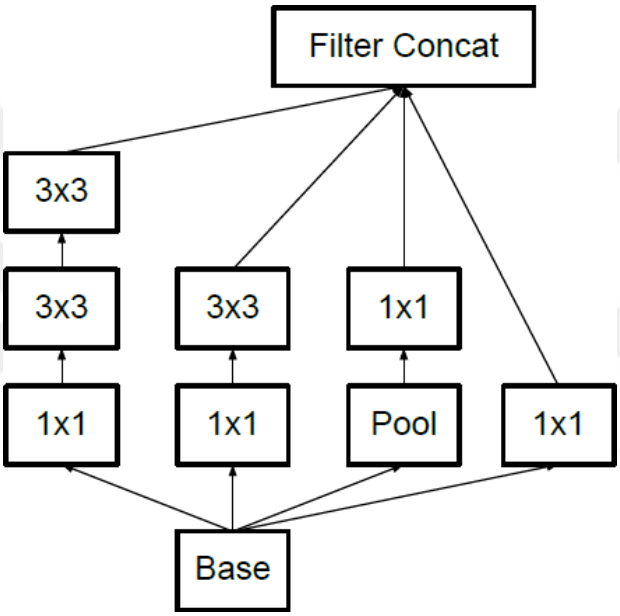
Bu bölümde tez çalışmalarında kullanılan evrişimsel sinir ağı modellerine ait hiperparametrelerden bahsedilmiştir.

3.3.8 Filtre Boyutu

Evrişimli katmanlarda 3×3, 5×5 ve 7×7 olmak üzere üç tip filtre boyutu kullanılmaktadır. Filtre boyutlarının büyük olması hesaplama açısından maliyetlidir

çünkü daha büyük filtre boyutlarının küçük boyutlarına göre çok sayıda işlem yapmak zorunda kalmaktadır. 3×3 neredeyse üç katıdır. Hesaplama açısından 5×5 filtre boyutlarından daha maliyetlidir. 5×5 filtreler hesaplama açısından daha ucuzdur, ancak daha fazla alan kapladığı için daha çok doldurulma şansı bulunmaktadır (Khanday ve Dadvandipour, 2020).

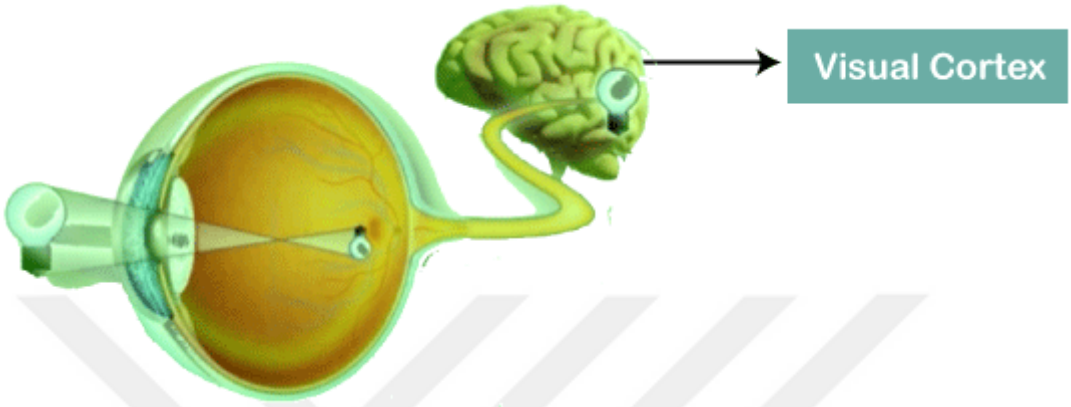
5×5 boyutlu bir filtre bir seferde geniş bir alanı kapsadığı için, hesaplama açısından daha ucuz hale gelmektedir. Şekil 3.11'de 1×1 ve 3×3 değişken boyutlarında filtrelerin kullanılma şekli gösterilmiştir.



Şekil 3.11 1x1 ve 3x3 filtre boyutlarını kullanan Başlangıç Modülleri

3.3.9 Dolgulama ve Adım Kaydırma

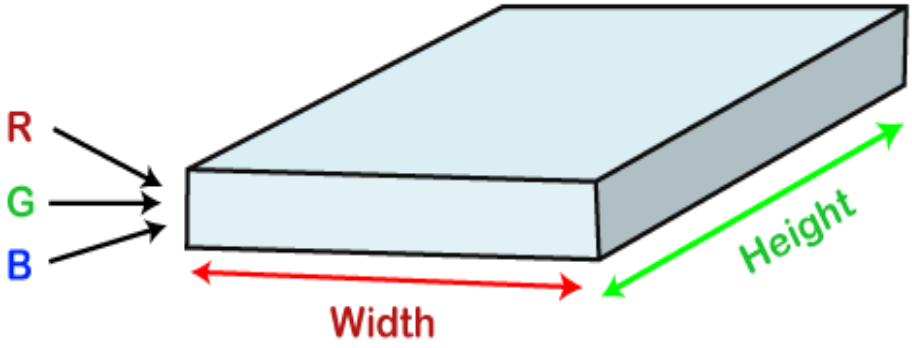
Konvolüsyonel Sinir Ağlarının nöronları arasındaki bağlantı modelini örnek aldığından daha önce bahsedilmişti. Bu kısımda görsel korteksten ilham alan özel bir ileri beslemeli yapay sinir ağı türünün çalışma mantığı anlatılmıştır.



Şekil 3.12 Biyolojik göze gelen bir görüntünün işlem birimine (beyne) aktarılması

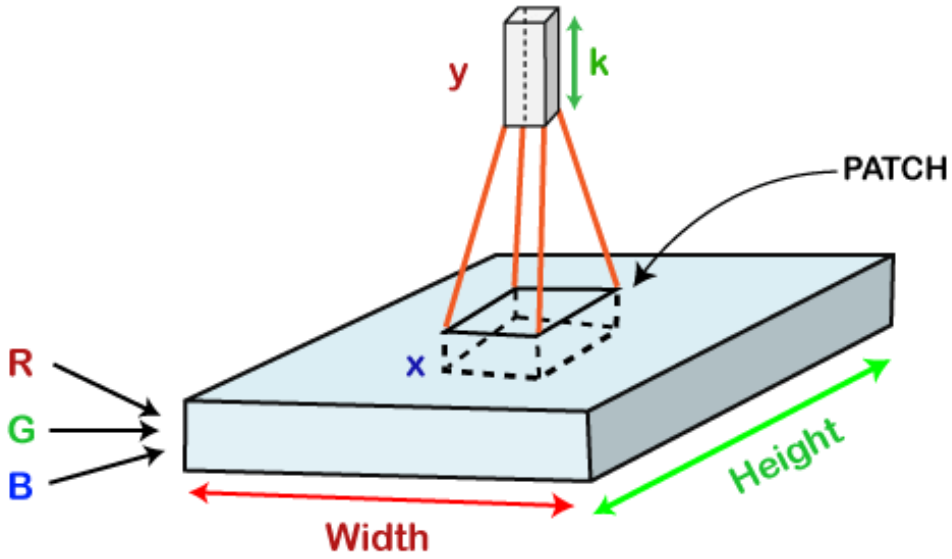
Görsel korteks, görsel alanlara bölgeye duyarlı küçük bir hücre bölgesini kapsamaktadır (Şekil 3.12). Bazı belirli yönlendirme kenarlarının mevcut olması durumunda, beynin içinde yalnızca bazı bireysel nöronal hücreler ateşlenmektedir, örneğin bazı nöronlar dikey kenarlara maruz kaldıklarında tepki vermektedir, ancak bazıları yatay veya çapraz kenarlara gösterildiğinde tepki vermektedir; Konvolüsyonel Sinir Ağlarının arkasındaki çalışma mantığı yani ilham alınan motivasyon da bundan başka bir şey olmamaktadır.

Covnet olarak da adlandırılan Konvolüsyonel Sinir Ağları, parametrelerini paylaşan sinir ağlarından başka bir şeyi kapsamamaktadır. Uzunluğu, genişliği ve yüksekliği kapsayan bir küboid olarak somutlaşan bir görüntü olduğunu varsayıldığında görüntünün boyutları, aşağıda verilen görüntüde gösterildiği gibi (Şekil 3.13) Kırmızı, Yeşil ve Mavi kanallarla temsil edilmektedirler.



Şekil 3.13 R G B renk uzayının uzamsal yapısı

Aynı görüntünün küçük bir parçasını aldığımızı ve ardından dikey olarak temsil edilen k adet çıktıya sahip küçük bir sinir ağı çalıştırdığımız varsayıldığında küçük sinir ağımızı görüntünün her yerine kaydırdığımızda yani adım kaydırma işlemi yaptığımızda, farklı genişlik, yükseklik ve derinlik oluşturan başka bir görüntü ile sonuçlanacaktır (Şekil 3.14). R, G, B kanallarına sahip olmak yerine, aslında Konvolüsyon kavramı olan daha az genişlik ve yüksekliğe sahip daha fazla kanalla karşılaştığımızı fark edeceğiz. Bu durumda, görüntününki ile benzer yama boyutuna sahip olmayı başarırsak, o zaman bu normal bir sinir ağı olurdu. Bu küçük yama nedeniyle bazı ağırlıklarımız bulunmaktadır.



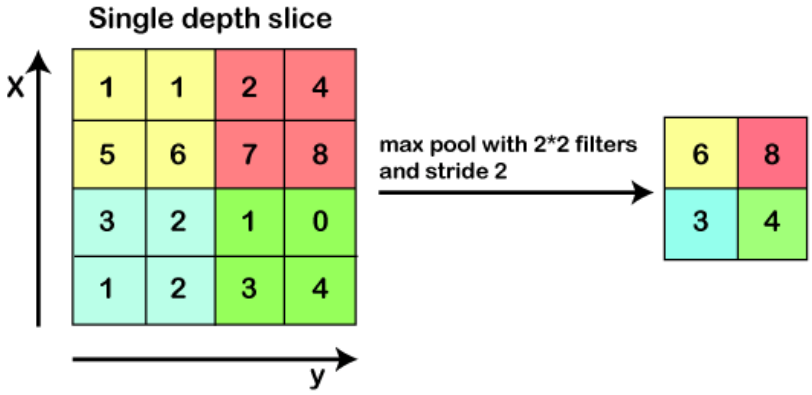
Şekil 3.14 Üç boyutlu renk uzayında koordinat yolu gösterimi

Matematiksel olarak şu şekilde anlaşılabilir; Evrişimli katmanlar, her filtrenin sağlanan giriş hacmininki kadar küçük genişlik, yükseklik ve derinliği kucaklayacağı şekilde bir dizi öğrenilebilir filtre içerir (görüntü giriş katmanıysa muhtemelen 3 olacaktır). 34x34x3 boyutundan oluşan görüntü üzerinde filtre boyutu axax3 olacak şekilde evrişimi çalıştırmak istediğimizi varsayalım. Burada a, yukarıdaki 3, 5, 7 vb. 'den herhangi biri olabilir. Görüntünün boyutuna göre küçük olmalıdır. Her filtre, ileri geçiş sırasında giriş hacminin her yerinde kayar. Adım adım kayar, her bir adımı daha yüksek boyutlu görüntüler için 2 veya 3 veya 4 değerini kapsayan bir adım olarak çağırır, ardından filtre ağırlıkları ile giriş hacminden yama arasında bir nokta çarpımı hesaplanmaktadır.

Filtre sayısına benzer bir derinlik değerine sahip bir çıkış hacmi elde etmek için filtrelerimizi kaydırdıkça ve kaydırdığımızda her filtre için 2 Boyutlu çıktı ile sonuçlanacaktır. Ardından, ağ tüm filtreleri öğrenecektir (Anonim, 2023c).

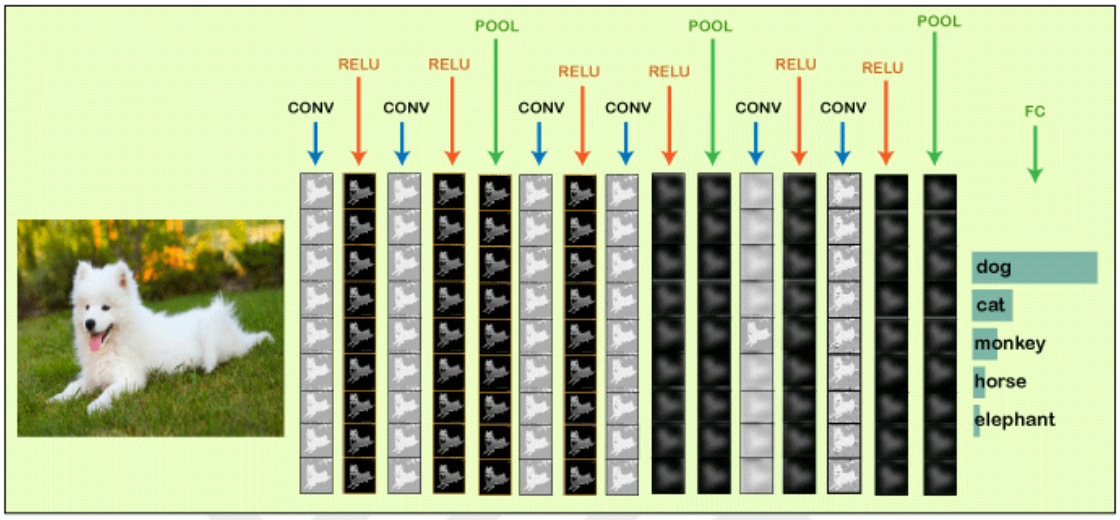
3.3.10 Havuzlama Katmanı Uygulama

Bu katman, [16x16x12] hacimle sonuçlanan uzamsal boyutlar (genişlik, yükseklik) boyunca bir alt örnekleme işlemi gerçekleştirmek için kullanılmaktadır (Şekil 3.15).



Şekil 3.15 Maksimum havuzlama filtresinin tek dilimlerden oluşma süreci

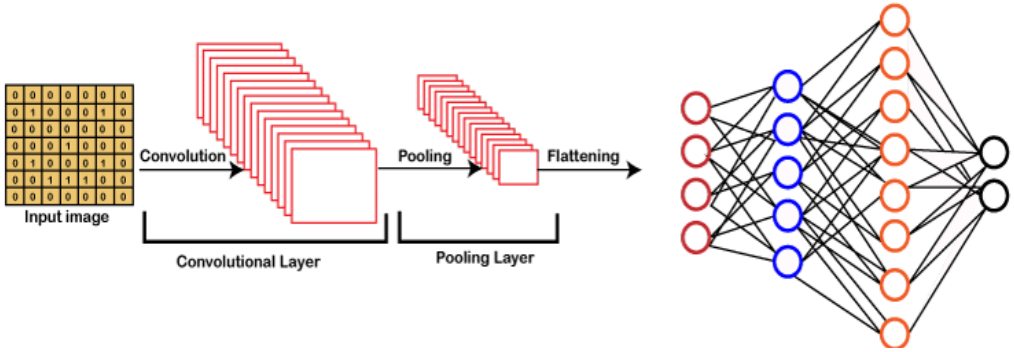
Bir önceki katmandan bir girdi alan ve ardından sınıf puanlarını hesaplayan ve sınıf sayısına eşit boyuta sahip 1 Boyutlu bir dizi ile sonuçlanan normal bir sinir ağı katmanı olarak tanımlanabilmektedir.



Şekil 3.16 Girdi, evrişim, ReLU ve havuzlama katmanlarında görüntü tanımlama

Bir evrişim (Convolution) katmanından oluşan özellik haritalarını oluşturmak için filtreler olarak da adlandırılan çoklu özellik dedektörlerini uygulayacağımız bir giriş görüntüsü ile başlanmaktadır. Ardından, bu katmanın üstünde, görüntülerimizdeki herhangi bir doğrusallığı ortadan kaldırmak veya doğrusal olmayanları artırmak için ReLU veya düzeltilmiş doğrusal birimi uygulanmaktadır (Şekil 3.16).

Ardından, Evrişimli katmana bir havuzlama katmanı uygulanmaktadır, böylece her özellik haritasından bir havuzlanmış özellik haritası oluşturulmaktadır, çünkü havuzlama katmanının ana amacı, görüntülerde mekansal değişmezliğe sahip olduğumuzdan emin olmaktır. Ayrıca, resimlerin boyutunu küçültmenin yanı sıra verilerimizin herhangi bir şekilde aşırı yüklenmesini önlemeye yardımcı olmaktadır. Bundan sonra, havuzlanmış tüm görüntüleri tek bir uzun vektöre veya tüm bu değerlerin sütununa düzleştirilecek ve ardından bu değerler yapay sinir ağında girilecektir. Son olarak, nihai çıktıyı elde etmek için görüntü yerel olarak bağlı katmana beslenecektir (Şekil 3.17).



Şekil 3.17 Giriş görüntülerinin evrişimli sinir ağında uygulanması

3.3.10.1 Öğrenme Hızı

Öğrenme hızı derin öğrenme modellerine ait her geri yayılımında modelin ağırlık ve yanlılık değerlerinin güncellenmesi hızı olarak tanımlanmaktadır.

GYYSA (Geri yayılımlı yapay sinir ağı)'nın eğitim sürecinin başlangıcında, nöronlar arasındaki ilişkiyi sağlayan ağırlık katsayılarına ilk olarak, rastgele küçük değerler atanır. Atanan başlangıç ağırlık değerlerinin büyük olması durumunda, eğitim işleminin başında, aktivasyon fonksiyonu doygunluğa ulaşabileceğinden, geri yayılım algoritması ilk yerel minimum (local minimum) noktasında tıkanır. Başlangıç ağırlık değerleri, GYYSA'nın öğrenme hızını ve istenilen çıktılara yakınsamasını etkilemektedir (Kavzoğlu ve Saka; 2005; Canan, 2006).

Hiperpamatreler içerisinde öğrenme hızı kullanılan ağırlıklara ne kadarlık kayıp eğimi uygulanacağını belirlemektedir. Ağırlıkların sürekli güncellenen değerlerindeki hızlı değişim öğrenme hızının yüksek olmasına bağlıdır. Buna göre öğrenme hızının düşük olması da en uygun eğim noktasına varış süresini uzatmaktadır. Derin öğrenme sinir ağı modellerinde optimum öğrenme hızı 0.1-0.001 aralığı olarak uygulanmaktadır. (Anonim, 2023d)

3.3.10.2 Aktivasyon Fonksiyonları

Yapılan işlemler sonucu üretilen değerlerin uç noktalara sapmaması ve beli aralıklarda kalması için aktivasyon fonksiyonları kullanılmaktadır. Yapay sinir ağlarında optimum değerlere ulaşılması için bu işlem hayati önem taşımaktadır. İşlem

başlangıcında random olarak belirlenen ağırlık değerleri girdi değerleri ile çarpılmaktadır. İleri ve geri yayılım işlemlerinde en uygun nöronun bulunması için ağırlık değerlerinin sürekli güncellenmesi gerektiğinden bahsedilmişti, ancak yapılan hesaplamaların bir anlam ifade etmesi için çok büyük veya çok küçük değerler üretmemesi yani belirli bir aralıkta kalmasının sağlanması önemlidir, bu noktada aktivasyon fonksiyonları kullanılmaktadır (Anonim, 2023e).

Çalışmanın bu bölümünde evrşimsel sinir ağı modellerine ait aktivasyon fonksiyonlarından aktif kullanılan birkaç tanesine denklemleri ve aldıkları aralık değerleri ile birlikte değinilmiştir (Şekil 3.18).

Foksiyon Adı	Fonksiyon Denklemi	Değer Aralığı
Sigmoid	$f(x) = \sigma(x) = \frac{1}{1 + e^{-x}}$	(0,1)
ReLU	$f(x) = \begin{cases} 0, & x < 0 \\ x, & x \geq 0 \end{cases}$	[0,∞)
Leaky ReLU	$f(x) = \begin{cases} 0.01, & x < 0 \\ x, & x \geq 0 \end{cases}$	(-∞, ∞)
Swish	$f(x) = 2x\sigma(\beta x) = \begin{cases} \beta = 0, & f(x) = x, \\ \beta \to \infty, & f(x) = 2max(0,x) \end{cases}$	(-∞, ∞)

Şekil 3.18 Bazı Aktivasyon fonksiyonları (Chetoui, 2021)

Yaygın olarak kullanılan aktivasyon fonksiyonları aşağıda gösterilmiştir.

Sigmoid Fonksiyonu: Yapay sinir ağlarında en çaygın kullanılan aktivasyon fonksiyonlarının ilkidir. Fonksiyon işlemleri [0,1] aralığında çıktı üretir.

***Tanh Fonksiyonu*:** [-1,1] aralığında çıktı üreten İleri-geri yayılım durumlarında kullanılan ve doğrusal olamayan bir fonksiyondur.

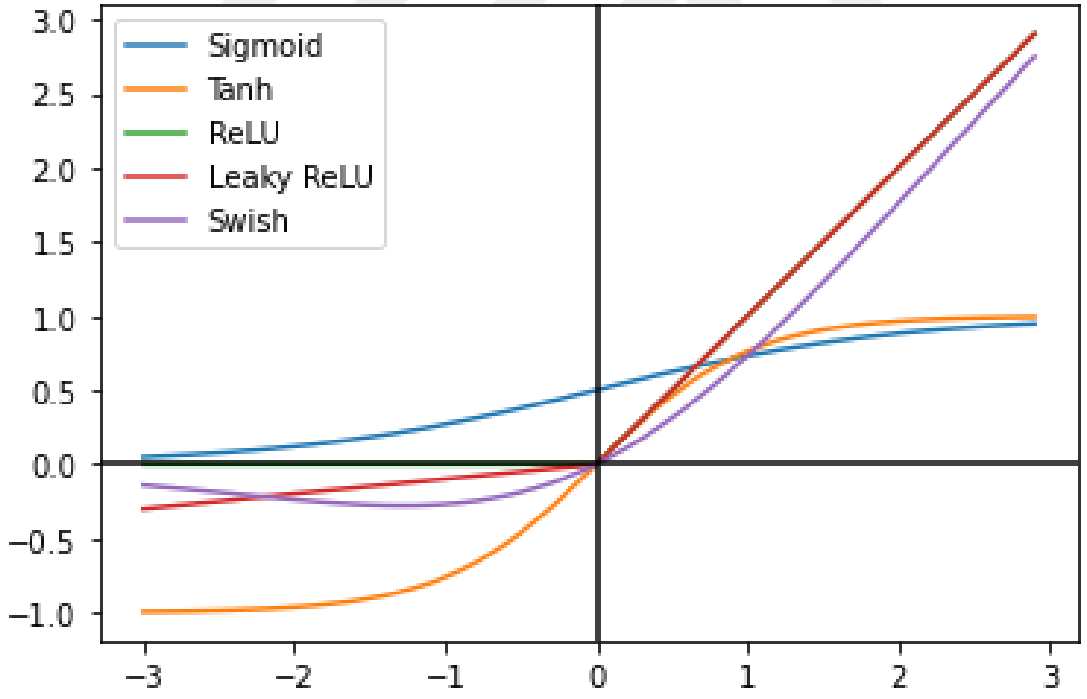
ReLU Fonksiyonu: Bu, etkinleştirme işlevini her öğeye uygulayacaktır. Uyguladıktan sonra işlevi, birim boyutu değişmeyecektir, yani 28×28×16.

Swish Fonksiyonu: Google araştırmacıları tarafından yeni keşfedilen bu fonksiyon girdiler ile sigmoid fonksiyonunun çarpımını çıktı olarak üretmektedir.

Pool (HAVUZ) Fonksiyonu: Bu katman alt örnekleme işlemi için kullanılır. Uzay boyunca uygulanır boyutlar, yani genişlik ve yükseklik boyunca. Alt örnekleme için adım adım kullanılır böylece ortaya çıkan hacim 14×14×16 olmaktadır.

FC: Tamamen bağlantılı katman, geçerli sınıf puanlarını hesaplamaktadır. Bu katmandaki her nöron bir önceki katmandaki nöronlara bağlı olması gerekir. Çıktının boyutu 1×1×10, burada on değeri, kullanılan sınıf sayısına karşılık gelir. Rakam tanıma 0-'dan oluştuğu için 9; bu nedenle, her kategori için bir tane olmak üzere on sınıf kullanılır.

Şekil 3.19'da ise aktivasyon fonksiyonları ve koordinat düzlemi üzerinde aldıkları değer aralıkları gösterilmiştir.



Şekil 3.19 Aktivasyon fonksiyonları ve değer aralıkları (Anonim, 2023f)

3.3.10.3 Optimizasyon ve En İyileme

Optimizasyon, makine öğreniminin kalbinde yer almaktadır. Çoğu makine öğrenimi sırasında optimizasyon problemlerine makine öğrenimi analistini bir sorunu çözmek için çalışırken modelleyici, uygun bir model ailesi seçerek sorunu formüle etmektedir ve mevcut verileri modellemeye uygun bir formata dönüştürmektedir. Optimizasyon problemleri bir yandan, matematiksel programlama teorisi, optimal çözümü oluşturan şey – optimallik koşulları, öte yandan matematiksel programlama algoritmaları, makine öğrenimi araştırmacılarını geniş verileri eğitmek için farklı araçlarla donatmaktadır. Yapay sinir ağlarında çok katmanlı yapılarda amaç eğitimde kullanılan modelin optimum bir genelleme üretmesidir. Daha sonra oluşan bu genelleme veri setinin eğitim ve test verilerine uygulanarak performansının ölçülmesi müknü olmaktadır (Bradley ve Mangasarian, 1998).

Derin öğrenme mimarisinde çok katmanlı yapay sinir ağının amacı eğitimde kullanılan modelin iyi bir genelleme elde etmesidir. Daha sonra bu genellemeyi doğrulama ve test verilerine uygulayarak performansını ölçmek mümkün olmaktadır.

> *"Eğitim sırasında oluşabilecek hataları minimum seviyeye indirebilmek için optimizasyon ve en iyileme yöntemleri olarak adlandırılan eğimli iniş (gradient descent) yöntemleri kullanılmaktadır. Optimizasyon süreci öğrenme hızının da dahil olduğu adım adım işleyen bir süreçtir" (Seyyarer vd., 2020).*

3.3.10.4 Devir sayısı ve Yığın Boyutu

Mevcut bir veri kümesine ait örneklerin farklı yöntemlerle sıralanmaları için örneklere ait zorluklara ulaşılması gerekmektedir. Bu çalışmada sıralamadan ziyade gruplama tekniği kullanılarak, veri kümesine ait her bir grubun eğitim süreleri, epok sayıları, doğruluk ve kesinlik değerleri temin edilmiştir. Veri kümelerine ait sıralamalarda ise kullanılan modele ve eğitim ortamına uygun olan yığın boyutuna (batch size) göre sınıf bazlı olarak sıralanmış mevcut verilerden alınan örneklerle oluşturulmaktadır. İniş yönlü eğim gösteren algoritmalarda eğitim örnekleri yerine

gruplar üzerinde çalışılarak parametreler daha hızlı bir şekilde güncellenebilmektedir. (Brownlee, 2021).

Modellerin başarı oranlarının artması için belirli sayılarda iterasyonlar ile eğitilmesi gerekmektedir. Bu çalışmada YOLOv5 ile 100 iterasyonluk, FasterCNN ile 1500 iterasyonluk, SSdMobilNet ile 20 iterasyonluk eğitim süreçleri sonucunda elde edilen başarı oranları kullanılmıştır. Eğitim süreçlerinde FasterCNN iterasyon (epok) sayısı yüksek olmasına karşın daha kısa sürede sonuç üretmiş ve ortalama doğruluk değeri %93 ile daha yüksek bulunmuştur. Bu sonuçlara bakıldığında yığın boyutu (Batch Size), devir sayısı ve adım sayısı gibi parametrelerin sonuçlar üzerinde ayrı ayrı etkileri görülmektedir.

3.3.11 Performans Değerlendirme Kriterleri

Bu bölümde modellerde kullanılan kesinlik, duyarlılık, F score, doğruluk, loss (kayıp veri) performans değerlendirme kriterlerinden bahsedilmiştir.

Çilek meyvesine ait gelişim evrelerinin tespit edilebilmesi için önerilen metodun performans değerlendirmesi aşağıdaki denklemlerde verilmiştir. Kullanılan bu parametre değerleri veri setinden çıkarılan öznitelikler üzerindeki davranışını tahmin etmemize olanak sağlamaktadır (Siuly vd., 2020).

Burada,

Kesinlik (Precision): Keskinlik değerine ait verileri göstermektedir. Veri seti içerisinde doğru olarak tahmin edilen çilek meyvesinin toplam çilek meyvesi sayısına oranıdır.

$$\text{Kesinlik } (Precision) = \frac{TP}{(TP+FP)} \tag{3.9}$$

Duyarlılık (Recall): Veri seti içerisindeki çilek meyvesine ait duyarlılık oranını göstermektedir. Veri setinde toplam istenilen çilek meyvesinin doğru olarak tespit edilme oranını göstermektedir.

$$\text{Duyarlılık } (Recall) = \frac{TP}{(TP+FN)} \tag{3.10}$$

F1 Skor: Sınıflandırma algoritmalarında sıklıkla kullanılan ölçülerden olan F1 Skor değeri için duyarlılık ve kesinlik değerleinin bulunup bunların harmonik ortalamalarının alınması gerekmektedir.

$$\text{F1Scorer} = 2x\frac{Precision\ x\ Recall}{(Pecision+Recall)} \tag{3.11}$$

Doğruluk (Accuracy): Doğruluk oranını tanımlamaktadır. Veri setindeki tam olgun, olgun, yarı olgun, olgunlaşmamış ve çiçek gibi olgunlaşma tanımlarının doğru olarak tespit edilme oranını göstermektedir.

$$\text{Doğruluk (Accuracy)} = \frac{Doğru\ Tahminler}{Toplam\ Tahminler} = \frac{TP+TN}{TP+TN+FP+FN} \tag{3.12}$$

TP: Önerilen modele göre veri setinde doğru olarak sınıflandırılan çilek meyvesi sayısını; Kullanılan modele ait veri setinde Doğru Pozitif (True Positive) olarak sınıflandırılan çilek sayısını ifade etmektedir.

FP: Yine kullanılan modelde eğitilen veri setindeki istenmeyen, hatalı Yanlış Negatif sayısını göstermektedir.

TN: Veri setindeki doğru olarak adlandırılan hatalı, istenmeyen çilek veri sayısını belirtmektedir.

FN: Kullanılan modele göre istenmeyen değer olarak hatalı, yanlış sınıflandırılan çilek meyvesi sayısını tanımlamaktadır.

3.4 Evrişimsel Sinir Ağları

Çok katmanlı yapay sinir ağlarının biyolojik sinir ağları gibi çalışarak derin sinir ağlarını oluşturduklarından daha önce bahsedilmişti. Bunlar içerisinde en yaygın olarak kullanılan derin sinir ağlarından biri evrişimsel sinir ağı modeli olarak görülmektedir. Evrişimsel sinir ağlarına ait en karakteristik özellik örüntü tanıma işlemlerinde yapay sinir ağına air kullanılan parametre sayılarını azalmaktır. Bu sayede eğitim sırasında görüntünün tamamının aynı anda kullanılmasına gerek duyulmamaktadır. İşlem sırasında farklı boyutlarda filtreler kullanılarak aynı anda işlenebilecek maksimum

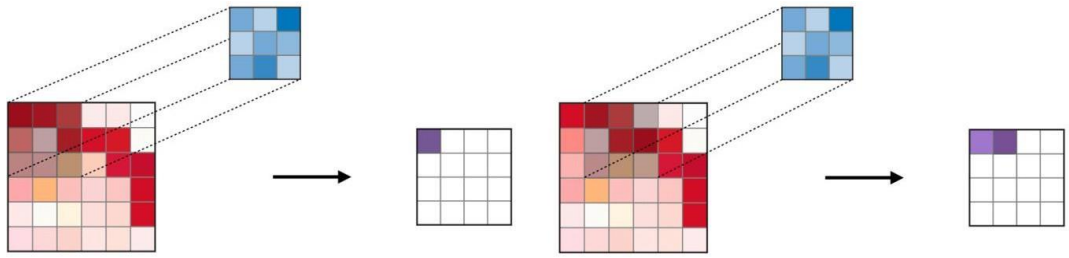
parametre sayını ciddi oranda azaltılabilmektedir. Evrişimsel sinir ağlarına ait bir diğer önemli faktör giriş görüntüsü daha dip katmanlara doğru yayıldıkça daha soyut özellikler öğrenilebilmektedir. Yüz tanımlama uygulamalarında giriş katmanlarında kenarlara ait özellikler, daha dip katmanlarda ise yüz hatları gibi ayırt edici özellikler tespit edilebilmektedir (Albawi vd., 2017).

Evrişimsel sinir ağlarına bakıldığında evrişim katmanı, havuzlama katmanı ve tam bağlantılı olmak üzere üç tür katmandan oluştuğu görülmektedir (O'Shea ve Nash, 2015).

3.4.1 Evrişim Katmanı

Girdi görüntülerinin evrişim işlemine tabi tutulduğu filtrelerin bulunduğu katmandır. Bu katman sinir ağının ana yapısını teşkil eden katmanlardan biri olarak görülmektedir. Kullanılan filtreler giriş görüntülerin boyutlarına göre taranması amacıyla kullanılmaktadır. Evrişim katmanı, filtre boyutu, adım kaydırma ve dolgulama parametrelerinin de bulunduğu katmandır. Bu katmanda oluşan çıktılar öznitelik veya aktivasyon haritası şeklinde adlandırılmaktadır (Amidi ve Amidi, 2018).

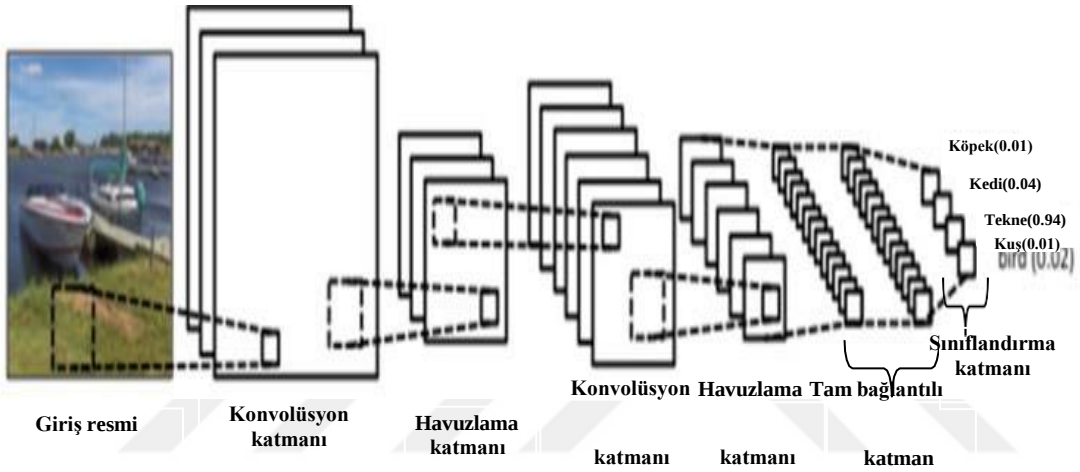
Şekil 3.20'de evrişim işlemine ait ilk adımlar gösterilmektedir. Başlangıç görüntülerinin sol üst kısmından başlanıp görüntü tanıma filtrelerinin sırasıyla uygulanması evrişim işlemi olarak tanımlanmaktadır.



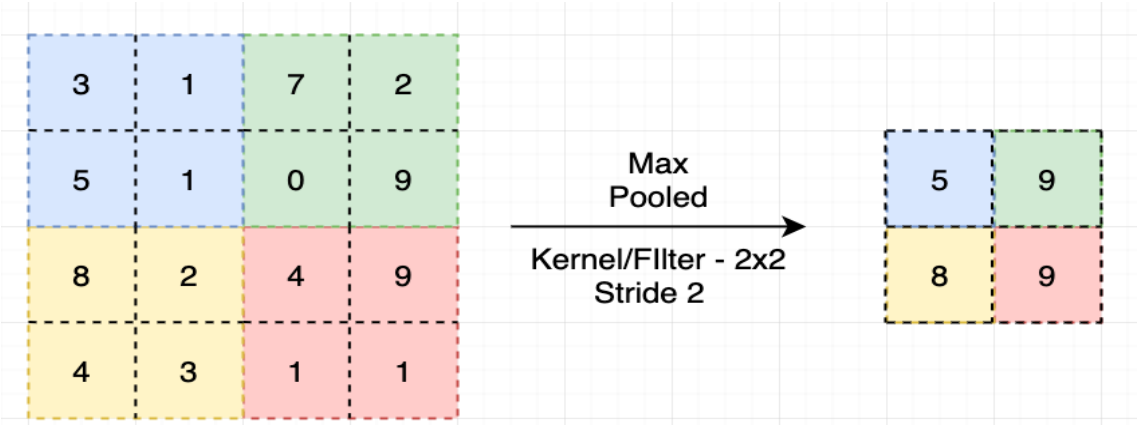
Şekil 3.20 Evrişim işlemi (Amidi ve Amidi, 2020)

3.4.2 Havuzlama Katmanı (Pooling Layer)

Konvolüsyon (dönüşüm) katmanından hemen sonraki katmandır. Belirli bir filtrenin tüm görüntü üzerinde dolaştırılması işlemi olan dönüşüm katmanından sonra diğer bir konvülsiyon katmanı için giriş boyutu (Genişlik x Yükseklik) düşürülmektedir (Şekil 3.21). Giriş boyutunun küçültülmesi (şekil 3.22) veride derinlik boyutunu etkilememektedir. Aşağı örnekleme olarak bilinen bu işlem sonucu boyuttaki azaltma bilgi kaybına yol açmaktadır. Tabi bu kayıp şu iki nedenden dolayı fayda sağlar, birinci küçük verilerle yapılacak işlemler sonraki katman üzerindeki iş yükünü azaltacak bir diğer faydası ise sistemin ezberlenmesini önleyecektir (Anonim, 2021a).



Şekil 3.21 Evrişimsel sinir ağının genel mimarisi (Ding vd., 2016)



Şekil 3.22 Maksimum havuzlama katmanı (Rana, 2020)

3.4.3 Tam Bağlantılı Katman

Derin ağ mimarilerinde konvolüsyon ve havuzlama katmanlarından sonra tam bağlantı katmanı gelmektedir. Tam bağlantı katmanlarında sınıflandırma işlemleri gerçekleştirilmektedir. Bu katmanda bulunan çıkış değeri sınıflandırması yapılacak nesne sayısına eşit olmaktadır.

Tam bağlantı katmanları ileri yönlü yayılım gösteren katmanlardır. Evrişimsel sinir ağları bu katmanla ileri yönde beslenmektedir. Kendinden önce verileri işleyen havuzlama ve aktivasyon katmanlarından gelen veriler bir vektör olacak şekilde düzleştirilmektedir. Düzleştirilen bu vektör ağırlık ve yanlılık değerleriyle birlikte matematiksel olarak hesaplanmaktadır (Arunava, 2018).

3.5 Evrişimsel Sinir Ağı Modelleri

Evrişimsel sinir ağı modellerinde görüntü algılama, işleme ve tanımlama işlemleri gözün yapını örnek almaktadır. Bu ağlar kullanılarak yapılan işlemler iki şekilde ele alınmaktadır. Bunlardan ilki olan tek aşamalı nesne tespit dedektörleri doğrudan evrişimsel sinir ağı çalışması ile nesne algılama yapmaktadır. RetinaNet, SSD ve YOLO tek aşamalı nesne tespit modelleridir. Bir diğer işlem ise iki aşamalı nesne tespit dedektörleri olan R-CNN, Fast-RCNN ve Faster-RCNN modelleridir. Nesne tespiti sırasında önce bölge önerisi (Tahmini) ikinci aşamada ise önerilen bölgelerde nesne algılama çalışmaları yapılmaktadır. Yapılan bu tezde tek aşamalı nesne tespit modellerine ait tanım ve uygulama kısmında yine bu modellere ait performans karşılaştırmaları yapılmıştır.

3.5.1 YOLO

YOLO Sadece Bir Kez Bak (You Only Look Once) nesne tanıma sırasında sınırlayıcı kutular kullanan ve sınıfa ait tahminler için sadece tek bir sinir ağı kullanan ve GoogleNet'ten ilham alan bir evrişimsel sinir ağı modellerinden biri olarak tanımlanabilmektedir. Bu tanıma göre görüntünün algılanması işlemi için tek aşama yani bir kez ağ içerisinden geçilmesiyle yapılmaktadır. Literatürde en hızlı nesne algılama algoritması YOLO'dur. Tüm algılama hattı tek bir ağda olduğundan özellikle

hız konusunda tercih edilen bir mimari olma özelliği göstermektedir. Aşağıdaki şekil 3.23'te evdeki kitaplığımda bulunan onlarca kitabın yolo ile anlık tespiti gösterilmiştir.

Şekil 3.23 YOLO ile anlık nesne tespiti örneği

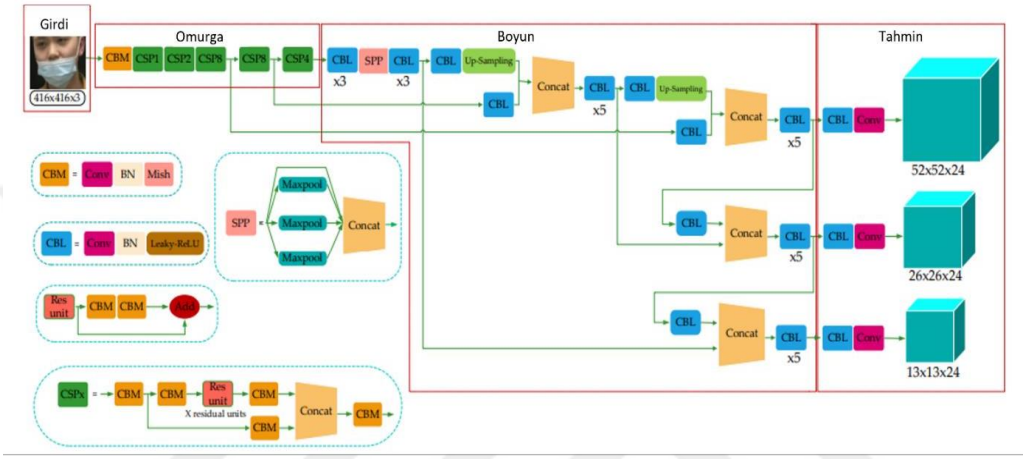
- Nesne tespiti sırasında sırasıyla;
- Girdiyi boyutlandırma
- Konvülasyon ağını çalıştırma
- En yüksek skorlu sınırlayıcı kutuyu seçme

İşlemleri uygulanmaktadır. YOLO açık kaynaklı ve C ve CUDA dilinde yazılmış bir sinir ağı uygulama çatısı olan "DarkNet" üzerinden uygulanmaktadır. Bu şekilde video akışını anlık işlemek için GPU işlem gücünden yararlanmaktadır. Yolo nesne tespiti yaparken tahminler için en yakın ihtimallerden yola çıkmaktadır ve en yüksek tahmin oranı hedef nesne için kullanılmaktadır. Nesne tespiti sırasında görüntünün yakınlığı, netliği, eğimi, büyüklüğü vb. faktörler nesne tespiti doğruluk oranını doğrudan etkilemektedir (Sevi vd., 2023).

3.5.2 YOLOv4

Sürekli güncellemeler alan YOLO nesne tanımalarından YOLOv4 Nisan 2020'de yayımlanmıştır. COCO veri seti üzerinden yapılan uygulamalarda yüksek performans sonuçları gösteren gerçek zamanlı (Real Time) bir nesne algılama algoritmasıdır. YOLO'nun her modeli bir önceki modele optimize edilerek geliştirilmektedir. YOLOv4 modeli de YOLOv3'e optimize edilmiştir. YOLOv4 yapay

sinir ağlarında anlatıldığı üzere omurga, boyun ve tahmin olmak üzere 3 kısımdan meydana gelmektedir (Şekil 3.24). Omurga kısmında Darknet53 ağı bulunmaktadır. Omurga kısmında ayrıca BoF (Bag of Freebies) ve BoS (Bag of Specials) adlı iki bölüm daha bulunmaktadır. Bu bölümler veri setini zenginleştirmek için kullanılmaktadır (Güçlü vd., 2021). Boyun, algılanan nesneleri farklı şekillerde ölçeklendirmede kullanılmaktadır. Tahminler ise kafa kısmında yapılmaktadır.

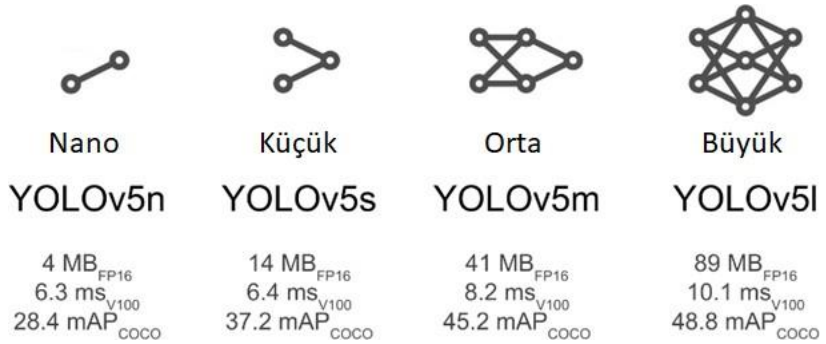


Şekil 3.24 YOLOv4 mimarisi (Zakria vd., 2022)

3.5.3 YOLOv5

YOLOv4'ten kısa bir süre sonra tasarımcıları olan Gleen ve ekibi tarafından YOLOv5 sürümü çıkmıştır. YOLOv5 önceki modellerine göre görünüt işleme sürelerinin kısaltılması noktasında oldukça güzel sonuçlar vermiştir. YOLOv5 Darknet yerine Pytorch eğitim ortamı altında derlenerek belirlenen nesne algılama işlemlerini daha kısa sürede tamamlamıştır. Ayrıca YOLOv5'in ağırlık dosya boyutlarıda önceki sürüm olan YOLOv4'ten %90 oranında düşüktür.

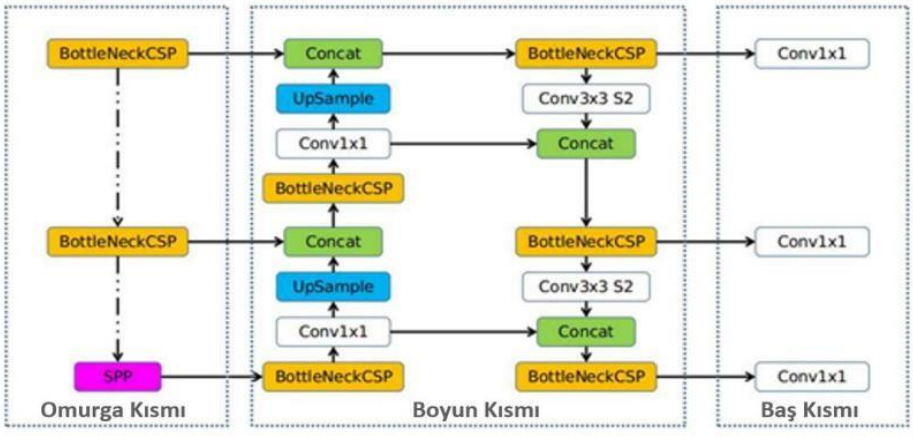
YOLOv5 bu özellikleri ile göümülü sistemlerde gerçek zamanlı tespit işlemleri için kullanışlı hale gelmektedir. Boyut ve model parametre miktarlarındaki artışa göre 4 alt model içermektedir. Bunlar; YOLOv5s, YOLOv5n, YOLOv5m ve YOLOv5l alt modelleridir. Bu alt modellerin temel farkı özellik çıkarma modüllerinin ve evrişim çekirdek miktarlarındaki farklılıktan kaynaklanmaktadır. Alt model trülerinin parametre sayısı ve buna bağlı boyutları Şekil 3.25'te gösterildiği gibi artış göstermektedir.



Şekil 3.25 YOLOv5 modeline ait alt mimariler

Bu tezde nesne algılama işlemlerinde sistemin minimum kaynak tüketimi ile maksimum performans sonucuna ulaşması beklendiği için YOLOv5 modeli tercih edilmiştir. YOLOv5 önceki sürümde olduğu gibi omurga, boyun be baş olmak üzere yine üç kısımdan oluşmuştur.

YOLOv5 omurga bölümünde CSPDarknet, boyun bölümünde PANet ve baş kısmında YOLO katmanı kullanılarak oluşturulmuştur (Şekil 3.26). İşleyişe bakıldığında kısaca veriler CSPDarknet'te özellik çıkarım işlemine tabi tutulmakta, PANet'te özellik birleştirme işlemi ve son aşama olan YOLO katmanında nesneye ait sınıf bilgisi, skor değerleri, koordinat bilgileri ve boyut bilgilerine ait çıktılar yer almaktadır (Güney, 2021).



Şekil 3.26 YOLOv5 mimarinin genel yapısı (Xu vd., 2021)

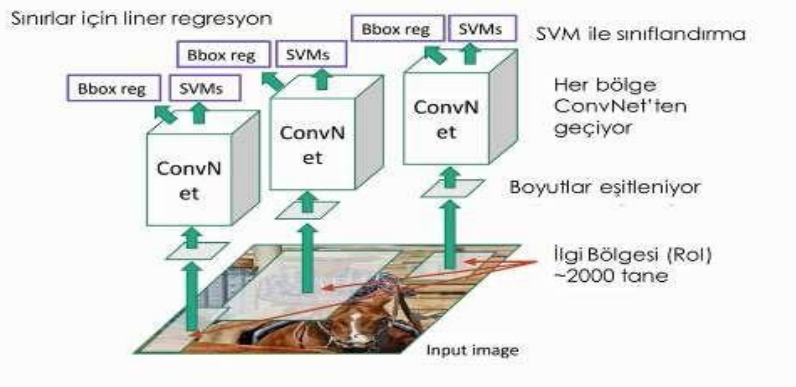
3.5.4 R-CNN, Fast R-CNN ve Faster R-CNN

Görüntü algılama tespit işlemlerinde görüntüye ait tüm pencereleri gezmek yerine bölge önerisiyle tespit edilen 2000 pencere üzerinden tahmin yapmak işlemi kolaylaştıracaktır. R-CNN (Region-based Convolutional Neural Network)'de bu şekilde çalışmaktadır (Girshick vd., 2014).

Belirtilen pencereler farklı boyutlarda olabileceği için R-CNN'de evrişimli sinir ağları (ESA)'dan geçirmek için farklı boyutlar eşitlenmektedir. Bu şekilde ESA'larda belli standart boyutlar geçirilmektedir. Sınıflandırma yapmak için ise DVM (gözetimli makine uygulamalarında kullanılan makine öğrenme metodu) kullanılmaktadır (Vapnik, 1999). Nesnelerin sınırları ise lineer regresyon kullanılarak tespit edilmektedir. Sınırlayıcı kutular bu şekilde dörtgenler olarak çizilmektedir.

> *"Lineer regresyon, bir veya birden fazla bağımsız değişken ile başka bir bağımlı değişken arasındaki bağlantıyı modellemek için kullanılan bir yöntemdir" (Anonim, 2023g).*

Sonuç itibariyle R-CNN nesne arama ve bulma işlemlerinde nesneye ait sınıfı döndürecek ve nesnenin görüntü içerisindeki yerini belirten 4 değer üretecektir. Üretilen bu değerler YOLO mimarisine olduğu gibi görüntü etrafında çizilecek olan kutucukların koordinat değerlerini içermektedir (Girshick vd., 2014). Aşağıda (Şekil 3.27)'de R-CNN modeli mimarisi gösterilmektedir.



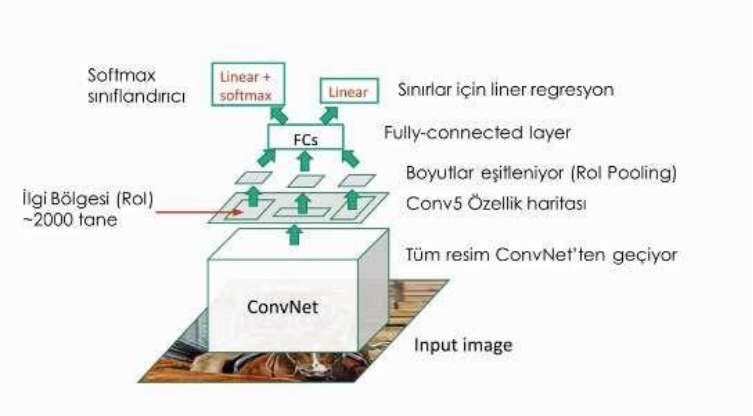
Şekil 3.27 R-CNN algoritması mimari yapı (Girshick vd., 2014)

R-CNN algoritması ile görüntü işleme yaparken görülen eksikliklere bakıldığında Fast R-CNN ve Faster R-CNN' de yapılan geliştirmeler daha net görülebilecektir (Girshick vd., 2014).

İlk olarak evrişim katmanlarından teker teker geçen tahmin bölgeleri R-CNN'i yavaş çalışan bir ağa dönüştürmektedir. Yaşanan bu zaman kaybı sorununu ortadan kaldırmak için Fast R-CNN modeli geliştirilmiştir. Fast R-CNN mimarisi yapı olarak R-CNN' benzemekle birlikte hızlandırmak için giriş görüntüsü üzerinde bölge önerisi yapmak yerine görüntü doğrudan evrişimli sinir ağından geçirilmektedir. Sonrasında üzerinde bölge önerileri olan yüksek çözünürlüğe sahip bir özellik haritası oluşturulmaktadır. Bu sayede ham görüntü üzerinde bölge önerisi ile zaman kaybı yaşanmadan işlem özellik haritası üzerinden bir defa yapılarak zaman kazanılmaktadır. Özellik haritalarında yine farklı boyutlar eşitlenir ve sonuçlar tam bağlantı katmanına aktarılarak işlem devam etmektedir. Sonrasında sınıflandırma yapılmakta ve tespit edilen nesnelere ait sınırlar çizilmektedir.

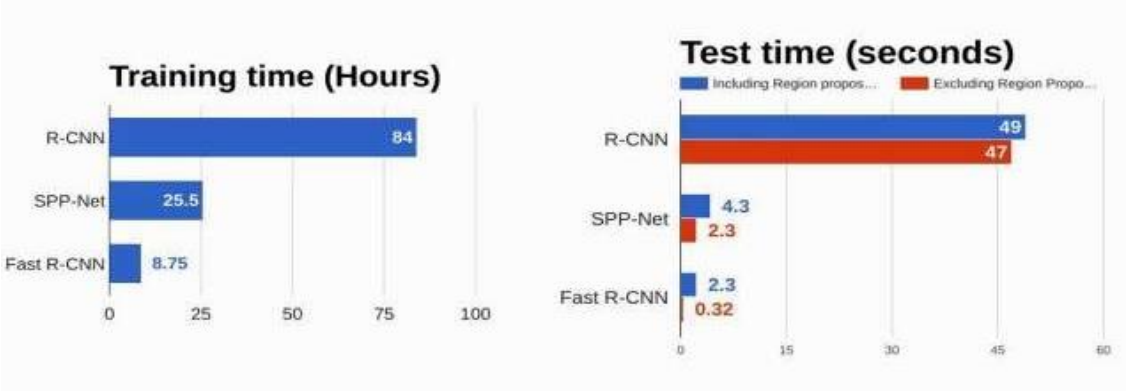
> *"Fast R- CNN' i R-CNN den ayıran bir farkta bu aşamada oluşmaktadır. R-CNN de sınıflandırma yapmak için DVM kullanılmaktayken Fast R-CNN de ise bu işlem için Softmax katmanı kullanılmaktadır" RCNN (Girshick vd., 2014).*

Şekil 3.28'de Fast R-CNN Yapısı ve görsel işleme katmanları gösterilmektedir



Şekil 3.28 Fast R-CNN mimari yapısı (Vapnik, 1999)

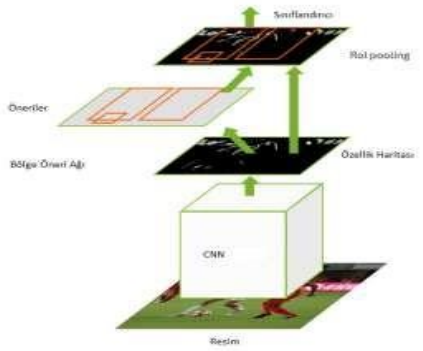
Şekil 3.29' daki değerler Fast R-CNN in 100 iterasyonluk zaman-performans farkını net bir şekilde göstermektedir. Buna göre Fast R-CNN aynı data set üzerinde 2.3 dk ile R-CNN'in aynı data seti 49 dk'da işlemesiyle oldukça yüksek hız değerine sahip olduğunu göstermiştir.



Şekil 3.29 VOC 2011 Fast R-CNN-Faster R-CNN karşılaştırması (Vapnik, 1999)

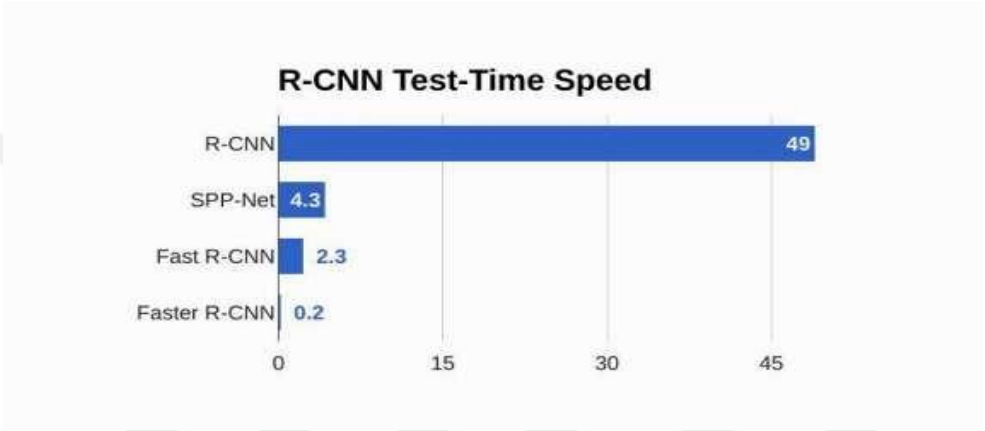
Yukarıdaki değerler (Şekil3.29) Fast R-CNN'in eğitim aşamasında oldukça hızlı olduğunu göstermektedir. Ancak bölge önerilerinin yapıldı test aşamasında mevcut zamanın çoğunu harcamaktadır. Modelin hızının daha da artması için bölge önerisinde kullanılan zamanın kısaltılması gerekmektedir. Faster R-CNN bölge önerilerinde harcanan zamanı kısaltmak için geliştirilmiş üst versiyon durumundadır. Bölge önerileri hız kazanmak için R-CNN'den farklı olarak ağ içerisinde yapılmaktadır (Vapnik, 1999).

Şekil 3.30'da Faster R-CNN in Mimari yapısı gösterilmektedir. Şekilde de görüldüğü gibi evrişimsel işlemler daha kısa yoldan sonuçlandırılmaktadır.



Şekil 3.30 Faster R-CNN Mimari Yapısı (Ren vd., 2015)

Faster R-CNN ağ mimarisi incelendiğinde giriş görüntüleri evrişimsel sinir ağlarından geçirilmekte ve bölge haritaları oluşturulmaktadır. “Bu kısımda r-CNN’den farklı olarak bölge önerileri RPN (Bölge Öneri Ağı-Region Proposal Network) aracılığıyla yapılmaktadır. RPN bölgeleri belirledikten sonra geri kalan işlemler R-CNN ile aynı olmaktadır. Bu üç farklı model PASCAL VOC veri seti üzerinde test edilmiş ve Şekil 3.31’deki gibi sonuçlar elde edilmiştir.” R-CNN test sonuçları incelendiğinde yapılan iyileştirmelerin doğru yönde çalıştığı görülmektedir.



Şekil 3.31 R-CNN-Fast R-CNN ve Faster R-CNN (Everingham vd., 2005)

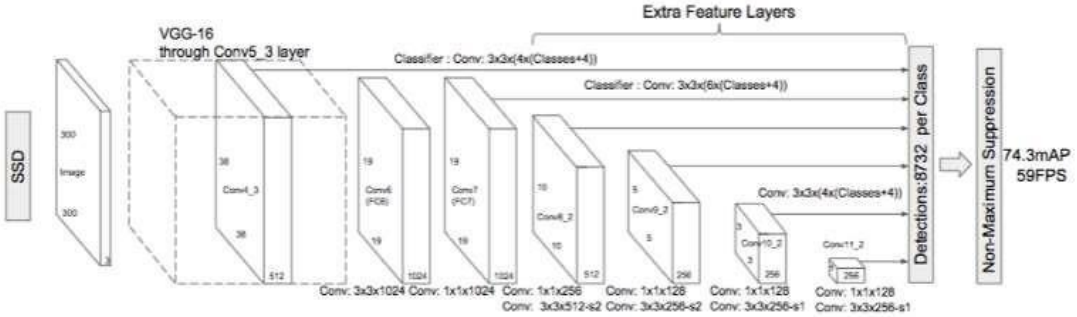
3.5.5 SSD MobileNet

SSD (Single Shot MultiBox Detector) MobileNet tek adımlı nesne tanıma algoritmasıdır. Bölge önerisi ve sınıflandırmayı iki aşamalı olarak gerçekleştirmektedir. R-CNN’de ilkin nesne bulunma ihtimali olan kısımlar tespit edilerek sınıflandırılmaktayken, SSD MobileNet tespit ve sınıflandırma işlemlerini tek sferde yağmaktadır. İlk aşama giriş resmi convolüsyonel sinir ağından geçirilerek farklı özellik haritaları çıkartılır. Bu özellik haritalarında 3x3’lük filtrelerle küçük sınırlayıcı kutular oluşturulmaktadır. İşte bu sınırlayıcı kutular ile hem sınırlar hemde sınıflandırmalar yapılabilmektedir. Oluşturulan bu kutular aktivasyon haritasında olduğu için küçük ve büyük ölçekli nesneleri tanımlayabilmektedir.

“Eğitim esnasında doğru olan sınırlar ile tahmin edilen sınırlar karşılaştırılmaktadır. En iyi tahmin yapan

ve 0.5 oranının üzerindeki dikdörtgenler pozitif olarak etiketlenmektedir" (Liu vd., 2016).

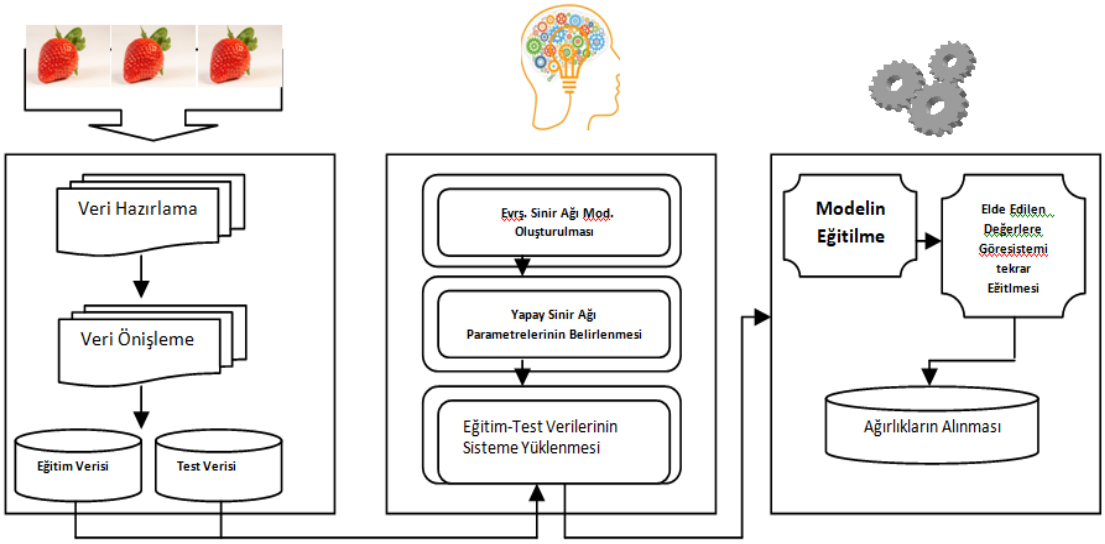
Şekil 3.32' de SSD modelinin mimarisi ve katmanları gösterilmektedir.



Şekil 3.32 SSD model mimari yapısı (Liu vd., 2016)

4. GERÇEKLEŞTİRİLEN UYGULAMA VE PERFORMANS SONUÇLARI

Gerçekleştirilen tez çalışmasına ait uygulamalar Şekil.4.1'de gösterildiği gibi temel olarak üç aşamadan meydana gelmektedir. İlk aşamada veri setinin oluşturulması ve alt işlemler, ikinci adımda derin öğrenme için kullanılacak evrişimsel sinir ağı modellerinin oluşturulması ve konfigürasyonu, üçüncü adımda ise modelin eğitimi ve çıkan sonuçların karşılaştırılması yer almaktadır.



Şekil 4.1 Temel sistem mimarisi

3.6 Kullanılan Veri Seti

Tez çalışmasında kullanılan veri seti Antalya ilimizin Kepez ilçesindeki anlaşmalı bir çilek serasından çekilen fotoğraflardan ve internetten bulunan görsellerden oluşan toplam 540 resimden elde edilmiştir. Bunların 399 adedi eğitim veri seti için, rastgele seçilen 121 tanesi eğitimler sonrası reel time test için kullanılmıştır. Seradan çekilen fotoğraflar için (Şekil 4.2) xaomi redmi note 8 cep telefonu kamerası kullanılmıştır. Fotoğraflar 1080x706 boyutunda, 756 Pixel PNG ve ortalama 1.40 MB standart boyutlarda arşivlenmiştir. Veri seti hazırlama ve önişleme evresinde, seradan çekilen ve internetten elde edilen görsel kaynaklar (Şekil 4.3) boyutlama, ayıklama, netleştirme vb. işlemlerden geçirildikten sonra etiketlenerek kullanılabilir veri seti haline getirilmiştir.

Şekil 4.2 Seradan elde edilen görüntüler

Şekil 4.3 İnternetten elde edilen görüntüler

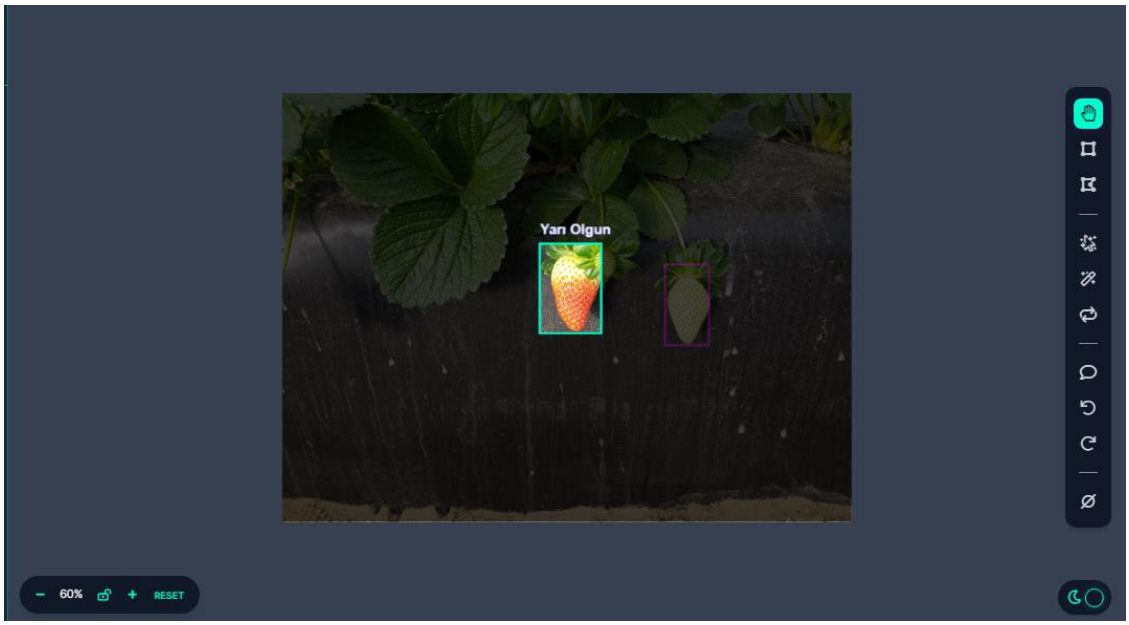
4.1.1 Veri Çoğullama

Çıktı olarak işlenen görüntülerin sisteme uyacak bir şekilde boyutlandırılması işleminden sonra Roboflow görüntü işleme uygulamasıyla çevrimiçi olarak veri çoğullama işlemi yapılmıştır. Eğitim için kullanılan 399 fotoğraftan veri çoğullama metotlarından olan ters çevirme ve yansıma ile 1594 adet resim elde edilmiştir. Bu sayede her bir görselin yansımalı ve ters görüntüleri elde edilerek hem eğitilen fotoğraf sayısı arttırılmış hem de eğitim sonrası farklı açılardan gelebilecek görüntüler için sistem hazır hale getirilerek olası ihtimaller öngörülmüştür.

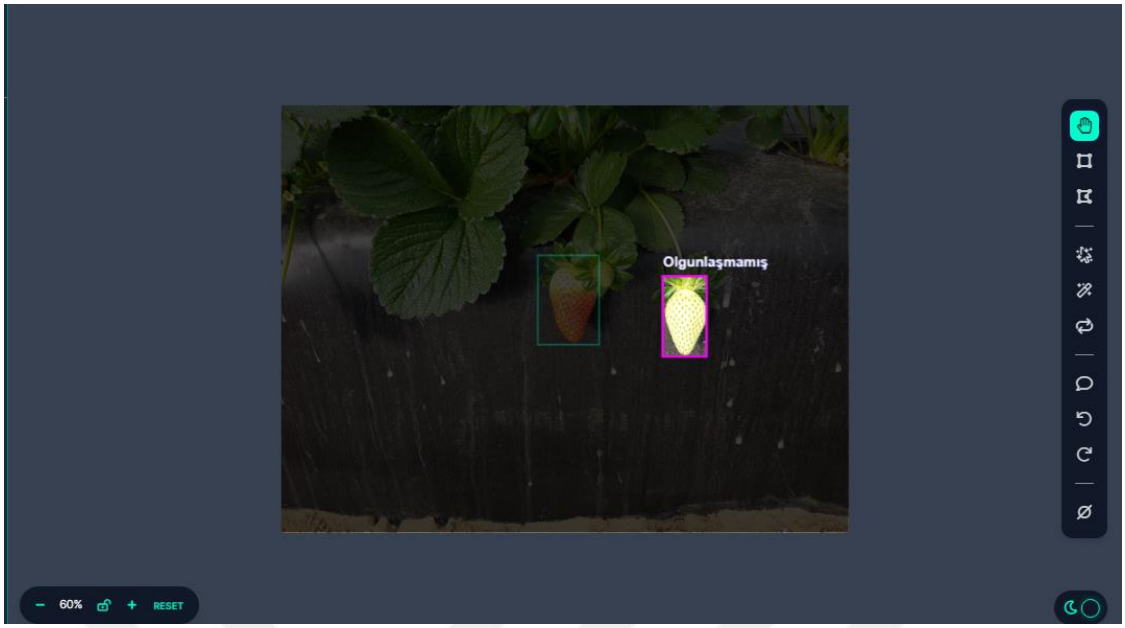
"Evrişimsel sinir ağları ile danışmanlı öğrenmede, toplanması ve hazırlanması oldukça maliyetli olan çok sayıda etiketli veriye ihtiyaç duyulmaktadır" (Geng vd., 2021).

4.1.2 Etiketleme İşlemi

Eğitim sonrası elde edilen tüm verileri etiketlemek gerekmektedir. Verilerin danışmanlı öğrenme yöntemi ile yapay sinir ağlarında kullanmak için etiketleme yapılması zorunluluktur. Tez çalışması yapay zeka metodlarıyla olgunluk kıyaslaması temeline dayandığı için veriler kategorize edilerek etiketlenmiştir (Şekil 4.4). Etiketleme tam olgun, olgun, yarı olgun (Şekil 4.4), olgunlaşmamış (Şekil 4.5) ve çiçek olmak üzere beş farklı kategoride yapılmıştır. Ham veri setindeki 399 fotoğraftan veri çoğullama ile 1594 adet görsel elde edildiğinden bahsedilmişti. Etiketleme için kullanılan her bir görselde çiçek, olgunlaşmamış, yarı olgun, olgun ve tam olgun sınıflarına ait 3 ila 7 farklı görsel olabildiğinden etiketlenen nesne sayısı aritmetik ortalama ile 7970 adedi bulabilmektedir. Etiketlenen nesne sayısındaki artış eğitim sonrası modelin doğru sonuç üretmesi için en önemli kriterlerden biri olarak bilinmektedir.



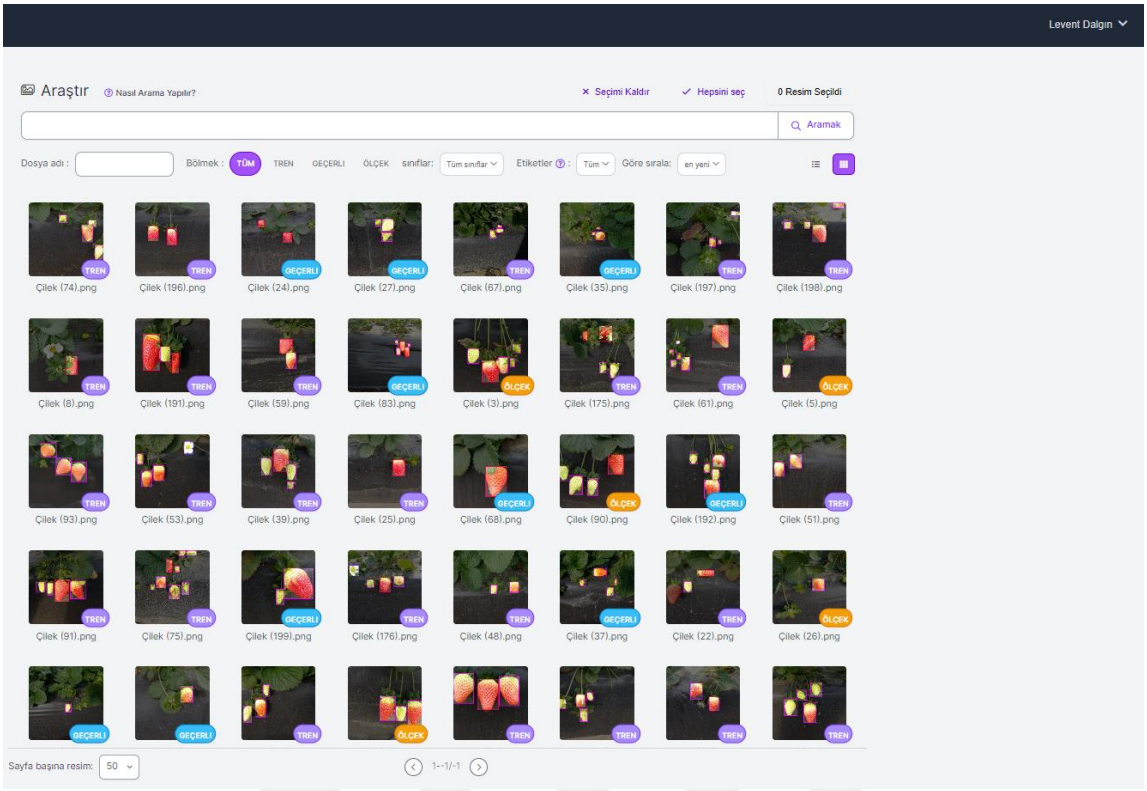
Şekil 4.4 Etiketleme (Yarı olgun çilek)



Şekil 4.5 Etiketleme (Aynı görselde olgunlaşmamış çilek)

Şekil 4.6 Veri çoğullama 90 derece döndürülmüş bir resimde etiketleme işlemi

Makine öğrenmede hedef, tahminlerin minimum hata ile yapılmasını sağlamaktır. Bunun için kullanılan veri setleri tahmin sonrası sağlama işlemi yapmak için eğitim, test ve doğrulama olarak bölümlere ayrılmaktadır (Şekil 4.7).



Şekil 4.7 Veri seti içerisinde eğitim, test ve doğrulama gösterimi

4.1.3 Veri Seti Sınıf Dağılımı

Veri setinde çoğullama ve her bir görsel içerisindeki farklı sınıflara ait nesnelerin etiketlenmesi işleminde 7970 adet etiketleme yapıldığı anlatılmıştı. Farklı sınıflara ait nesnelerin toplam veri seti içerisindeki ortalama sayısal dağılımı aşağıdaki çizelgede gösterilmiştir.

Çizelge 4.1 Veri seti içerisindeki etiketlenen nesne sayısı dağılımı

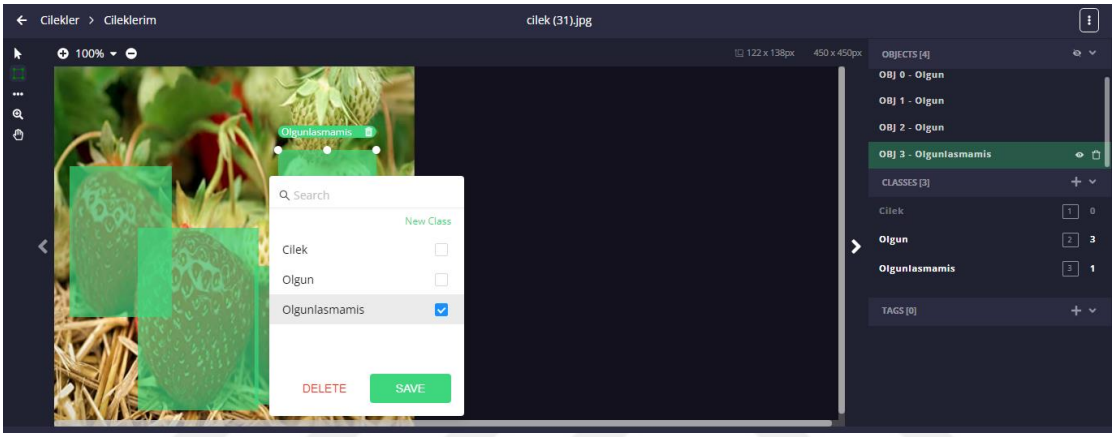
Sınıf	Çiçek	Olgunlaşmamış	Yarı Olgun	Olgun	Tam Olgun	Toplam
Etiket Sayısı	160	5652	1200	399	559	7768

Veri seti için hazırlanan 540 resimden homojen olması için rastgele 399 adedi 3 farklı model tarafından eğitilmiş, geriye kalan 141 adet resim eğitimler tamamlandıktan sonra modellerin sağlıklı çalışıp çalışmadığını kontrol etmek için kullanılmıştır. Real time test sonuçları çalışmanın sonunda sunulmuştur.

Test için kullanılan 141 adet resim için ortalama sınıf saysı dağılımı aşağıdaki çizelgede (Çizelge 4.2) gösterilmiştir.

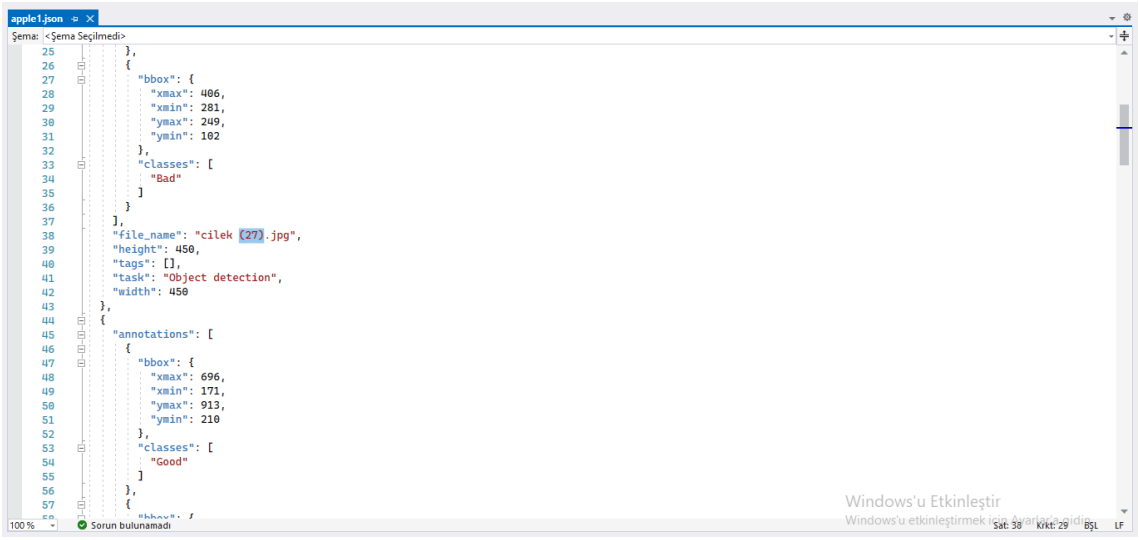
Çizelge 4.2 Test için ayrılan veri seti içerisindeki etiketlenen nesne sayısı dağılımı

Sınıf	Çiçek	Olgunlaşmamış	Yarı Olgun	Olgun	Tam Olgun	Toplam
Etiket Sayısı	53	1884	332	133	186	2588

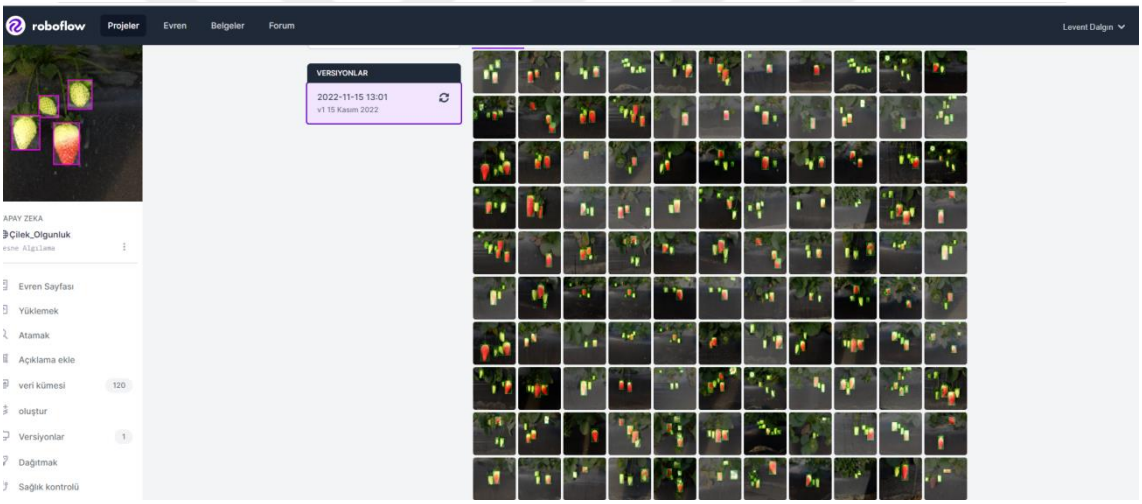


Şekil 4.8 İnternetten alınan görsellerin etiketlenme süreci

Etiketlenen her bir resim için x, y koordinat bilgilerini içeren txt uzantılı dosyalar oluşturulur. Bu .txt dosyalarında her resim için beş satırlık işlem yapılır; resim etiket numaralandırma, x koordinatı, y koordinatı, genişlik ve yükseklik değerleri belirlenir. Veri seti eğitimi sonrası oluşan bu koordinat bilgileri yapay zeka algoritmalarının nesnenin konum bilgilerinden yola çıkarak nesneyi tespit etmesini ve benzer resimler için yine benzer konum bilgileri kullanmasını kolaylaştırarak nesne tespit işlemlerinin hızlı ve doğru sonuçlanmasını sağlar. Veri setlerine ait etiketlerin YOLOv5 Pytorch için .txt formatında diğer mimariler için ise JSON uzantılı olarak hazırlanmasının sebebi sinir ağı modellerinin veri setlerini eğitirken farklı formatlar kullanmalarından kaynaklanmaktadır. Elde edilen .txt dosyalara ait içerikler ve görsellere ait koordinat ve boyut bilgileri Şekil 4.9'da gösterilmiştir.

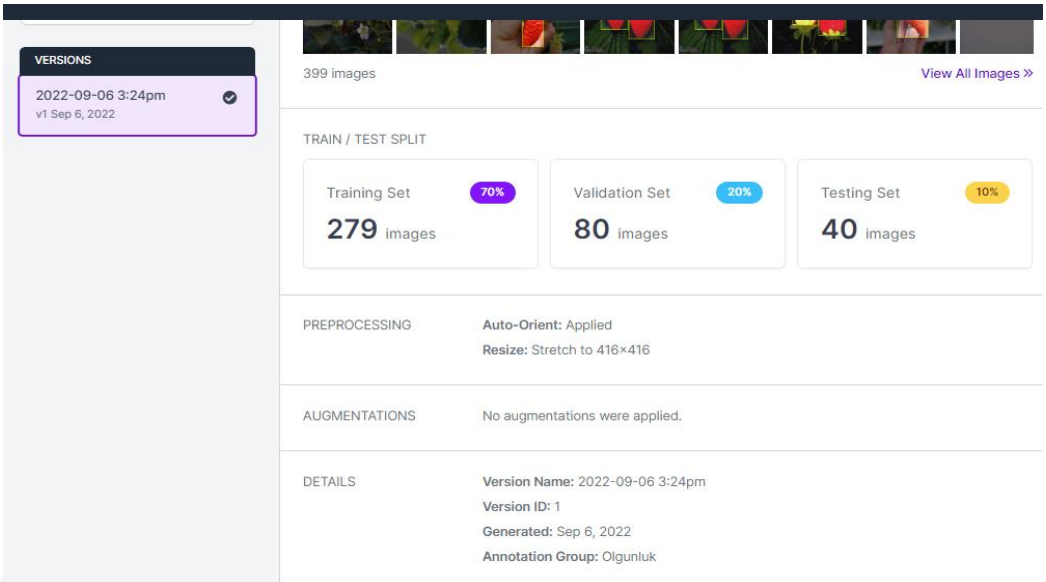


Şekil 4.9 Etiketlenen verilere ait koordinat ve boyut bilgileri



Şekil 4.10 Etiketleme sonrası veri setinden bir görüntü

Etiketleme işlemi bittikten sonra şekil 4.11'de gösterildiği gibi veriler eğitim, doğrulama ve test olmak üzere üç bölüm halinde ayrıştırılır. Veri seti bilgisayarlı görü işlemlerinde standart olarak yapıldığı gibi %70 eğitim, %20 doğrulama ve %10 test oranı belirlenerek üç kısma ayrılmıştır. Kullanılan veri seti sonuç olarak 279 adet eğitim, 80 adet doğrulama ve 40 adet test görüntüsü oranlarıyla oluşturulmuştur.

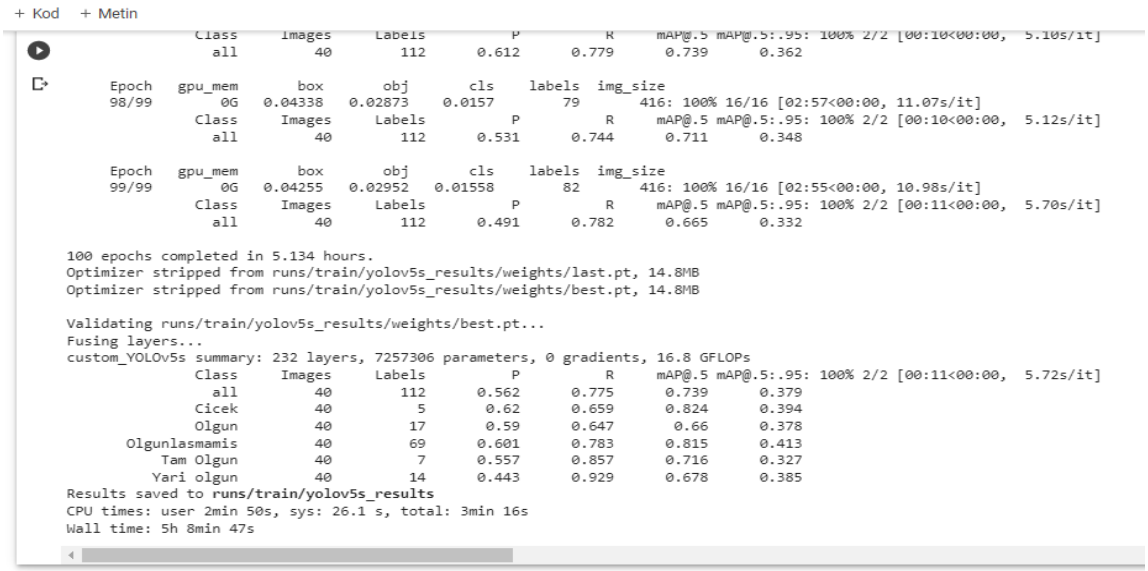


Şekil 4.11 Veri setinin eğitim, doğrulama ve test olarak ayrıştırılması

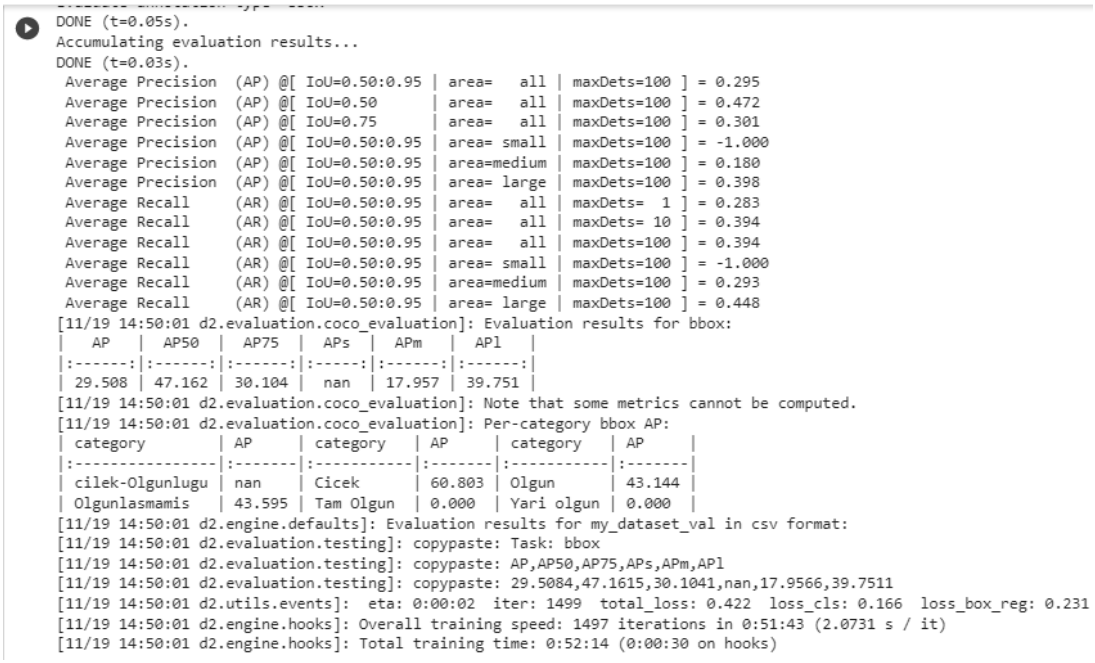
4.1.4 Verilerin Eğitilmesi

ESA işlemlerinde verilerin eğitilmesi bilgisayarları oldukça yormaktadır. Bu sebepten ötürü yüksek işlem hacimli işlemciler ve ekran kartları tercih edilmektedir. Tez aşamasında veriler modeller tarafından eğitilirken yüksek performanslı cihazlara sahip olunmadığı için alternatif bir çözüm üretilmiş, veri setleri eş zamanlı olarak Google Coloboratory uzak sunucu bilgisayarlarıyla eğitilmiştir. Bu sunucu bilgisayarın işletim sistemi Linux4.19.112+-x86_64-with-Ubuntu-18.04-bionic + ekran kartı modeli Tesla P100-PCIE16GB olmak üzere eğitimlerde ciddi yararlılıklar göstermiştir.

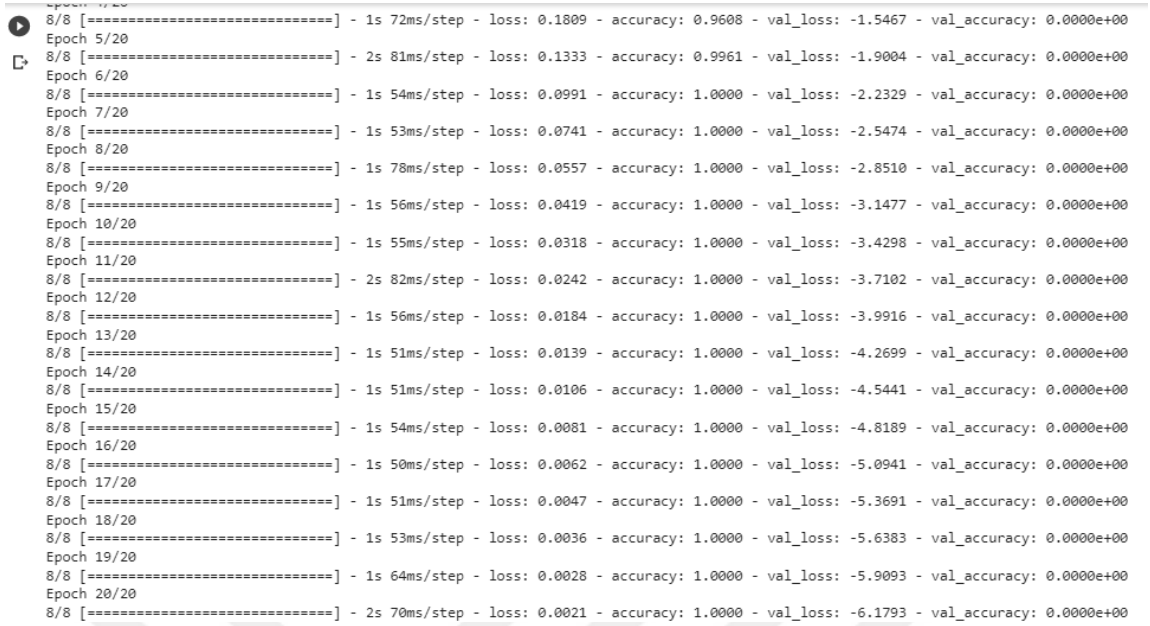
Verilerin eğitilmesi veri setinin büyüklüğüne bağlı olarak saatler hatta günler alabilmektedir. Aşağıda (Şekil 4.12) Yolov5 algoritmasına ait 100 iterasyonluk eğitim sürecinden, FasterCNN algoritmasının 1497. İterasyonundan bir ekran görüntüsü (Şekil 4.13) ve SSDMobilNet eğitim sürecinden bir görüntü (Şekil 4.14) bulunmaktadır.



Şekil 4.12 Yolov5 algortimasının 100 iterasyonluk eğitim süreci



Şekil 4.13 FasterCNN Algoritmasının 1497 iterasyonluk eğitim süreci



Şekil 4.14 SSD MobilNet v2 eğitim sürecinden bir görüntü

3.7 Değerlendirme Metrikleri

Tez çalışması için kullanılan veri seti eğitim sonunda ulaşılan kesinlik, doğruluk, başarı ve verim gibi değerlerin reel ölçütlerle uyumlu olması gerekmektedir. Verimlilik ve performans gibi değerlerin ölçülmesi için birçok farklı değerlendirme metriği kullanılmaktadır. Tez çalışmasının bu kısmında hata matrisi tanımlarıyla birlikte verimlilik ve performans ölçümlerinde kullanılan kesinlik, duyarlılık, doğruluk, F1 score vb. değerlendirme metriklerine değinilmiştir.

4.1.5 Hata Matrisi

Bilgisayarlı görü uygulamalarında nesne tespiti yapılırken değerlendirmeler için ölçek olarak hata matrisleri kullanılmaktadır. Hata matrisleri için tahmin edilen ve gerçek sınıflarda geçerli olmak üzere dört durum söz konusu olmaktadır. Çizelge 4.3'te belirtilen bu durumlardan doğru pozitif (true positive) ve doğru negatif (true negative) ölçekleri başarı oranını arttırmaktadır. Çilek olgunluğu ölçümlerinde olgun (ripe) ve olgun değil (un ripe) olmak üzere iki temel değerlendirme söz konusu olmaktadır. Bu iki temel sınıflandırma için doğru pozitif durumu olgun olan çilek meyvesini kullanılan

modelin de spam olarak doğru tahmin etmesidir. Doğru negatif durumu ise modelin olgun olmayan çilek meyvesini yine gerçeğe uygun olarak olgun değil şeklinde tahmin etmesidir. Bununla birlikte yanlış pozitif (false positive) durumu ise olgun olamayan (un ripe) olarak etiketlenmiş çilek meyvesini modelin olgun (ripe) olarak tahmin etmesi, son durum olan yanlış negatif (false negatif) durumu ise olgun olarak etiketlenmiş çilek meyvesini modelin olgun değil (un ripe) olarak tahmin etmesi olarak tanımlanmaktadır.

Nesne algılama çalışmalarında algılanan nesneler sınırlayıcı kutu (bounding box) içerisinde gösterildiği için hata matrisinin değerlendirme matriği olarak yeterli olmadığı durumlar olabilmektedir. Yeterli eğitimlerin yapılması, itrasron sayısının arttırılması ile modelin başarısı istenen düzeye ulaşabilmektedir.

Çizelge 4.3 Hata matrisi

HATA MATRİS DEĞERLERİ		**Tahmin Edilen Sınıf**	
		(+)	**(-)**
		True Positive	**False Negative**
	(+)	Gerçek değerin ve tahmin edilen değerin 1 olduğu sınıf (çilek olgun/tahmin olgun)	Gerçek değerin 1 ve tahmin edilen değerin 0 olduğu sınıf (çilek olgun/tahmin olgun değil)
Gerçek Sınıf		**False Positive**	**True Negative**
	(-)	Gerçek değerin 0 ve tahmin edilen değerin 1 olduğu sınıf (çilek olgun değil/tahmin olgun)	Gerçek değerin ve tahmin edilen değerin 0 olduğu sınıf (çilek olgun değil/tahmin olgun değil)

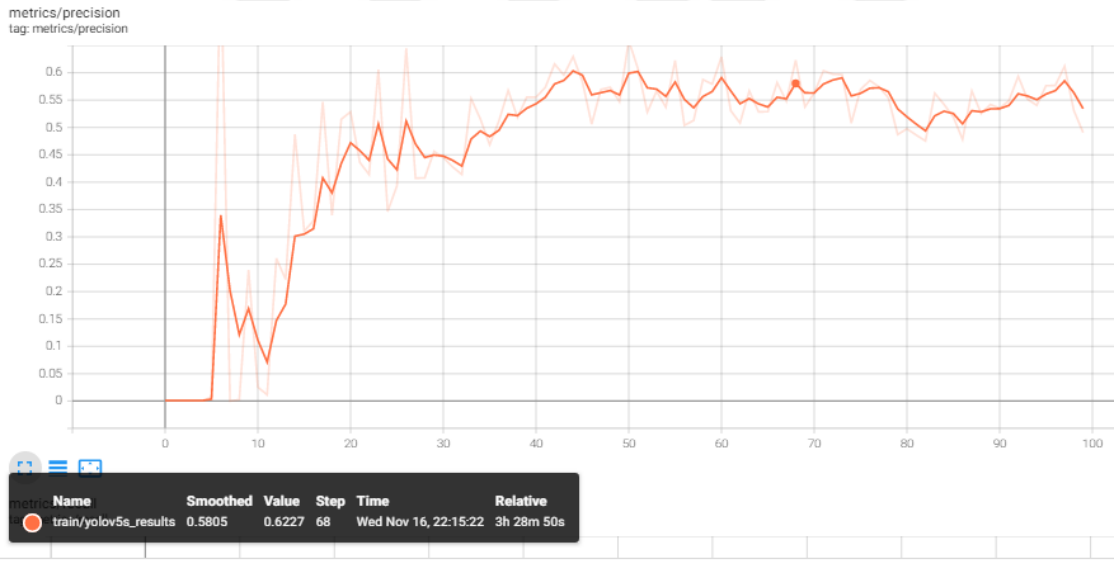
4.1.6 Kesinlik (Precision)

Kesinlik (precision), hesaplanan değer sonuç olarak pozitif sınıf olarak etiketlenen ve pozitif sınıf olarak tahmin edilen değerlerin oranıdır.

Eşitlik 4.1'de kesinlik değerinin nasıl hesaplandığı formülüze edilmiştir. TP: True Positive, FP: False Positive

$$Kesinlik\ (Prcision) = \frac{TP}{(TP + FP)} \tag{4.1}$$

Şekil 4.15'te YOLOv5 modelinin çilek veri seti kullanılarak elde edilen kesinlik grafiği 100 epoksluk eğitim sonucu verilmektedir. Kesinlik ve doğruluk değerleri doğru orantılı olduğu için kesinlik değerindeki artışlar doğrudan doğruluk değerini de arttırmaktadır.



Şekil 4.15 YOLOv5 modelinin kesinlik grafiği

Çizelge 4.4 Modellerin Kesinlik (Precision) karşılaştırmaları

$$\text{Kes} = \frac{\text{DP}}{\text{DP} + \text{YP}}$$

Algoritma	T.P	F.P	Kesinlik (Precision)
Yolo V5 Step 44 Süre 2’ 15”	0.5344	0.8491	0.3835
Faster CNN	0.9261	0.9732	0.487600695
SSDMobilNet			0.227

Çizelge 4.4 incelendiğinde YOLO ve Faster CNN algoritmalarında T.P ve F.P değerlerine bağlı olarak kesinlik (Precision) değerleri hesaplanarak, SSDMobilNet’te ise eğitim sonucunda üretilen kesinlik değerleri tabloda karşılaştırmalı olarak verilmiştir.

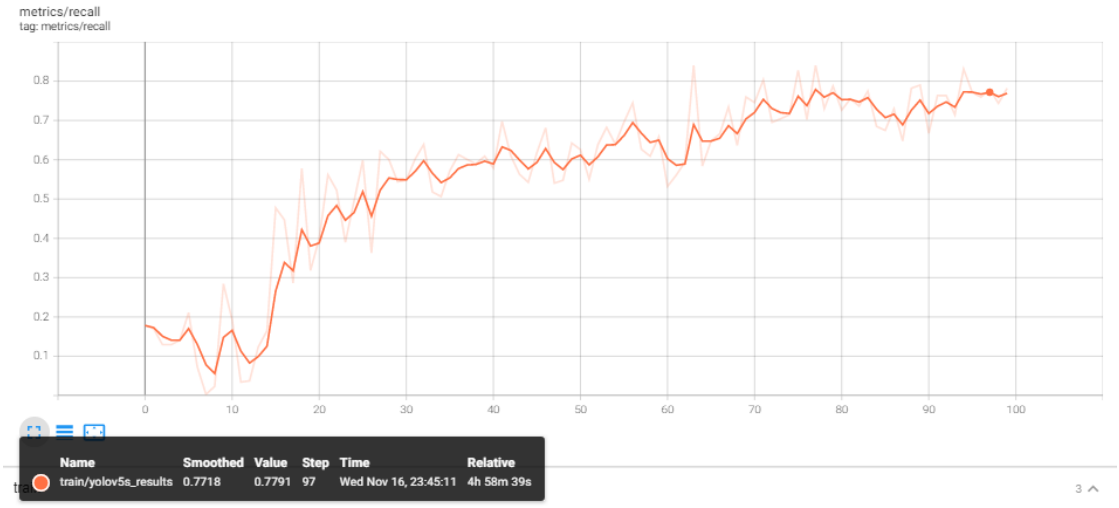
4.1.7 Duyarlılık (Recall)

Duyarlılık (recall), Gerçek değeri pozitif olarak etiketlenmiş nesnelerin ne kadar doğru tahmin edildiğini gösteren sonuç olarak hesaplanmaktadır.

Eşitlik 4.2’de duyarlılık değerinin doğru pozitif ve yanlış negatif değerlerine göre hesaplanma formülü gösterilmiştir. TP: True Positive, FN: False Negative

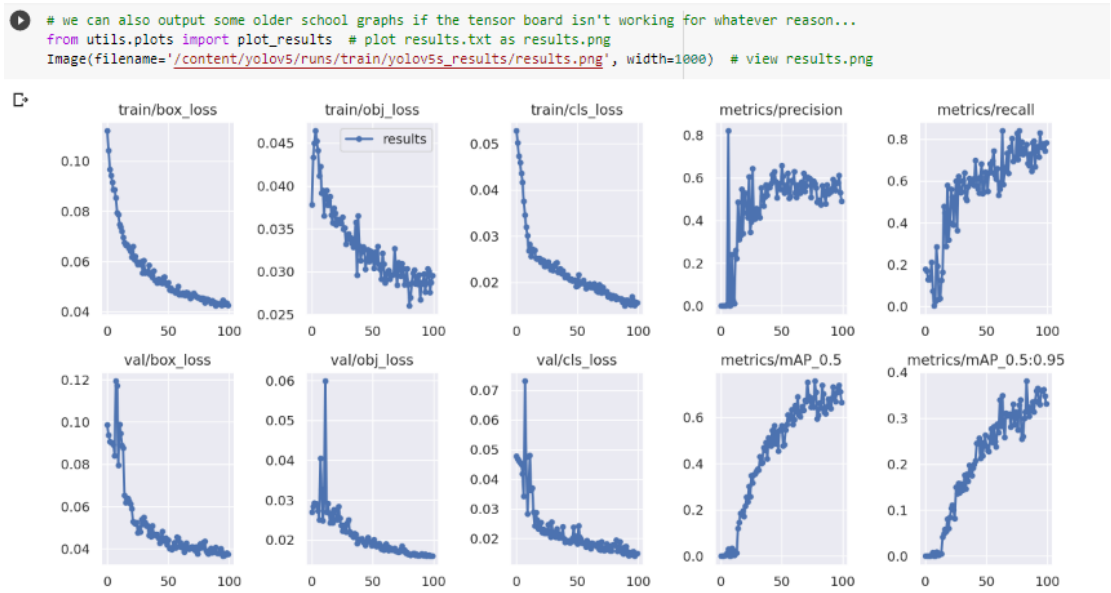
$$Duyarlılık\ (Recall) = \frac{TP}{(TP + FN)} \tag{4.2}$$

Şekil 4.16’daki grafik YOLOv5 modelinin çilek veri setine ait 100 iterasyonluk eğitim süreci duyarlılık sonucunu göstermektedir. Duyarlılık hesaplanırken duyarlılık ve doğruluk doğru orantılı olduğu için pozitif sonuçların oranı duyarlılık sonucunu doğrudan etkilemektedir.



Şekil 4.16 YOLOv5 modelinin duyarlılık grafiği

Şekil 4.17'de YOLOv5 modelinin çilek veri seti kullanılarak yapılan 100 iterasyonluk eğitim sürecinde elde edilen kesinlik, duyarlılık ve loss (kayıp veri) grafiği toplu olarak gösterilmektedir. Kesinlik ve duyarlılık ölçütlerinin pozitif sonuçları kullanarak doğru sonuca ulaşma oranını etkileyen değerler olduğu için artış göstermeleri doğruluk oranının da arttığını gösterdiği gibi, kayıp veri oranının şekilde gösterildiği gibi negatif artış göstermesi de algoritmanın doğruya yakın çalıştığını göstermektedir.



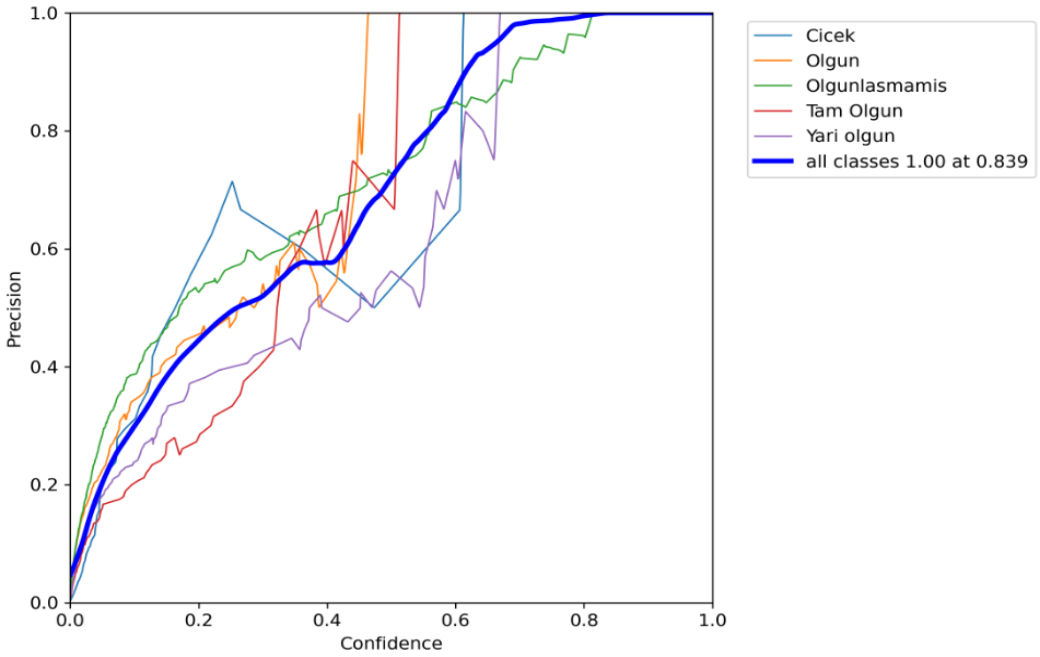
Şekil 4.17 Yolov5 modelinin kesinlik, duyarlılık ve kayıp veri grafiği

4.1.8 F Score:

F score modelin gücü, Precision (Kesinlik) ile Recall (Duyarlılık) değerlerinin harmonik ortalamasıdır. Bu sebeple başarının iyi bir göstergesidir. Şekil 4.18'de Yolov5 modelinin çilek veri seti ile elde edilen F Score eğrisi gösterilmektedir. Eşitlik 4.3'te ise F score değerinin hesaplanma formülü gösterilmiştir.

$$FScore = 2x\frac{Recall\ x\ Precision}{(Recall + Precision)} \tag{4.3}$$

Şekil 4.18'de çilek veri seti ile yapılan 100 iterasyon sonucu gösterilmiştir. Eğitim modeli olarak YOLOv5'in kullanıldığı süreçte kesinlik-duyarlılık değer korelasyonu grafiği gösterilmektedir.



Şekil 4.18 YOLOv5 modelinin 100 epok eğitiminin 99. Adımında F Score eğrisi

Kesinlik (Precision)-Duyarlılık (Confidence) grafiğinde kesinlik ve duyarlılık birbiriyle doğru orantılı olduğu için kesinlik değerindeki her artış doğruluk sonucunda doğrudan etkilemektedir. Çilek meyvesi eğitim sonuçlarına bakıldığında en yüksek kesinlik/duyarlılık değerine sahip nesne çiçek sınıfı, en başarısız nesne grubu ise yarı

olgun çilek olmuştur. Bu değerler kesinlik-doğruluk açısından bütün sınıflar için gösterilmektedir.

Kesinlik değeri durallık değeriyle doğru orantılı olduğundan dolayısıyla kesinlik değerinin duyarlılık değerini doğrudan etkilediğinden bahsedilmişti. Kullanılan modellerden biri olan FasterCNN modeli incelendiğinde kesinlik/duyarlılık başarı değerinde etiket ve tahmin değeri karşılaştırmasında eğitim sonucunda çiçek sınıfı en yüksek değeri alırken, tam olgun çilek ve yarı olgun çilek sınıflarının en başarısız sonucu aldığı görülmektedir.

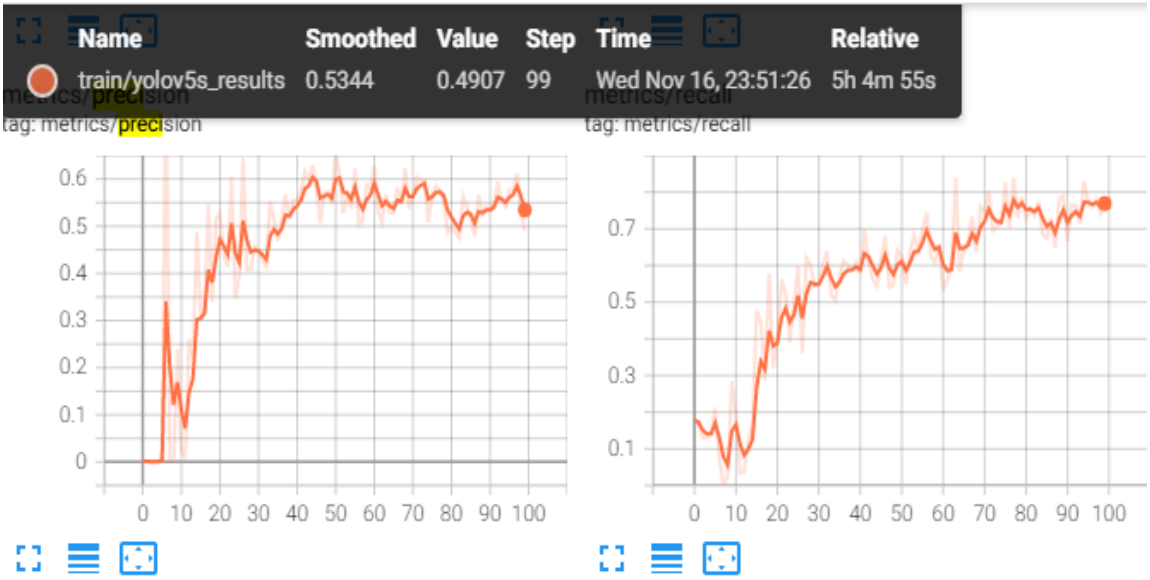
Çizelge 4.5 Modellerin F1 skor karşılaştırmaları

$$\text{F1 Skor } = 2 * \frac{\text{Duy} \times \text{Kes}}{\text{Duy} + \text{Kes}}$$

Algoritma	**Precision**	**Recall**	**F1 Score**
Yolo V5 (99.step)	0.5344	0.768	0.630250614
Faster CNN	0.312	0.380	0.3426589537
SSD MobilNet	0.227	0.299	0.2580722433

Modellere ait F1 Skor değerleri karşılaştırmaları incelendiğinde en yüksek değerin 0.6302 ile YOLO modeline, en düşük değerin ise 0.2580 ile SSD MobilNet modeline ait olduğu görülmektedir (Çizelge 4.5).

Bir modelin tercih edilmesi için en temel kriterlerden ikisi modelin tahmin yüdesindeki kesinlik ve duyarlılık değerleri olmaktadır. Modellere ait eğitim süreçleri ilerledikçe artan iterasyonlara bağlı olarak kesinlik ve duyarlılık değerlerindeki değişimler aşağıda gösterilmektedir (Şekil 4.19).



Şekil 4.19 Yolov5 99. İterasyonda kesinlik ve duyarlılık değerleri

```
DONE (t=0.00s)
creating index...
index created!
Running per image evaluation...
Evaluate annotation type *bbox*
DONE (t=0.02s).
Accumulating evaluation results...
DONE (t=0.02s).
 Average Precision  (AP) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.312
 Average Precision  (AP) @[ IoU=0.50      | area=   all | maxDets=100 ] = 0.494
 Average Precision  (AP) @[ IoU=0.75      | area=   all | maxDets=100 ] = 0.418
 Average Precision  (AP) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = -1.000
 Average Precision  (AP) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.403
 Average Precision  (AP) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.227
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=  1 ] = 0.286
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets= 10 ] = 0.380
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.380
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = -1.000
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.470
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.299
[11/19 14:57:31 d2.evaluation.coco_evaluation]: Evaluation results for bbox:
|   AP   |  AP50  |  AP75  |  APs  |  APm   |  APl   |
|:------:|:------:|:------:|:-----:|:------:|:------:|
| 31.183 | 49.407 | 41.795 |  nan  | 40.307 | 22.745 |
[11/19 14:57:31 d2.evaluation.coco_evaluation]: Note that some metrics cannot be co
[11/19 14:57:31 d2.evaluation.coco_evaluation]: Per-category bbox AP:
| category        | AP     | category    | AP     | category    | AP     |
|:----------------|:-------|:------------|:-------|:------------|:-------|
| cilek-Olgunlugu | nan    | Cicek       | 70.000 | Olgun       | 45.378 |
```

Şekil 4.20 Faster CNN kesinlik – duyarlılık değerleri

4.1.9 Doğruluk (Accuracy)

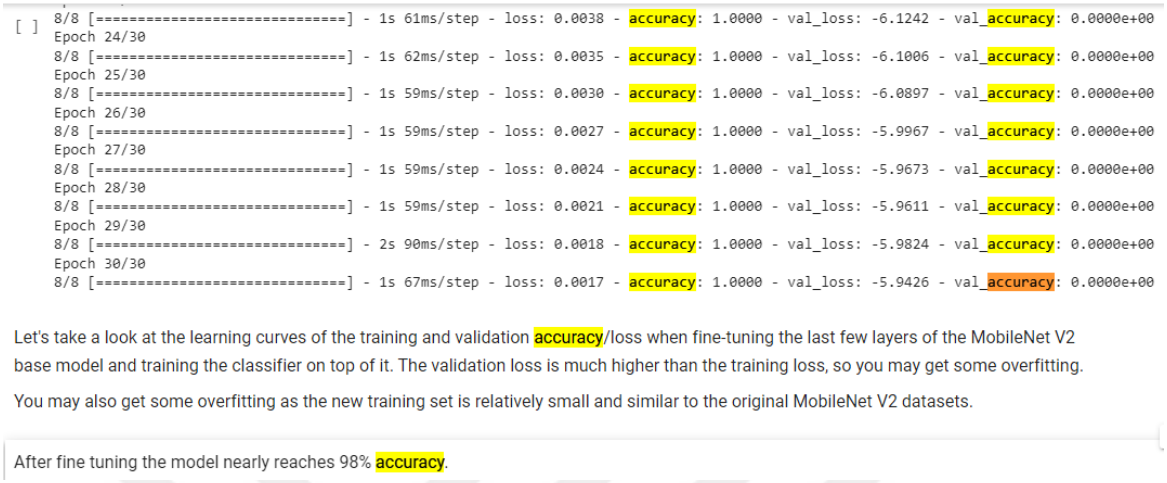
Doğruluk (Accuracy) değerlerinin belirlenebilmesi için doğru tahmin değerlerinin ve toplam tahmin değerlerinin bilinmesi gerekmektedir. Eşitlik 4.4'te belirtildiği gibi doğru tahmin değerlerinin toplam tahmin değerlerine oranı bize doğruluk değerini vermektedir.

$$\boldsymbol{Doğruluk\ (Accuracy)} = \frac{\textbf{Doğru Tahminler}}{\textbf{Toplam Tahminler}} \tag{4.4}$$

Şekil 4.21 SSD MobilNet eğitim ve doğrulama doğruluk grafiği

MobileNet V2 temel modelinin son birkaç katmanında ince ayar yaparken ve bunun üzerine sınıflandırıcıyı eğitirken eğitim ve doğrulama doğruluğun/eğitim kaybının öğrenme eğrilerine bir göz atıldığında doğrulama kaybının eğitim kaybından çok daha yüksek olduğu görülmektedir (Şekil 4.21).

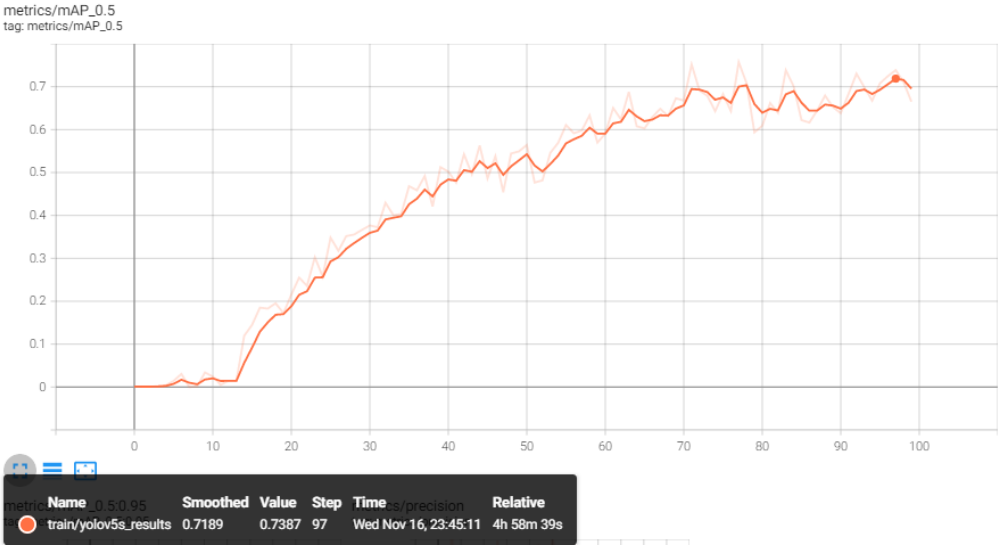
Algoritma verileri eğittikten sonra model neredeyse %98 doğruluğa ulaşmaktadır (Şekil 4.22).



Şekil 4.22 SSD MobilNet %98 ortalama doğruluk değeri

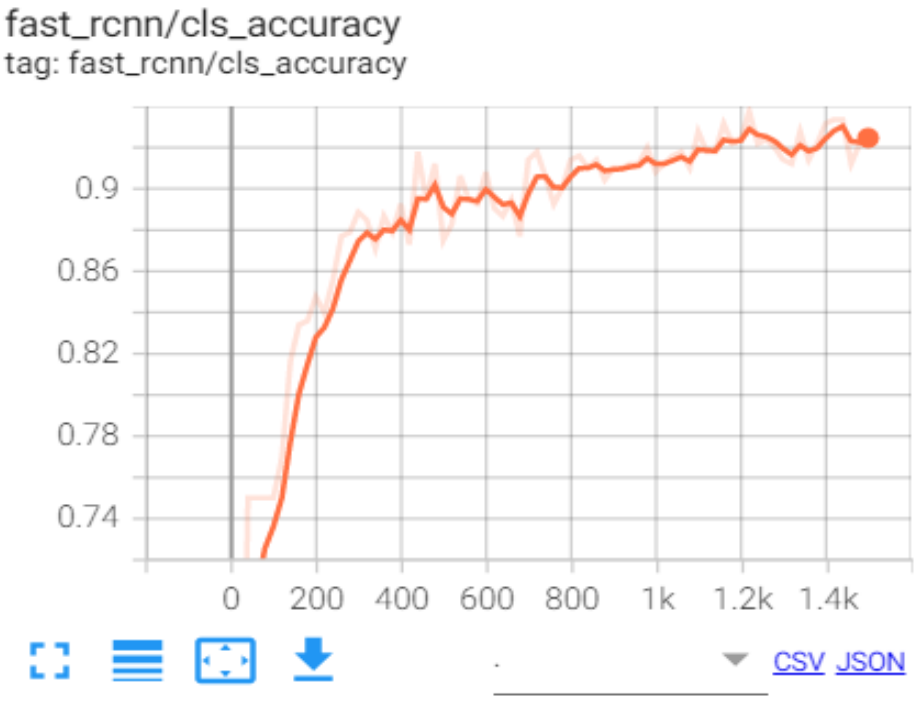
Çilek veri setinin performans sonuçlarından anlaşıldığı kadarıyla YOLOv5 modeli ile en iyi doğruluk oranına ulaşması amacıyla epok sayısı 100 olacak şekilde eğitim süreci başlatılmıştır. Eğitim süreci yaklaşık olarak 5 saat 8 dakika sürmüş ve başarı olarak 0.7189 ortalama doğruluk oranı elde edilmiştir. Bu sonuçlara göre model eğitim ve doğrulama veri setinde bulunan görüntülerdeki nesneleri yaklaşık %72 oranında sınıflandırma ve yerelleştirme yapabilmiştir. Şekil 4.23'te veri setinin eğitim sonucunda aldığı ortalama doğruluk oranı grafiği gösterilmektedir.

```
Results saved to runs/train/yolov5s_results
CPU times: user 2min 50s, sys: 26.1 s, total: 3min 16s
Wall time: 5h 8min 47s
```

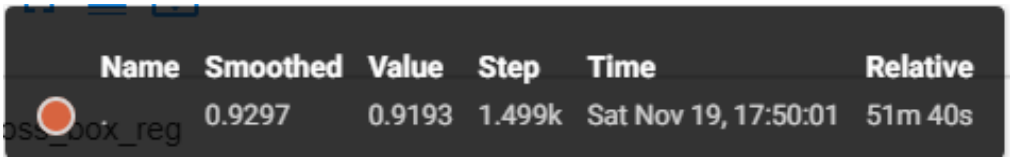


Şekil 4.23 Yolov5 Doğruluk Oranının 100 adım sonucu gösterimi

Faster CNN grafiği (Şekil 4.24) incelendiğinde doğruluk oranının 1'e çok yakın, ortalama 0.98 olduğu görülmektedir. Daha önce belirtildiği gibi doğruluk oranı 1'e yaklaştıkça algoritmanın doğru tahmin oranı artmıştır denebilmektedir.



Şekil 4.24 Faster CNN doğruluk grafiği



Şekil 4.25 Faster CNN doğruluk değerleri

Yapılan hesaplamalar sonucu modellerin doğruluk oranları karşılaştırıldığında kullandığımız veri seti ve iterasyon sayılarına bağlı olarak en yüksek doğruluk değerinin 0.98 ile SSDMobilNet'e ait olduğu görülmektedir (Çizelge 4.5).

Çizelge 4.5 Kullanılan 3 modelin doğruluk karşılaştırması

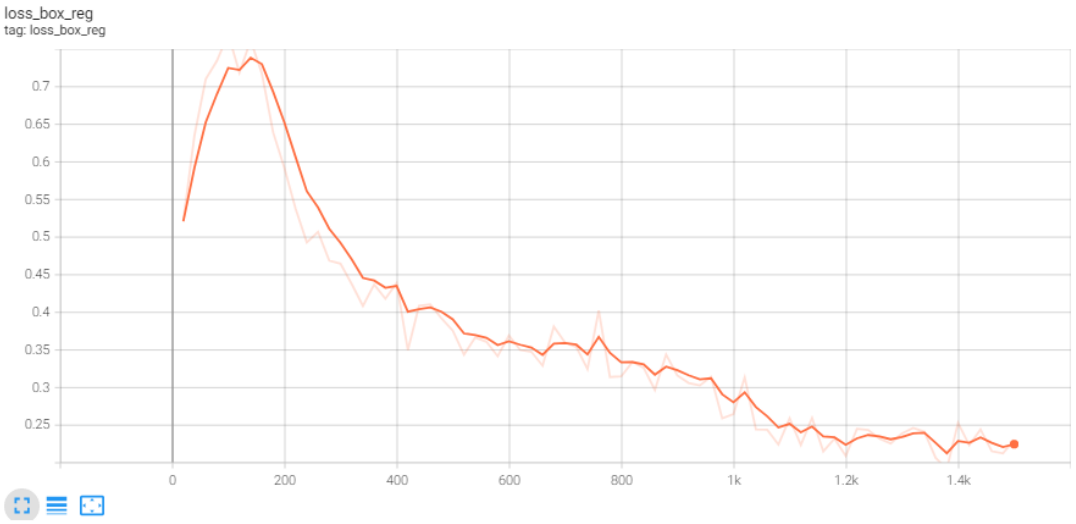
Algoritma	**Doğru Tahminler**	**Toplam Tahminler**	**Doğruluk**
Yolo V5 (100.step)	0.6955	0.967450271	0.7189
Faster CNN(1.4K)	0.9246	1.0043	0.9297
SSD MobilNet	0.98	1.000	0.98

4.1.10 Loss (kayıp veri) Grafik Karşılaştırmaları

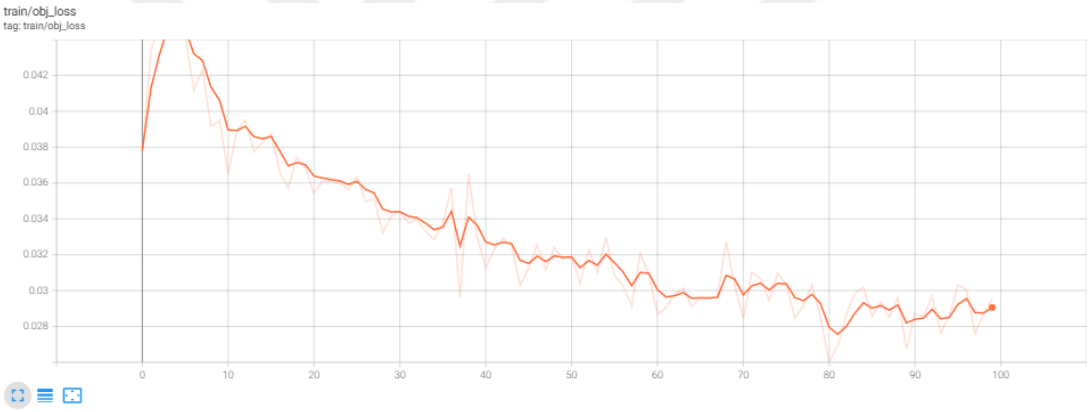
Eğitim aşamasında kontrol edilmesi gereken en önemli parametrelerden biride Loss (Kayıp Veri) grafiklerinin incelenmesidir. Bu aşamada tensorflow tarafından desteklenen tensorboard aracılığıyla grafikler oluşturulmaktadır.

Eğitimi yapılan dört farklı modelin loss grafikleri aşağıdaki şekillerde sunulmuştur.

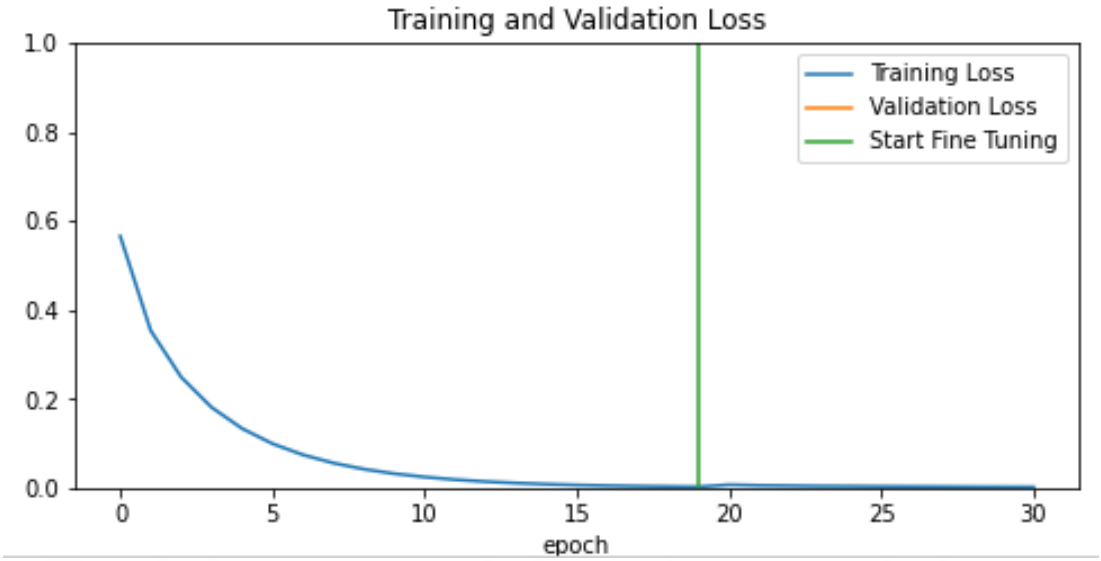
Loss grafikleri incelendiğinde Yolov5, FasterCNN ve SSD MobilNet loss değerlerinin 0.05 değerinin altına kadar düştüğü görülmektedir (Şekil 4.26, 4.27, 4.28). Bu değer 0'a ne kadar yakın olursa eğitimin o kadar başarılı gerçekleştiği anlamına gelmektedir. Bu başarıda eğitim adım sayısı ve öğrenme oranı önemli rol oynamaktadır. SSD MobilNet V2 modelindeki başarı oranının diğer modellere göre daha fazla olduğu tespit edilmiştir.



Şekil 4.26 FasterCNN loss garafiği



Şekil 4.27 YOLOv5 modelinin loss grafiği



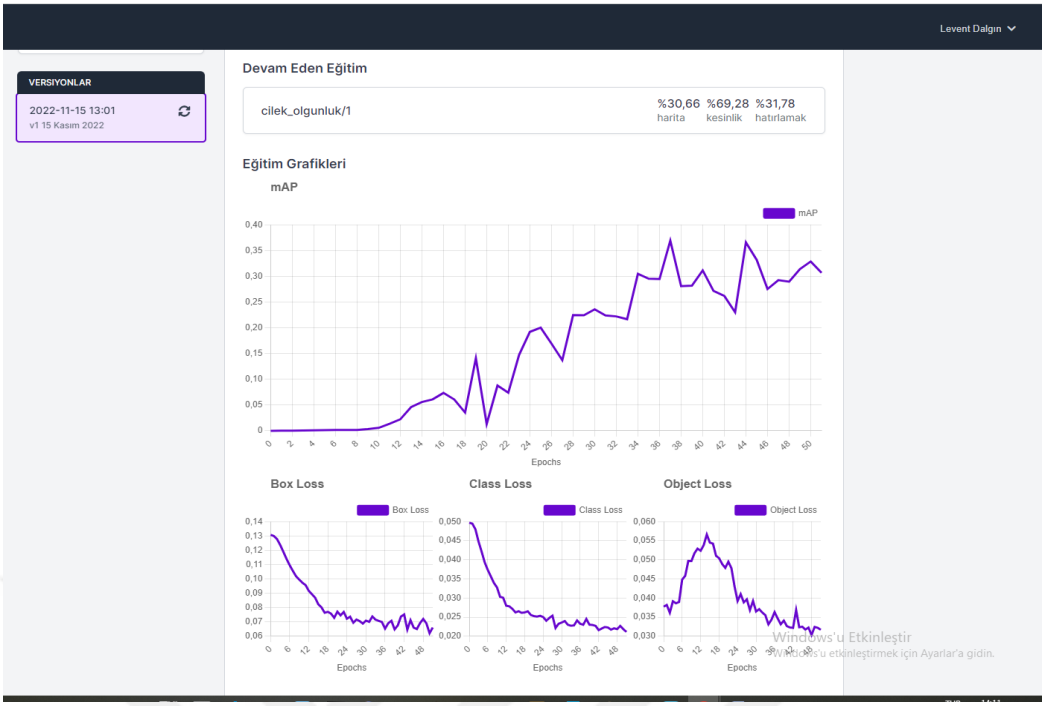
Şekil 4.28 SSD MobilNet loss grafiği

Eğitimler sonrası kayıp veri (Loss) oranının düşük olması (Çizelge 4.6) modelin sağlıklı çalıştığını gösteren diğer önemli krtiterlerden birisi olarak kullanılmaktadır.

Çizelge4.6 Modellerin kayıp veri karşılaştırması

Algoritma	**Doğru Tahminler**	**Toplam Tahminler**	**Loss (Kayıp Veri)**
Yolo V5 (100.step)	0.01563	0.941566265	0.166
Faster CNN(1.4K İterasyon) (Total Loss)	0.2247	0.532464455	0.422
SSD MobilNet	1.0000	0.9983302885	0.0017

Eğitimlere ayrılan süre arttıkça iterasyon sayılarındaki artışa bağlı olarak kayıp veri (Loss) değerlerindeki azalmalar aşağıda gösterilmektedir (Şekil 4.29).



Şekil 4.29 Eğitimde epok sayısı artışına bağlı olarak loss değerindeki değişimler

Yine iterasyon sayısındaki artışlara bağlı olarak doğruluk ve kesinlik değerlerindeki artışlar şekilde gösterilmektedir (Şekil 4.30).

Şekil 4.30 Eğitimde epok sayısı artışı ile precision ve recall değişimleri

Iterasyon süreci içerisinde değişimlerin sınırlayıcı kutu, sınıf ve nesnede nasıl gerçekleştiğini belirten değerler (Şekil 4.31) 288. iterasyon baz alınarak gösterilmektedir.

Şekil 4.31 Eğitimde 288 iterasyon sonucu kutu, sınıf ve nesnede loss değişimleri

4.1.11 Tüm Modellerin Çalışma Sonuçlarının Karşılaştırılması

Modellerin her birine ait hesaplanan kesinlik, doğruluk, kayıp veri, F-score değerlerinin toplam karşılaştırmalı grafiği Çizelge 4.7'de verilerek hangi modelin aranan hangi kriterde nasıl sonuçlar ürettiği gösterilmektedir. Bu tabloda hesaplamalar sonucu en yüksek kesinlik değerinin Yolov5'e en yüksek doğruluk oranının ise SSDMobilNet'e ait olduğu görülmektedir.

Çizelge 4.7 Modellerin kesinlik, F score, Doğruluk ve Kayıp Veri karşılaştırmaları

Model Adı	**Kesinlik**	**F Score**	**Doğruluk**	**Loss (Kayıp Veri)**
Yolo V5 (100.step)	0.6293	0.820009207	0.7189	0.166
Faster CNN(1.4K İterasyon)	0.487600695	0.901734104	0.9297	0.422
SSD MobilNet	0.227	-	0.98	0.0017

Çizelge 4.8 Modellerin boyut, sınıf, iterasyon, süre, doğruluk karşılaştırmaları

Model Adı	**Görüntü Boyutu**	**Sınıf / Nesne**	**Epok(İterasyon) Sayısı**	**Eğitim Süresi**	**Ortalama Doğruluk Oranı**
Yolo V5 (100.step)	640x640	5/540	100	5saat 8dk 47 sn	%72
Faster CNN(1.4K İterasyon)	640x640	5/540	1420	0saat 52dk 14sn	%93
SSD MobilNet	640x640	5/540	100	3saat 17dk 0sn	%98

Modellerin doğruluk değerleri karşılaştırıldığında (Çizelge 4.8) %98 ile SSD MobilNet 1. Sırada, %93 doğruluk oranıyla Faster CNN ikinci sırada ve %72 doğruluk oranıyla Yolov5 modelinin 3. Sırada olduğu görülmektedir.

3.8 Ek Parametreler

Çilek veri seti kullanılarak farklı evrişimsel sinir ağı modelleri deneysel çalışmaya tabi tutulmuş ve elde edilen sonuçlar verimlilik, hız ve doğruluk olarak karşılaştırılmıştır. Kullanılan evrişimsel sinir ağı modelleri önceki bölümlerde ayrıntılı olarak anlatılmıştır. Ayrıca elde edilen sonuçlar tablo, grafik ve diyagram olarak detaylı olarak anlatılmış ve elde edilen sonuçlar sayısal olarak listelenmiştir.

Tüm bunlarla birlikte evrişimsel sinir ağlarında bir ağın eğitimi esnasında modelin çalışmasını etkileyen bazı ek parametreler vardır. Bu parametrelerinde doğru bir şekilde okunması gerekir. Bu parametreler veri setine ve kullanılan modele göre değişiklik göstermektedirler. Çizelge 4.9'da uygulama süresince kullanılan modellere ait eğitilebilir, eğitilemez ve toplam parametre değerleri gösterilmektedir.

Çizelge 4.9 Farklı parametrelerin kullanılan modellerdeki sonuçları

Farklı Parametrler	**YOLOv5**	**FasterCNN**	**SSDMobilNet**
Total params:	7265882	1024	2,259,265
Trainable params:	7265882	704	1,862,721
Non-trainable params:	0	320	396,544
Percent Value	%100	%68,75	%82,42
Toplam	7265882	1024	2,259,265

Çizelge 4.9 incelendiğinde veri setlerinin eğitilme performansları karşılaştırılmasında %100'lük verimle Yolov5 en iyi sonucu verirken, %82'lik oranla SSDMobilNet ikinci sırada ve ortalama %68'lik oranla FasterCNN modeli üçüncü sırada yer almaktadır.

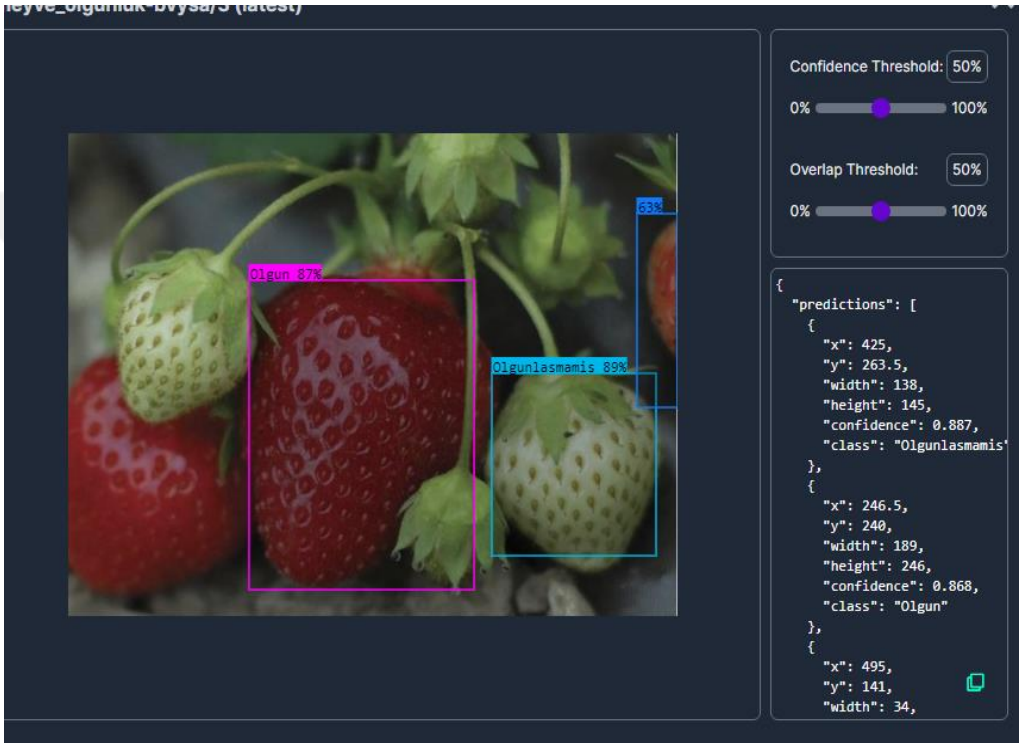
Modellerin veri seti eğitimleri bittikten sonra sağlıklı çalıştığını ıspat eden sonuçlar şekillerde gösterilmektedir. Yolov5 modeline ait sonuçlar Şekil 4.32'de gösterilmektedir.

Şekil 4.32 Yolov5 Algoritması çilek veri seti eğitim sonuçları

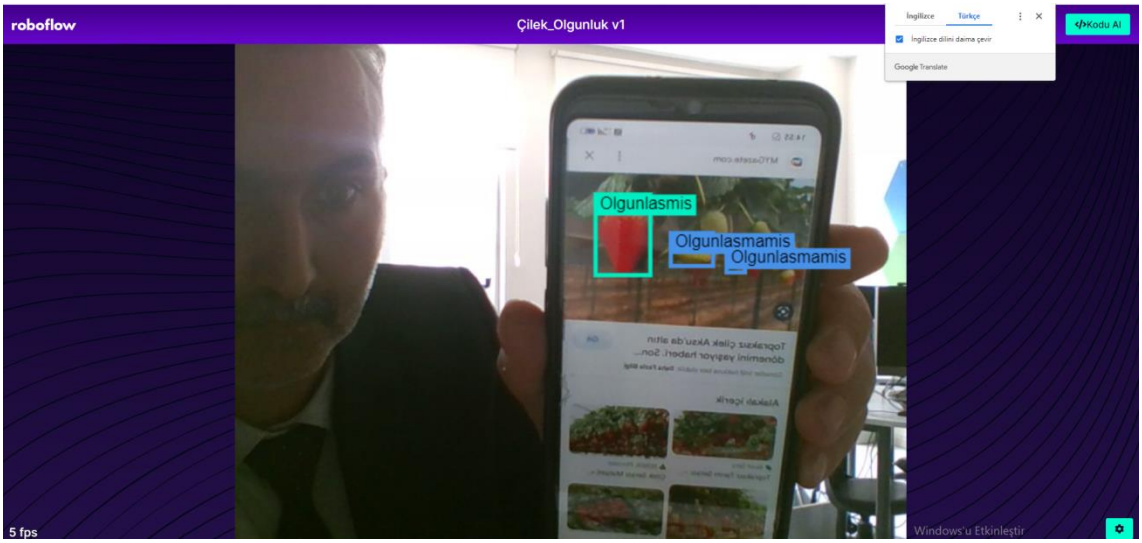
Yine karşılaştırması yapılan modellerden Faster CNN modeline ait sonuçlar şekil 4.33'te gösterilmektedir.

Şekil 4.33 Faster CNN v2 Algoritması çilek veri seti eğitim sonuçları

Tüm eğitimler tamamlandıktan sonra veri setinden alınan veya internetten gelişigüzel seçilen onlarca resim üzerinde yapılan denemelerde modellerin sağlıklı sonuçlar ürettiği görülmektedir. Şekil 4.34 incelendiğinde veri setinde kullanılmayan, internetten gelişigüzel alınan bir görsele ait reel time sonuçlar görülemektedir. Bu görselde resme ait olgunluk değerleriyle birlikte sınırlayıcı kutulara ait koordinat değerleride görülmektedir.



Şekil 4.34 Eğitim sonrası ham resimlerin anlık test edilmesi



Şekil 4.35 İnternetten rastgele seçilen resimlerin real time olarak test edilmesi

Eğitimler bittikten sonra dışarıdan alınan herhangi bir çilek meyvesi demeti üzerinden sonuçlar test edildiğinde şekil 4.35'te görüldüğü gibi çilek meyvesi demetinde bulunan olgunlaşmış ve olgunlaşmamış çileklerin sınıflandırması, tahmin yüzdesiyle birlikte anlık olarak elde edilebilmektedir. Buda eğitimin doğru bir şekilde sonuçlandığını, yapılan deneysel işlemin sağlıklı çalıştığını göstermektedir.

5. TARTIŞMA VE SONUÇ

Çilek meyvesine ait gelişim dönemlerinin farklı evrişimsel görüntü işleme modelleriyle karşılaştırmalı performans ve verimlilik çalışmaları yapılmıştır. Tez çalışması boyunca her modele ait farklı parametre değerleri karşılaştırılmıştır.

Tez çalışması başladığı andan itibaren hissedilen en büyük eksiklik literatürde benzer çalışmaların eksikliğidir. Dünyada çok geniş yetiştirme alanı bulunan ve dört mevsim verim elde edilen çilek meyvesine ait bu tür bir çalışmanın ilk olması literatüre katkı sunması açısından değerlidir.

Çalışma boyunca anlaşmalı olunan seradan ve internetten elde edilen görüntüler Label İmage, Tensorflow, roboflow vb. farklı etiket programlarıyla etiketlenmiş, bunlardan hız ve verimlilik için en uygun olanın roboflow tespit edilerek eğitim çevrimiçi olarak roboflow üzerinden yapılmıştır. Görüntülerin roboflow ile etiketlenme işlemi bitip, eğitim süreçleri farklı modeller tarafından defalarca test edilmiştir. Saatler süren iterasyon sonuçlarına bakılarak Yolov5, Faster CNN ve SSD MobilNet modelleri kesinlik, doğruluk, Fscrore ve Loss (kayıp veri) kriterlerine göre karşılaştırılabilmiştir.

Sonuçlara bakıldığında kesinlik için %63 ile Yolov5, Fscore değeri için %90 ile FasterCNN, doğruluk için %98 ile SSD MobilNet modellerinin daha iyi sonuçlar verdiği görülmüştür. Verilerin eğitim sırasındaki verim oranı parametreleri karşılaştırıldığında ise iyi sonucu %100'lük bir verimle Yolov5 modeli göstermektedir.

Yakın zamanda yapılan benzer bilimsel araştırma sonuçlarına bakıldığında, 2021 yılında yapılan bir çalışmada %69 doğruluk oranıyla Yolov5'in hava fotoğrafçılığında en iyi sonucu verdiği (Murat, 2021).

Yine yakın zamanda yapılan bir çalışma sonucunda Yolo için girdi boyutu büyüdükçe, geliştirilen algoritma performansı ve doğruluk oranının düştüğü gözlenmiştir. Literatür üzerinde yapılan karşılaştırmada hız olarak en yüksek tepki değerine sahip olmasına rağmen, Yolo algoritması başarım oranı olarak Faster R-CNN(VGG-16) algoritmasının gerisinde kalmıştır. Tespit hızını arttırmanın doğruluk payını düşürdüğü gözlenmiştir

Yine trafikte bulunan araçların sınıflandırılması üzerine yapılan bir çalışmada Faster R-CNN ResNet50 modeli çıkan sonuçlara göre %94.24 lük oranla en başarılı sınıflandırma sonuçlarını vermiştir. SSD Mobilenet v2 modeli ise %88.32 lik doğruluk

oranıyla 4 farklı model arasında en düşük orana sahiptir. SSD Mobilenet v2 versiyonu doğruluk oranıyla sonuncu olsa da test süresindeki hızıyla dikkat çekmektedir (Tan, 2019)

Yapılan ilk iki çalışma değerlendirildiğinde hava fotoğrafçılı için ilkinde Yolo ikincisinde Faster CNN daya iyi sonuç vermiştir. Son çalışmada ise (Tan, 2019) Faster CNN modelinin trafikteki araçlar için daha iyi sonuçlar ürettiği görülmektedir.

Tez çalışması başladıktan sonra Çin Weifang üniversitesince yapılan benzer bir çalışmada ise An vd. (2022) yaptıkları çalışmada çilek meyvesine ait gelişim yüzdelerini YOLO modelinin farklı versiyonlarını karşılaştırarak yapmışlardır. Elde ettikleri deneysel Sonuçlar, YOLO modelinde hassasiyetinin, doğruluğunun ve geri çağırmasının farklı versiyonlarda sırasıyla %94.26, %93.15 ve %90.72 olduğunu hesaplamışlardır. Ancak yapılan çalışma sadece YOLO modelinin kendi içindeki versiyon karşılaştırmasıdır (An vd., 2022).

Bu çalışmada yukarıda verilen örneklerden farklı olarak Yolo, Faster CNN ve SSD MobilNet olmak üzere üç farklı model karşılaştırılmış ve veri seti için ortalama 1mt uzaklıktan elde edilen çilek görsellerinin olgunluk sınıfı karşılaştırmaları yapılmıştır. Çalışma bu yönleriyle değerlendirildiğinde tarımda yapay zeka ve hasat robotları için yardımcı kaynak ve benzer çalışmalar için hazır veri niteliği taşımaktadır.

Çalışmanın son kısmının ilk aşamasında çilek meyvesi gelişim oranlarını gösteren görsel sonuçlar ve ikinci aşamada etiketsiz ham veriler kullanılarak hızlı bir biçimde meyve olgunluk seviyelerini gösteren sonuçlar elde edilmiştir. Bu sonuçlara bakıldığında modellerin sağlıklı bir şekilde sonuç ürettiği görülmektedir.

KAYNAKLAR

Aksoy, B., Halis, H. D., Salman, O (2020*). Elma bitkisindeki hastalıkların yapay zekâ yöntemleri ile tespiti ve yapay zekâ yöntemlerinin performanslarının karşılaştırılması*. International Journal of Engineering and Innovative Research, Araştırma makalesi, Isparta Uygulamalı Bilimler Üniversitesi, Teknoloji Fakültesi, Mekatronik Mühendisliği Bölümü, Isparta, Türkiye.

Aladağ, Ç. H. (2009). *Yapay sinir ağlarının mimari seçimi için tabu arama algoritması* Doktora Tezi, Hacettepe Üniversitesi, Fen Bilimleri Enstitüsü, Ankara, Türkiye.

Albawi, S., Mohammed, T. A., Al-Zawi, S. (2017, August). Understanding of a convolutional neural network. International conference on engineering and technology (ICET) (pp. 1-6). Ieee. **2017 Uluslararası Mühendislik ve Teknoloji Konferansı (ICET),** Antalya, Türkiye.

Al-Masri, A. N., Ab Kadir, M. Z. A., Hizam, H., Mariun, N. (2015). *Simulation of an adaptive artificial neural network for power system security enhancement including control action*. Applied Soft Computing, 29, 1-11. Electronics Engineering, University Putra Malaysia, Malaysia.

Alruwaili, M., Abd El-Ghany, S., Shehab, A. (2019). *An enhanced plant disease classifier model based on deep learning techniques*. International Journal of Advanced Technology and Engineering Exploration Department of Computer Science, College of Science and Arts, Jouf University, Saudi Arabia.

Amidi, A., Amidi, S. (2020). *Recurrent neural networks*, Optimal Speed and Accuracy of Object Detection, MDAKM Group, Department of Computer Engineering and Informatics, University of Patras, Patras, Greece.

An, Q., Wang, K., Li, Z., Song, C., Tang, X., Song, J. (2022). *Real-Time Monitoring Method of Strawberry Fruit Growth State Based on YOLO Improved Model*. IEEE Access, School of Mechanical and Automation, Weifang University, Weifang 261061, China.

Anonim. (2018a) *Yapay sinir ağları nedir?* Erişim tarihi: 18.05.2021. Erişim adresi: http://kod5.org/yapay-sinir-aglari-ysa-nedir/

Anonim. (2018b). *The Perceptron* Erişim Tarihi: 23 Haziran 2020. Erişim adresi: https://towardsdatascience.com / the-perceptron-3af34c84838c

Anonim. (2020). *Etiketleme yöntemleri* Erişim Tarihi 12 Ocak 2020. Erişim adresi: https://github.com/puzzledqs/BBox-Label-Tool

Anonim. (2021a). *Reducing Loss*: Gradient Descent Erişim Tarihi: 02 Kasım 2022. Erişim adresi: https://developers.google.com/machine-learning/crash-course/reducing-loss/gradientdescent.

Anonim. (2021b). *Reducing Loss: Gradient Descent* Erişim tarihi: 02 Kasım 2020. Erişim adresi: https://developers.google.com/machine-learning/crash-course/reducing-loss/gradientdescent.

Anonim. (2021c). *Convolutional Neural Networks for Visual Recognition* Stanford University, Erişim tarihi: 5 Ocak 2021. Erişim adresi: https://cs231n.github.io/convolutional-networks/#pool.

Anonim. (2023a). *Yapay zekâ ve alt alanları hiyerarşisi gösterimi* Erişim tarihi: 31 Ocak 2023. Erişim adresi: https://www.datarobot.com/wiki/artificial-intelligence/

Anonim. (2023b). *İnsan Sinir Hücresi* Psikopataloji bilimi. Erişim tarihi: 3 Ocak 2022. Erişim adresi: https://www.psikopatolojibilimi.com/wp-content/uploads/2019/01/4.png 22

Anonim. (2023c). *Evrişimli sinir ağları*, Erişim tarihi: 7 Mart 2021. Erişim adresi: https://www.javatpoint.com/keras-convolutional-neural-network

Anonim. (2023d). *Aktivasyon Fonksiyonları*. Erişim tarihi: 3 Şubat 2023. Erişim adresi: http://buyukveri.firat.edu.tr/2018/04/17/derin-sinir-aglari-icin-aktivasyon-fonksiyonlari/

Anonim. (2023e). *Lineer Regraesyon nedir*, Erişim tarihi: 1 mayıs 2023. Erişim adresi: https://www.bilimma.com/lineer-regresyon-nedir/

Anonim. (2023f). *Derin Sinir Ağları İçin Aktivasyon Fonksiyonları*. Erişim tarihi: 26 Ocak 2022. Erişim adresi: http://buyukveri.firat.edu.tr/2018/04/17/derin-sinir-aglari-icin-aktivasyon-fonksiyonlari/

Arunava, A. (2018). *Convolutional Neural network*. In Proceedings of the International Conference On Applied Sciences And Engineering, Pattaya, Tayland.

Atalay, M., Çelik, E. (2017). Büyük veri analizinde yapay zekâ ve makine öğrenmesi uygulamalari-artificial intelligence and machine learning applications in big data analysis. Kırklareli Üniversitesi, Türkiye.

Baranwal, S., Khandelwal, S., Arora, A. (2019, February). Deep learning convolutional neural network for apple leaves disease detection. In Proceedings of international conference on sustainable computing in science, technology and management (SUSCOM), Amity University Rajasthan, Jaipur-India.

Bradley, P. S., Mangasarian, O. L. (1998, July). *Feature selection via concave minimization and support vector machines*. Computer Sciences Department University of Wisconsin Madison, United States of Amerika.

Brahimi, M., Arsenovic, M., Laraba, S., Sladojevic, S., Boukhalfa, K., Moussaoui, A. (2018). Deep learning for plant diseases: detection and saliency map visualisation. *Human and Machine Learning*: Visible, Explainable, Trustworthy and Transparent, Bilgisayar Bilimleri Bölümü, USTHB Üniversitesi, Cezayir.

Brownlee, J. (2021). *Difference Between a Batch and an Epoch in a Neural Network*. Erişim tarihi: 21 Mayıs 2022. Erişim Adresi: https://machinelearningmastery.com/difference-between-a-batch-and-an-epoch

Bulut, F., Amasyalı, M. F., Sönmez, A. C. (2014). *Uzman Kararlarının Yeni Bir Geçiş Fonksiyonuyla Birleştirmesi. Akıllı Sistemlerde Yenilikler ve Uygulamaları*. Bilgisayar Mühendisliği Bölümü Elektrik-Elektronik Fakültesi Yıldız Teknik Üniversitesi, Davutpaşa, İstanbul, Türkiye.

Cadzow, J. A. (2002). *Minimum ℓ1, ℓ2, and ℓ∞ norm approximate solutions to an overdetermined system of linear equations*. Digital Signal Processing, Elektrik ve Bilgisayar Mühendisliği Bölümü, Vanderbilt Üniversitesi, Nashville, Tennessee, ABD.

Canan, S. (2006). *Yapay sinir ağları ile GPS destekli navigasyon sistemi*, Doktora Tezi. Selçuk Üniversitesi, Fen Bilimleri Enstitüsü, ElektrikElektronik Mühendisliği, Konya.

Castelman, R. K. (1996). **Digital image processing**. Prentice hall, Englewood Cliffs, New Jersey, USA.

Chetoui, M., Akhloufi, M. A., Yousefi, B., Bouattane, E. M. (2021). Explainable COVID-19 detection on chest X-rays using an end-to-end deep convolutional neural network architecture. *Big Data and Cognitive Computing*, Basel, İsviçre.

Costa, C., Murphy, M. (2015). **Bourdieu, habitus and social research**: The art of application. Springer: Lecturer in Tecnology Enhanced Learninng, Universty of strathclyde, UK.

Çarkacı, N. (2018). *Derin Öğrenme Uygulamalarında En Sık kullanılan Hiper-parametreler* Erişim tarihi: 01. 12. 2022. Erişim adresi: https://medium.com/deep-learning-turkiye/derinogrenme-uygulamalarinda-en-sik-kullanilan-hiper-parametreler.

Dawson, J. (2007). The Essential Turing: Seminal Writings in Computing, Logic, Philosophy, Artificial Ingelligence, and Artificial Life plus The Secrets of Enigma, Clarendon Press, Oxford University Press, New York.

DeChant, C., Wiesner-Hanks, T., Chen, S., Stewart, E. L., Yosinski, J., Gore, M. A., Lipson, H. (2017**). *Automated identification of northern leaf blight-infected maize plants from field imagery using deep learning***. Phytopathology, Department of Computer Science, Columbia University, New York. Central China Normal University, China

Egrioglu, E., Aladag, C. H., Yolcu, U., Uslu, V. R., Basaran, M. A. (2009). A new approach based on artificial neural networks for high order multivariate fuzzy time series. Expert Systems with Applications, Department of Statistics, Ondokuz Mayis Universitesi, Samsun, Türkiye.

Eiben, A., Smit, SK (2011). *Parameter setting for configuring and analyzing evolutionary algorithms*. Herd and Evolutionary Computing, Deparment of Computer Science Doctorate, Vrije Universty, Amsterdam.

Everingham, M. Williams C. Winn J. Zisserman A. (2013). *The PASCAL Visual Object Classes* Erişim tarihi: 16 12 2020 Erişim adresi: http://host.robots.ox.ac.uk/pascal/VOC/

Fang, T., Chen, P., Zhang, J., Wang, B. (2019). Identification of apple leaf diseases based on convolutional neural network. **In Intelligent Computing Theories and Application: 15th International Conference, ICIC** 2019, Nanchang, China.

Ferentinos, K. P., Barda, M., Damer, D. (2019). An image-based deep learning model for cannabis diseases, nutrient deficiencies and pests identification. **In Progress in Artificial Intelligence: 19th EPIA Conference on Artificial Intelligence, EPIA**, Nanchang, China.

Fujita, E., Kawasaki, Y., Uga, H., Kagiwada, S., Iyatomi, H. (2016). Basic investigation on a robust and practical plant diagnostic system. **IEEE international conference on machine learning and applications,** Applied Informatics, Graduate School of Science and Engineering, Hosei University, Japan.

Geng, R., Ma, Y., Huang, W. (2021). An improved helmet detection method for YOLOv3 on an unbalanced dataset. **In 2021 3rd International Conference on Advances in Computer Technology**, Information Science and Communication, Uniiversty of Cambridge, UK.

Girshick, R., Donahue, J., Darrell, T., Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation**. In Proceedings of the IEEE conference on computer vision and pattern recognition**, Columbus, OH, ABD.

Golik, P., Doetsch, P., Ney, H. (2013). *Cross-entropy vs. squared error training*: a theoretical and experimental comparison. In Interspeech, Computer Science Department, RWTH Aachen University, Aachen, Germany.

Guo, Z., Zhang, X., Mu, H., Heng, W., Liu, Z., Wei, Y., Sun, J. (2020). Single path one-shot neural architecture search with uniform sampling. **In Computer Vision–ECCV: 16th European Conference**, Glasgow, UK.

Gurney, K. (1997). **An introduction to neural networks**. CRC press: Profesör,

Bilişim, Bilgi İşlem ve Siber Sistemler Okulu, Kuzey Arizona Üniversitesi Londra.

Güçlü, E., Aydın, İ., Şahbaz, K., Akın, E., Karaköse, M. (2021). *Demiryolu bağlantı elemanlarında bulunan kusurların YOLOv4 ve bulanık mantık kullanarak tespiti*. Fırat Üniversitesi, Mühendislik Fakültesi, Bilgisayar Bölümü, Elazığ, Türkiye.

Güney, E. (2021). Sürücü asistan sistemleri için mobil GPU tabanlı gerçek zamanlı durum analizi ve tespit uygulamaları, Sakarya Üniversitesi, Fen Bilimleri Enstitüsü, Erdivan, Sakarya.

Harman, G. (2021). *Destek Vektör Makineleri ve Naive Bayes Sınıflandırma Algoritmalarını Kullanarak Diabetes Mellitus Tahmini*. Avrupa Bilim ve Teknoloji Dergisi, (32), 7-13. Doi:10.31590/ejosat.1041186

Hazır, E. (2021) *Denetimli ve denetimsiz öğrenme*. Enstitüsü, Yüksek Lisans Tez, Adana. Erişim tarihi: 2 Şubat 2023 Erişim adresi: https://eneshazr.medium.com/supervised-unsupervised-learning-makine-%C3%B6%C4%9Frenmesi-b903bc09430e

İslamoğlu, E. (2015). *Aralık Değerli Zaman Serilerinde Kullanılan Modelleme Teknikleri*. Erzincan University Journal of Science and Technology, 8(2), 178-193. Doi: 10.18185/eufbed.04685

Kabalcı, E. (2014). *Yapay Sinir Ağları. Ders Notları* erişim tarihi: 02 mayıs 2022. Erişim adresi: https://ekblc.files.wordpress.com/2013/09/ysa.pdf

Karakaya, S. (2021). Türkiye'deki Illerin Göç Göstergelerinin Python Kullanılarak K-Ortalamalar Kümeleme Yöntemi Ile Araştırılması, Doctoral dissertation, Bursa Uludag University, Turkey.

Kavzoglu, T., Saka, M. H. (2005). Modelling local GPS/levelling geoid undulations using artificial neural networks. Journal of geodesy, 78, 520-527. Doi: 10.1007/s00190-004-0420-3

Keefe, P. D. (1992). *A dedicated wheat grain image analyser*. Plant varieties & seeds, Cambridge CB3 0LE, UK.

Khanday, O., Dadvandipour, S. (2020). *Convolutional neural networks and impact of filter sizes on image classification.* Multidiszciplináris Tudományok, 10(1), 55-60. Doi: 10.35925/j.multi.2020.1.7

Khitthuk, C., Srikaew, A., Attakitmongcol, K., Kumsawat, P. (2018). Plant leaf disease diagnosis from color imagery using co-occurrence matrix and artificial intelligence system. **International Electrical Engineering Congress** (iEECON) (pp. 1-4). IEEE. Krabi, Tayland.

Kızılboğa, A. Y. (2021). *Yapay sinir ağları ve derin öğrenme teknikleri kullanılarak elma ve ayvada çeşitli hastalıkların tespit edilmesi*, Master's thesis, Fen Bilimleri Enstitüsü/Bilgisayar Mühendisliği Ana Bilim Dalı. Gelişim Üniversitesi, İstanbul.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., Berg, A. C. (2016). Springer International Publishing. **European Conference, Amsterdam, The Netherlands**, Proceedings, Part I 14 (pp. 21-37). University of Michigan, Ann-Arbor, USA

Lu, Y., Yi, S., Zeng, N., Liu, Y., Zhang, Y. (2017*). Identification of rice diseases using deep convolutional neural networks*. Neurocomputing, 267, 378-384. Doi: 10.1016/j.neucom.2017.06.023

Maltarollo, V. G., Honório, K. M, Silva, A. (2013). *Applications of artificial neural networks in chemical problems*. Artificial neural networks-architectures and

applications, 203-223. Doi: 10.57772/51275
Mohanty, S. P., Hughes, D. P., Salathé, M. (2016). *Using deep learning for image-based plant disease detection*. Frontiers in plant science, 7, 1419. Doi: 10.3389/fpls.2016.01419
Murat, S. (2021). *İnsansız hava aracı görüntülerinden derin öğrenme yöntemleriyle nesne tanıma*, Maltepe Üniversitesi, Lisansüstü Eğitim Enstitüsü, İstanbul.
Nachtigall, L. G., Araujo, R. M., Nachtigall, G. R. (2016). Classification of apple tree disorders using convolutional neural networks. In 2016 IEEE 28th **International Conference on Tools with Artificial Intelligence** (ICTAI) (pp. 472-476). IEEE. San Jose, CA, ABD
Neuman, M. Sapirstein, E. Shwedyk, W. Bushuk. (1989). *Wheat grain colour analysis by digital image processing*. II. Wheat class discrimination. Journal of Cereal Science 10: 183-188.Computer Vision and Pattern Recognition (CVPR), pp. 580-587, 2014. Doi: 10.1016/S0733-5210(89)80047-5
Nizam, H., Akın, S. S. (2014). Sosyal medyada makine öğrenmesi ile duygu analizinde dengeli ve dengesiz veri setlerinin performanslarının karşılaştırılması. **XIX. Türkiye'de İnternet Konferansı**, 1(6). İstanbul Üniversitesi, İstanbul.
Okur, S. (2009). Parametrik ve parametrik olmayan basit doğrusal regresyon analiz yöntemlerinin karşılaştırmalı olarak incelenmesi. YuksekLisansTezi, Çukurova Universitesi, Adana.
O'Shea, K., Nash, R. (2015). *An introduction to convolutional neural networks*. arXiv preprint arXiv:1511.08458. Doi: 10.48550/arXiv.1511.08458
Öğücü, M. O. (2006). *Yapay sinir ağları ile sistem tanıma*, İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Doctoral dissertation. Doi: 504041113
Piccinini, G. (2004). *The First computational theory of mind and brain*: a close look at mcculloch and pitts's "logical calculus of ideas immanent in nervous activity". Synthese, 141, 175-215. Doi: 10.1023/B:SYNT.0000043018.52445.3e
Rana, K. (2020). *Pooling layer—short and simple*. Artificial Intelligence in Plain English.https://ai.plainenglish.io/pooling-layer-beginner-to-intermediate fa0dbdce
Rashidi, H. H., Tran, N. K., Betts, E. V., Howell, L. P., Green, R. (2019). *Artificial intelligence and machine learning in pathology: the present landscape of supervised methods.* Academic pathology, 6, 2374289519873088. Department of Pathology and Laboratory Medicine, University of California Davis, 4400 V St, Sacramento, CA 95817, USA.
Ren, S., He, K., Girshick, R., Sun, J. (2015). *Faster r-cnn: Towards real-time object detection with region proposal networks*. Advances in neural information processing systems, ISBN: 9781510825024
Sevi, M, Aydın, İ, Akın, E.(2023). *YOLOv5 ile Topluluk Öğrenmesine Dayalı Olarak Ray Yüzeyindeki Kusurların Tespiti*. 115-132. Doi: 10.47072/demiryolu.1205483, Fırat Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği, Elâzığ, Türkiye
Seyyarer, E., Ayata, F., Uçkan, T., Karci, A. (2020). *Derin öğrenmede kullanilan optimizasyon algoritmalarinin uygulanmasi ve kiyaslanmasi*. Computer Science, 5(2), 90-98. Van Yüzüncü Yıl Üniversitesi, Türkiye.
Singh, V., Misra, A. K. (2017). *Detection of plant leaf diseases using image segmentation and soft computing techniques*. Information processing in Agriculture, 4(1), 41-49. Doi: 10.1016/j.inpa.2016.10.005
Siuly, S., Alçin, Ö. F., Kabir, E., Şengür, A., Wang, H., Zhang, Y., Whittaker, F.

(2020). *A new framework for automatic detection of patients with mild cognitive impairment using resting-state EEG signals*. IEEE Transactions on Neural Systems and Rehabilitation Engineering, 28(9), 1966-1976. Doi: 10.1109/TNSRE.2020.3013429

Sladojevic, S., Arsenovic, M., Anderla, A., Culibrk, D., Stefanovic, D. (2016). *Deep neural networks based recognition of plant diseases by leaf image classification*. Computational intelligence and neuroscience. Doi: 10.1155/2016/3289801

Suthaharan, S. (2014). *Big data classification: Problems and challenges in network intrusion prediction with machine learning*. ACM SIGMETRICS Performance Evaluation Review, 41(4), 70-73. Doi: 10.1145/2627534.2627557

Sutton, R. S., Barto, A. G. (1998). **Introduction to reinforcement learning** (Vol. 135, pp. 223-260). Reinforcement Learning and Artificial Intelligence Laboratory Department of Computing Science University of Alberta, Canada.

Tan, Z. (2019). *Derin öğrenme yardımıyla araç sınıflandırma*, Master's thesis, Fırat Üniversitesi, Fen Bilimleri Enstitüsü, Diyarbakır.

TBD. (2019). Yapay zekâ Üzerine Çalıştay, Metin Ger, Sema Onurlu, **Türkiye' nin Yapay Zeka Stratejisi, TOBB ETÜ**. Türkiye Bilişim Derneği, Ankara.

Tecim, V., Aydın, C., Tarhan, Ç., Hakan, A., Komesli, M. (2022). *Üniversitlerde Akıllı Kampüs Uygulamaları İçin Altyapı Sistemi Oluşturulması.* Journal of Research in Business, 7(IMISC 2021 Special Issue), 132-147. Doi: 10.54452/jrb.1025423

Vapnik, V. N. (1999). *An overview of statistical learning theory*. IEEE transactions on neural networks, 10(5), 988-999. Doi: 10.1109/72.788640

Waslander, S. L., Hoffmann, G. M., Jang, J. S., Tomlin, C. J. (2005). Multi-agent quadrotor testbed control design: Integral sliding mode vs. reinforcement learning. In 2005 **IEEE/RSJ International Conference on Intelligent Robots and Systems** (pp. 3712-3717). IEEE. Edmonton, AB, Kanada.

Xu, R., Lin, H., Lu, K., Cao, L., Liu, Y. (2021). *A forest fire detection system based on ensemble learning*. Forests, 12(2), 217. Doi: 10.3390/f12020217

Yılmaz, D. Ö. Ü. A., Yayın, K. (2021). *Yapay Zeka. Kodlab Yayın Dağıtım Yazılım*, Erişim tarihi: 22. 06. 2022. Erişim Adresi: https://www.etiya.com/tr/urunler/yapay-zeka-platformu

Yukselturk, E., Ozekes, S. Türel, Y. K. (2014). *Predicting dropout student: an application of data mining methods in an online education program*. European Journal of Open, Distance and e-learning, 17(1), 118-133. Fırat Üniversitesi, Diyarbakır.

Yüceşahin, M. M., Tuysuz, S. (2011). *Ankara kentinde sosyo-mekânsal farklılaşmanın örüntüleri*: Ampirik bir analiz. Coğrafi Bilimler Dergisi, 9(2), 159-188. Doi: 10.1501/Cogbil_0000000123

Zakria, Z., Deng, J., Kumar, R., Khokhar, M. S., Cai, J., Kumar, J. (2022). *Multiscale and direction target detecting in remote sensing images via modified YOLO-v4*. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 15, 1039-1048. Doi: 10.1109/JSTARS.2022.3140776

ÖZ GEÇMİŞ

Kişisel Bilgiler

Adı Soyadı	:	Levent DALGIN

Eğitim Bilgileri

Lisans	:	
Üniversite	:	Van Yüzüncü Yıl Üniversitesi
Fakülte	:	Eğitim Fakültesi
Bölüm	:	Bilgisayar ve Öğretim Teknolojileri Öğretmenliği
Mezuniyet Yılı	:	2010
Yüksek Lisans	:	
Üniversite	:	Van Yüzüncü Yıl Üniversitesi
Enstitü	:	Fen Bilimleri Enstitüsü
Anabilim Dalı	:	Yapay Zeka ve Robotik Anabilim Dalı
Mezuniyet Yılı	:	2023

Akademik Yayınlar

Dalgın, L., Canayaz, M. (2023*). Investigation of strawberry fruit development using YOLO object detection algorithm,* **1st International Conference On Scientific and Innovative Studies ICSIS** 2023, Konya, Turkey.

VAN YÜZÜNCÜ YIL ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
LİSANSÜSTÜ TEZ ORİJİNALLİK RAPORU

Tarih. 22/06/2023

Tez Başlığı: "ÇİLEK MEYVESİ GELİŞİMİNİN DERİN ÖĞRENME METOTLARIYLA KARŞILAŞTIRMALI İNCELENMESİ"

Yukarıda başlığı belirtilen tez çalışmamın, kapak sayfası, giriş, ana bölümler ve sonuç bölümlerinden oluşan toplam 81 (seksen bir) sayfalık kısmına ilişkin, 05/05/2023 tarihinde tez danışmanım tarafından ithenticate adlı intihal tespit programından aşağıda belirtilen filtrelemeler uygulanarak alınmış olan orijinallik raporuna göre tezimin benzerlik oranı %16 (on altı) dır.

Uygulanan filtreler aşağıda verilmiştir:

- Kabul ve onay sayfası hariç,
- Teşekkür hariç,
- İçindekiler hariç,
- Simge ve kısaltmalar hariç,
- Gereç ve yöntemler hariç,
- Kaynakça hariç,
- Alıntılar hariç,
- Tezden çıkan yayınlar hariç,
- 7 kelimeden daha az örtüşme içeren metin kısımları hariç (Limit match size to 7 words)

Van Yüzüncü Yıl Üniversitesi Lisansüstü Tez Orijinallik Raporu Alınması ve Kullanılmasına İlişkin Yönergeyi inceledim ve bu yönergede belirtilen azami benzerlik oranlarına göre tez çalışmamın herhangi bir intihal içermediğini; aksinin tespit edileceği muhtemel durumda doğabilecek her türlü hukuki sorumluluğu kabul ettiğimi ve yukarıda vermiş olduğum bilgilerin doğru olduğunu beyan ederim.

Gereğini bilgilerinize arz ederim.

Tarih ve İmza

Adı Soyadı: Levent Dalgın
Öğrenci No: 20910001147
Anabilim Dalı: Yapay Zeka ve Robotik
Programı:
Statüsü: (x) Yüksek lisans ( ) Doktora

DANIŞMAN	**ENSTİTÜ ONAYI**
UYGUNDUR	UYGUNDUR