



T.C.
BATMAN ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ
MAKİNA MÜHENDİSLİĞİ ANABİLİM DALI

**DERİN ÖĞRENME YÖNTEMLERİ KULLANI-
LAK DİYARBAKIR YÖRESİ KARPUZU GÖ-
RÜNTÜLERİNDEN AĞIRLIĞININ TAHMİN
EDİLMESİ**

YÜKSEK LİSANS

Halil KAYRA

Danışman
Dr. Öğr. Üyesi Savaş KOÇ

Ağustos, 2022
BATMAN
Her Hakkı Saklıdır

TEZ KABUL VE ONAYI

Halil KAYRA tarafından hazırlanan “Derin Öğrenme Yöntemleri Kullanılarak Diyarbakır Yöresi Karpuzu Görüntülerinden Ağırlığının Tahmin Edilmesi” adlı tez çalışması 29/07/2022 tarihinde aşağıdaki jüri üyeleri tarafından oy birliği ile Batman Üniversitesi Lisansüstü Eğitim Enstitüsü Makine Mühendisliği Anabilim Dalı’nda YÜKSEK LİSANS Tezi olarak kabul edilmiştir.

Jüri Üyeleri

İmza

Başkan

Dr. Öğr. Üyesi Şehmus BADAY

.....

Danışman

Dr. Öğr. Üyesi Savaş KOÇ

.....

Üye

Dr. Öğr. Üyesi Kerim KARADAĞ

.....

Yukarıdaki sonucu onaylarım.

Prof. Dr. Osman PAKMA
Lisansüstü Eğitim Enstitüsü Müdür V.

TEZ BİLDİRİMİ

Bu tezdeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edildiğini ve tez yazım kurallarına uygun olarak hazırlanan bu çalışmada bana ait olmayan her türlü ifade ve bilginin kaynağına eksiksiz atıf yapıldığını bildiririm.

DECLARATION PAGE

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all materials and results that are not original to this work.

İmza

Halil KAYRA

Tarih: 17.08.2022

ÖZET

YÜKSEK LİSANS

DERİN ÖĞRENME YÖNTEMLERİ KULLANILARAK DİYARBAKIR YÖRESİ KARPUZU GÖRÜNTÜLERİNDEN AĞIRLIĞININ TAHMİN EDİLMESİ

Halil KAYRA

BATMAN ÜNİVERSİTESİ LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ
MAKİNA MÜHENDİSLİĞİ ANABİLİM DALI

Danışman: Dr. Öğr. Üyesi Savaş KOÇ

2022, 73 Sayfa

Jüri

Dr. Öğr. Üyesi Savaş KOÇ
Dr. Öğr. Üyesi Şehmus BADAY
Dr. Öğr. Üyesi Kerim KARADAĞ

Bu tez çalışmasında Diyarbakır yöresinde yetişen karpuzların görüntüleri kullanılarak derin öğrenme metotlarıyla karpuzların ağırlık tahmini yapılmıştır. Çalışmada 5000 adet karpuz görüntüsü kullanılmıştır. Diyarbakır yöresinde yetiştirilen karpuzların market ve semt pazarlarında fotoğrafları çekilerek bilgisayar ortamında kayıt altına alınmıştır. Toplanan karpuz görüntülerinin arka planları alındıktan sonra maskeleri Python programında yapılmıştır. Maskeler U-Net mimarisinde kullanılmak üzere eğitim dosyasına alınmıştır. Derin öğrenme yöntemlerinde evrişimli sinir ağları ile U-Net mimarisi kullanılmıştır. Görüntülerin segmentasyonu başarılı bir şekilde yapılmıştır. U-Net mimarisi %99,65 oranında başarılı bir şekilde karpuz görüntüsünü geometrik olarak tahmin etmiştir. U-Net modelinde elde edilen görüntülerden piksel alan metodu ile karpuzun görüntüde kapladığı alan oranı hesabı yapılmıştır. Görüntülerin boy ve en pikselleri belirlendikten sonra yapay sinir ağı ile eğitimlerine geçilmiştir. En iyi mimariyi bulmak için yapay sinir ağları 9 farklı mimari ile eğitimi yapılmıştır. En iyi mimari 4 gizli katmana sahip 1024 birime ayrılmış olan ve aktivasyon fonksiyonu olarak ReLU kullanılan mimari olmuştur. Görüntülerden elde edilen karpuz verilerinin çok katmanlı yapay sinir ağlarında eğitilmesi ile test doğruluk oranı %92,59 ve eğitim doğruluk oranı ise %94,43 bulunmuştur. Sonuç olarak oluşturulan derin öğrenme yöntemi sayesinde 65 cm mesafede fotoğrafı çekilen karpuz görüntülerinin kaç kilo aralığında olduğu tahmin edecek bir program oluşturularak dijital tarım alanındaki çalışmalara katkıda bulunacaktır.

Anahtar Kelimeler: Derin öğrenme, Evrişimli sinir ağları, Karpuz, U-Net, Yapay sinir ağları

ABSTRACT

MS THESIS

ESTIMATING WEIGHT FROM DIYARBAKIR REGION WATERMELON IMAGES USING DEEP LEARNING METHODS

Halil KAYRA

**INSTITUTE OF GRADUATE STUDIES
OF BATMAN UNIVERSITY
THE DEGREE OF MASTER OF SCIENCE
IN MECHANICAL ENGINEERING**

Advisor: Asst. Prof. Savaş KOÇ

2022, 73 Pages

Jury

Asst. Prof. Savaş KOÇ

Asst. Prof. Şehmus BADAY

Asst. Prof. Kerim KARADAĞ

In this thesis study, using the images of watermelons grown in the Diyarbakır region, the weight estimation of watermelons was made by deep learning methods. 5000 watermelon images were used in the study. Photographs of watermelons grown in the Diyarbakır region were taken in markets and neighborhood markets and recorded in a computer environment. After the backgrounds of the collected watermelon images were taken, their masks were made in the Python program. Masks are included in the training file to be used in the U-Net architecture. Convolutional neural networks and U-Net architecture are used in deep learning methods. Segmentation of images has been done successfully. The U-Net architecture has successfully predicted the watermelon image geometrically at a rate of 99.65%. The area ratio of the watermelon in the image was calculated by using the pixel area method from the images obtained in the U-Net model. After the height and width pixels of the images were determined, training with the artificial neural network was started. To find the best architecture, artificial neural networks were trained with 9 different architectures. The best architecture was the one that was divided into 1024 units with 4 hidden layers and ReLU was used as the activation function. By training the watermelon data obtained from the images in multilayer artificial neural networks, the test accuracy rate was 92.59% and the training accuracy rate was 94.43%. As a result, thanks to the deep learning method created, a program that will estimate the weight of watermelon images taken at a distance of 65 cm will contribute to the studies in the field of digital agriculture.

Keywords: Artificial neural networks, Convolutional neural networks, Deep learning, U-Net, Watermelon

ÖNSÖZ

Bu tez çalışması 5 bölümden meydana gelmektedir. Birinci bölümde tez çalışması hakkında bilgiler verilmiştir ve çalışmanın amacı açıklanmıştır. Konular açıklanıp tez hakkında bilgi edinme hedeflenmiştir.

İkinci bölümde kaynak araştırması için diğer çalışmalar hakkında bilgiler verilmiştir. Kaynaklar derin öğrenme alanında yapılan çalışmaların gelişim süreci izlenmiştir.

Üçüncü bölümde tez çalışmasının nasıl yapıldığı belirtilmiştir. Kullanılan yöntemler detaylı bir şekilde açıklanmıştır. Evrişimli sinir ağları kullanılarak U-Net mimarisinde karpuz görüntülerinin segmentasyon analizi yapılmıştır. Yapay sinir ağlarında ise karpuz görüntülerinden elde edilen verilerle eğitim yapılmıştır. Görsellerle çalışmanın içeriği zenginleştirilmiştir.

Dördüncü bölümde çalışmanın sonuçları açıklanmıştır. Sonuçlar grafikler şeklinde değerlendirilip görsellerle de desteklenmiştir. U-Net görüntü analizinde %99,65 oranında bir eğitim başarısı sağlanmıştır. Çok katmanlı yapay sinir ağlarında görüntülerden elde edilen verilerle yapılan eğitimde test doğruluk oranı %92,59 ve eğitim doğruluk oranı ise %94,43 bulunmuştur.

Beşinci bölümde çalışmanın sonuçları açıklanmıştır. Sonuçlara göre önerilerde bulunulmuştur.

Bu tez çalışmasında her zaman güler yüzü ve samimiyetiyle yanımda olan danışman hocam Dr. Öğr. Üyesi Savaş KOÇ'a çok teşekkür ederim.

Tez çalışması sırasında maddi ve manevi yardımcı olan Annem Ayfer KAYRA, babam Tahsin KAYRA, kardeşlerim Mazlum KAYRA, Mesut KAYRA, Melike KAYRA, amcam Ahmet KAYRA, kuzenim Evin BAŞAK ERÇEL, arkadaşım Yusuf TOFTİK ve Makine Mühendisleri Sedat Emin KAYA ve Abdullah ELBEY'e çok teşekkür ederim.

Halil KAYRA
BATMAN-2022

İÇİNDEKİLER

ÖZET	iv
ABSTRACT.....	v
ÖNSÖZ	vi
İÇİNDEKİLER	vii
SİMGELER VE KISALTMALAR	x
ÇİZELGE DİZİNİ	xi
ŞEKİL DİZİNİ	xii
1. GİRİŞ	1
1.1. Yöresel Diyarbakır Karpuzu	4
1.2. Derin öğrenme.....	5
1.3. Derin Öğrenme Uygulamaları.....	6
1.3.1. Derin öğrenmenin özellikleri.....	7
1.4. Yapay Sinir Ağları	7
1.4.1. Yapay sinir hücresi ve genel yapısı	8
1.5. Aktivasyon Fonksiyonları	9
1.5.1. Birim basamak fonksiyonu.....	10
1.5.2. Doğrusal fonksiyonu	11
1.5.3. Sigmoid fonksiyonu.....	11
1.5.4. Hiperbolik tanjant fonksiyonu	12
1.5.5. ReLU (Rectified Linear Unit) fonksiyonu	13
1.5.6. Leaky ReLU fonksiyonu	14
1.5.7. Swish (A Self-Gated/Kendinden Geçitli) fonksiyonu	14
1.5.8. Softmax fonksiyonu.....	15
1.6. Yapay Sinir Ağlarının Sınıflandırılması	16
1.7. Derin Öğrenme Platformları	17
1.7.1. Torch.....	17
1.7.2. TensorFlow	17
1.7.3. Keras.....	18
1.7.4. DL4J	18

1.8. Derin Öğrenmede Kullanılan Algoritma Türleri.....	18
1.8.1. Evrişimli sinir ağları (CNN).....	18
1.9. Segmentasyon İçin Derin Öğrenme	19
1.9.1. Eşik	20
1.9.2. Kenar bazlı segmentasyon	20
1.9.3. Bölge bazlı segmentasyon	21
1.10. Semantik segmentasyon	21
1.10.1. U-Net	22
1.10.2. Mask R-CNN.....	23
1.10.3. DeepLab	24
2. KAYNAK ARAŞTIRMASI	26
3. MATERYAL ve YÖNTEM	34
3.1. Verilerin Elde Edilmesi.....	34
3.2. Verilerin İşlenmesi	36
3.2.1. Anaconda	36
3.2.2. Spyder.....	36
3.2.3. Python.....	36
3.3. Python Kütüphaneleri.....	37
3.3.1. Numpy	37
3.3.2. Pandas.....	37
3.3.3. Matplotlib	37
3.3.4. Seaborn	38
3.3.5. OpenCV	38
3.3.6. Os.....	38
3.3.7. Scikit Learn	38
3.3.8. Scikit-image (Skimage)	38
3.3.9. TensorBoard	39
3.3.10. Tqdm.....	39
3.3.11. Random.....	39
3.4. Matlab	39
3.5. U-Net mimarisi ile görüntü alanını belirleme	40
3.5.1. Gerekli kütüphanelerin yüklenmesi.....	40
3.5.2. Maske oluşturma	40
3.5.3. Verilerin hazırlanması	42

3.5.4. U-Net modelinin oluşturulması	42
3.6. Piksel İle Karpuzların En ve Boylarının Belirlenmesi	46
3.7. Piksel alanın bulunması.....	47
3.8. Verilerin Yapay Sinir Ağları İle Eğitilmesi	52
4. ARAŞTIRMA BULGULARI ve TARTIŞMA	55
4.1. U-Net Sonuçları	55
4.2. Yapay Sinir Ağları Sonuçları	58
4.3. En İyi Sonucun Değerlendirilmesi	61
5. SONUÇLAR ve ÖNERİLER.....	64
5.1. Sonuç.....	64
5.2. Öneriler	64
KAYNAKLAR	66
ÖZGEÇMİŞ	73

SİMGELER VE KISALTMALAR

Simgeler

A	Karpuz şeklinin kapladığı oran
e	Euler sayısı
g	Gram
ka	Karpuz şeklinin kapladığı alan
kg	Kilogram
L	Kayıp fonksiyonu
pa	Tüm piksel alanı

Kısaltmalar

CNN	Evrişimli sinir ağı
DCNN	Derin evrişimli sinir ağları
DÖ	Derin öğrenme
GAN	Üretken düşman ağları
MLP	Çok katmanlı algılayıcılar
MÖ	Makine öğrenimi
MP	Megapiksel
ReLU	Rektifiye edilmiş lineer birim
RNN	Tekrarlayan sinir ağı
YSA	Yapay sinir ağı
YZ	Yapay zeka

ÇİZELGE DİZİNİ

Çizelge 3.1. Karpuz ağırlıklarına göre sınıflandırma işlemi	53
Çizelge 3.2. Kullanılan verilerin sıralandırılması.....	53
Çizelge 4.1. Eğitim Sonuçları.....	59



ŞEKİL DİZİNİ

Şekil 1.1. Diyarbakır Karpuzu	5
Şekil 1.2. Yapay zekada derin öğrenmenin yeri	6
Şekil 1.3. Sinir hücresi yapısı	8
Şekil 1.4. Yapay sinir ağının katmanları	8
Şekil 1.5. Yapay sinir hücresinin yapısı	9
Şekil 1.6. Birim basamak fonksiyonu grafiği	10
Şekil 1.7. Doğrusal fonksiyonu grafiği	11
Şekil 1.8. Sigmoid fonksiyonu grafiği	12
Şekil 1.9. Hiperbolik tanjant fonksiyonu grafiği	13
Şekil 1.10. ReLU fonksiyonu grafiği	13
Şekil 1.11. Leaky ReLU fonksiyonu grafiği	14
Şekil 1.12. Swish fonksiyonu grafiği	15
Şekil 1.13. Softmax fonksiyonu grafiği	16
Şekil 1.14. MLP yapısı	17
Şekil 1.15. CNN yapısı ve katmanları	19
Şekil 1.16. a) Nesne algılama b) Semantik segmentasyon c) Örnek segmentasyon	21
Şekil 1.17. U-Net mimarisi	22
Şekil 1.18. Mask R-CNN mimarisi	24
Şekil 1.19. DeepLab mimarisi	25
Şekil 3.1. Diyarbakır Karpuzu'nun satıldığı yerlerden kareler	34
Şekil 3.2. 576 numaralı karpuzun fotoğrafının çekim esnası	34
Şekil 3.3. Beyaz örtüde çekilen 243 ve siyah örtüde çekilen 504 numaralı karpuz	35
Şekil 3.4. 595 numaralı karpuz ve 600 numaralı karpuzların hassas tartım terazisi ile tartılması	35
Şekil 3.5. Python kütüphanelerinin yükleme kodları	40
Şekil 3.6. 50 numaralı karpuzun arka planının silinmesi	41
Şekil 3.7. Maske oluşturma kodu	41
Şekil 3.8. 50 numaralı karpuzun silinen arka planının maske görüntüsü	42
Şekil 3.9. U-Net modelinin akış şeması	43
Şekil 3.10. U-Net modelinin kodlayıcı kısmı kodları	45
Şekil 3.11. U-Net modelinin kod çözücü kısmı kodları	46
Şekil 3.12. Paint'de piksel yardımıyla karpuzların en ve boyunun belirlenmesi	47
Şekil 3.13. Piksel ile alan bulma kodu	47
Şekil 3.14. Piksel boyu ile piksel eninin karşılaştırılması	48
Şekil 3.15. Piksel boyu ile ka/pa oranının karşılaştırılması	48
Şekil 3.16. Piksel boyu ile kg'ların karşılaştırılması	49
Şekil 3.17. Piksel boyu ile sınıflandırmanın karşılaştırılması	49
Şekil 3.18. Piksel eni ka/pa oranının karşılaştırılması	50
Şekil 3.19. Piksel eni ile kg'ların karşılaştırılması	50
Şekil 3.20. Piksel eni ile sınıflandırmanın karşılaştırılması	51
Şekil 3.21. ka/pa oranı ile kg'ların karşılaştırılması	51
Şekil 3.22. ka/pa oranının sınıflandırma ile karşılaştırılması	52
Şekil 3.23. Kg'lar ile sınıflandırmanın karşılaştırılması	52
Şekil 3.24. YSA modelinin tamamlama ve çalıştırma kodları	54
Şekil 3.25. YSA'nın modeli	54
Şekil 4.1. Tur sayısına göre doğruluk grafiği	55
Şekil 4.2. Tur sayısına göre kayıp grafiği	56

Şekil 4.3. Orjinal görüntü boyutu 128x128 olarak değiştirilmiş b) Maskesi c) U-Net mimarisinin tahmin ettiği segmentasyon	57
Şekil 4.4. a) 80 numaralı karpuzun orijinal görüntü boyutu 128x128 olarak değiştirilmiş b) U-Net mimarisinin tahmin ettiği segmentasyonu	58
Şekil 4.5. a) 95 numaralı karpuzun orijinal görüntü boyutu 128x128 olarak değiştirilmiş b) U-Net mimarisinin tahmin ettiği segmentasyonu	58
Şekil 4.6. Aktivasyon fonksiyonlarının eğitim başarıları	60
Şekil 4.7. Aktivasyon fonksiyonlarının eğitim kayıp oranları	61
Şekil 4.8. Doğruluk başarıları	62
Şekil 4.9. Kayıp oranları	62
Şekil 4.10. Tahmin edilen doğru ve yanlış görüntü sayısı	63



1. GİRİŞ

Tarım, insanoğlunun avcılığı bırakıp yerleşik düzene geçişinin bir parçasıdır. Bu günlerde tarım ileri teknolojilerle artık daha kolay yapılmakta ve daha çok verim alınmaktadır. Tarım ürünlerini kontrol etmek ve onlar hakkında daha fazla bilgi toplamak için en son teknolojilerle analizler yapılmaktadır. Görüntüleri çekilen tarım ürünlerinin şekillerine bakılarak yorum yapılabilir ve fikirler sunulabilir. Bu fikirlerden biri karpuzların şekillerine bakarak tartmadan ağırlığı tahmin etme gibi zamandan tasarruf uygulamalarının yapılabilmesidir. Geleneksel tarım yerine artık dijital tarım tercih edilerek zamandan tasarruf ve kaliteli ürün elde edilebilir. Dijital tarımda bilgisayar yardımıyla bitkilerin verimi kontrol edilebilir.

Tahmin edildiği gibi tarım teknolojisi alanı da diğer alanlar gibi büyümekte ve yoğun bir iş haline gelmektedir. Tarım teknolojisinin ilerlemesiyle birlikte geleneksel üretim sistemleri modern, üretken ve yenilikçi sistemlere dönüşmektedir. Son zamanlarda çiftçiler, üretim faaliyetlerinin yanı sıra yönetsel faaliyetlerin de çevresel faktörlerle daha fazla etkileşim gerektiren yeni modeller haline geldiğini fark etmektedir. Tarım kavramı son zamanlarda su tasarrufu, akıllı tarım, kaliteli tarım, yüksek verim ve çevreyi kirletmeyen tarım anlamına gelmektedir. Tüm bu dönüşümleri gerçekleştirmek için en gerekli ve etkili yaklaşım dijital tarımdır (Ozdogan ve ark., 2017).

Dijital devrim, üretkenliği ve karar vermeyi artırmak için modern makineler, bilgisayarlı araçlar ve bilgi ve iletişim teknolojilerini (BİT) kullanarak tarımı dönüştüren bir etkiye sahiptir. Küresel konumlandırma sistemi (GPS) ve uzaktan algılamadan büyük verilere, yapay zeka (YZ) ve makine öğreniminden robotik ve nesnelerin interneti'nden tarıma kadar birçok yenilikçi teknolojinin uygulanması, üretimin artmasına, maliyetlerin düşmesine ve çevre üzerinde daha az etkiye yol açmaktadır. Veriye dayalı çözümler, sürdürülebilir ve kaynak tasarrufu sağlayan üretimin potansiyelini ortaya çıkaran bir devrimdir (Fountas ve ark., 2020).

Görüntü işlemede sinir ağlarının 200'den fazla uygulaması bulunmaktadır. Bilgisayarda görme, endüstriyel uygulamalar, biyomedikal uygulamalar, hava gözlem ve tahmin uygulamaları gibi son zamanlarda insan hayatına etkisi artan uygulamalardır. Görüntü işleme algoritmaları için yeni iki boyutlu sınıflandırma altında çeşitli uygulamalar halinde kategorilere ayrılmaktadır. Birinci boyut, algoritma tarafından gerçekleştirilen görevlerin türünü belirtmektedir. Bu boyutta ön işleme, veri azaltma/özellik çı-

karma, segmentasyon, nesne tanıma, görüntü anlama ve optimizasyon gibi görev türlerini içermektedir. İkinci boyut, algoritma tarafından işlenen girdi verilerinin soyutlama düzeyini yakalamaktadır. Bu boyutta piksel düzeyi, yerel özellik düzeyi, yapı düzeyi, nesne düzeyi, nesnelerin toplama düzeyi ve sahne özellikleri gibi düzeylerdir. Altı görev türünün her biri, sinirsel temelli yaklaşıma belirli kısıtlamalar getirirken bu özel koşullar ayrıntılı olarak tartışılmaktadır. Görüntü işlemede örüntü tanıma tekniklerinin uygulanması ve özellikle sinir ağlarının uygulanması ile ilgili çözülmemiş problemlerin bir derlemesi yapılır. Son olarak, sinir ağlarının gelecekteki uygulamalarına genel bir bakış sunarak bunları yeni gelişmelerle ilişkilendirebilir (Egmont-Petersen ve ark., 2002).

Yapay zeka, nesnelerin interneti ve blok zinciri (blockchain) teknolojisi gibi yeni nesil bilgi ve iletişim teknolojilerinin yaygınlaşması teknolojik ve endüstriyel devrimi hızlandırmaktadır. YZ hükümetlerden, endüstrilerden ve akademilerden büyük bir ilgi görmektedir. Tüm bu gelişmeler, bilgi entegrasyonuna dayalı bir YZ incelemesi sunmayı amaçlamaktadır. Arka plan, sürücüler, teknolojiler ve uygulamaların yanı sıra yapay zekanın geliştirilmesine ilişkin mantıksal görüşleri kullanarak yapay zekanın kapsamının artırılması gerekmektedir (Zhang ve Lu, 2021).

Görüntü bölütleme yani segmentasyon, renkli görüntü analizi ve örüntü tanıma alanında çok gelişmekte olan bir araştırma konusudur. Bu amaçla birçok gelişmiş algoritma geliştirilmiştir. Ancak, bu algoritmalar genellikle segmentasyon sonuçları, yanlış sınıflandırma ve aşırı segmentasyondan muzdarip görünmektedir. Bunların arkasındaki nedenler ise çekim, aktarım ve renk alanı dönüştürme sırasında görüntü kalitesinin bozulmasıdır. Bu nedenle, segmentasyona geçmeden renkli görüntüden fazla pikselleri veya gürültüleri kaldırabilen verimli bir görüntü iyileştirme tekniği gerekmektedir (Bora, 2017).

Akıllı sistemler için yapay sinir ağları (YSA) tarımsal araştırmalarda karmaşık sorunlarda çeşitli çözümler sunan yeni bir teknoloji haline gelmektedir. Arızalarda klasik makinelerin çözemediği birçok sorunu çözebildiği için YSA, özellikle toplum için daha iyi ürünlerin inovasyonunda ve geliştirilmesinde önemli hale gelmektedir. YSA mimarisinin temelleri, girdi parametreleri olarak farklı ürün performans faktörlerinin kullanıldığı ürün verimi tahmininde başarıyla uygulanmasına dayanmaktadır. Spesifik olarak, YSA özellikle mahsul verimini tahmin etme söz konusu olduğunda, bir dizi tarımsal problemi incelemek için en iyi çözümlerden biridir. Mahsul verimiyle bitkilerin sağlığı ve kalitesi incelenebilmektedir. YSA'nın verim değişimini diğer yöntemlere

göre daha iyi açıkladığı sonucuna varılmaktadır. YSA'nın hassas tarıma uygulanması, hassas tarım kavramının küresel gıda talebini karşılamının sürdürülebilir bir yolu olarak gelecekte değerlendirilmesinde önemli bir rol oynamaktadır. Bununla birlikte, gelecekteki gıda talebinin sürdürülebilirliğini sağlamak için YSA'nın mahsul verimi üzerindeki etkisi hakkında daha fazla araştırmaya ihtiyaç vardır (Khairunniza-Bejo, ve ark., 2014).

Derin öğrenmenin (DÖ) ortaya çıkışı, endüstride teşhis ve prognostik teknikleri değiştirmektedir. Endüstriyel teşhiste muazzam ilerleme kaydederek Endüstri 4.0'ın sürdürülmesinde önemli bir rol oynayarak Endüstri 5.0 yolunu açmaktadır. En iyi örnek olarak makineler başta olmak üzere endüstriyel alt sistemlerin durum izlemesinde popülerlik kazanmaktadır. Farklı uygulamalardaki makineler çeşitli nedenlerle arızalanmaya başlamaktadır. Bu nedenle durumlarını izlemek ve arızalarını takip etmek çok önemlidir. Giriş verileri ve özellik işleme teknikleri açısından makineler için DÖ tabanlı durum izlemenin modern durumunun bir değerlendirmesini sağlamaktadır. Özellikle işleme tekniklerini kullanarak hedeflenen problemler üzerinde makine durumunun izlenmesinde DÖ modellerinin etkinliğine çeşitli girdi özelliklerinin uygulanmasını test ederek bu özellikleri ve nasıl ele alındığını açıklamaktadır. Ayrıca DÖ modellemesindeki ilerlemeleri, makineler için DÖ tabanlı tanı yöntemlerini ve hibrit tanılama tekniklerini tartışarak değerlendirmektedir. Bu gözden geçirme araştırmacılara, özellik işleme ile ilgili araştırma boşluklarını belirlemede yardımcı olurken makine durum izlemesine uygulanan DÖ modellerinin uygulanmasına etkin bir şekilde katkıda bulunabilmektedir (Soother ve ark., 2021).

DÖ, umut verici sonuçlar ve büyük potansiyel ile görüntü işleme ve veri analizi için yeni ve modern bir teknik yaratarak DÖ çeşitli alanlarda başarıyla uygulanarak son zamanlarda tarım alanına da girmektedir. İncelenen belirli tarımsal sorunlara, kullanılan belirli modeller ve çerçevelere, kullanılan verilerin kaynaklarına, doğasına ve ön işlenmesine ve ele alınan her bir işte kullanılan kriterlere göre elde edilen genel performansları incelemektedir. Ayrıca, sınıflandırma veya regresyon performansındaki farklılıklar için derin öğrenmenin şu anda popüler olan diğer tekniklerle karşılaştırmalarına da olanak sağlamaktadır. Sonuçlar incelendiğinde derin öğrenmenin yüksek doğruluk sağladığı ve yaygın olarak kullanılan görüntü işleme tekniklerinden daha iyi performans gösterdiği görülmektedir. (Kamilaris ve Prenafeta-Boldú, 2018).

İnsanoğlunun varoluşundan beri tarım, hem beslenme hem de ekonomik bir olgu olurken insan nüfusunun sürekli artmasından dolayı tarım alanlarının zamanla yetersiz

hale gelecek olması kaçınılmazdır. Tarımda artan talebi karşılamak zamanla imkansız hale gelecektir. İnsanlar gelecekte kendisini bekleyen bu sorunları daha dar tarım alanları kullanarak ve daha yüksek verim ile derin öğrenme yöntemleri kullanılarak çözmeyi umut etmektedir. Tarımda birçok ürün insanların geçim kaynağı olarak yetiştirilirken bazı ürünlerde ülke ekonomisine ciddi bir şekilde getiri sağlamaktadır. Bu ürünler ülkeye önemli bir ekonomik getiri sağlıyorsa o zaman bu ürünün olabildiğince kaliteli ve yüksek verimli olması için çalışmalar yapmak gerekmektedir.

Diyarbakır yöresi ve etrafındaki arazilerde yetiştirilen karpuzların satılması bazen tarlada göz kararıyla toptan bir şekilde satılmaktadır. Bazen de tarladan toplandıktan sonra satıcılara verilmektedir. Bu çalışmada tarlada yetiştirilen karpuzların market ve pazarlarda tek tek fotoğrafları çekilerek görüntülerinden ve piksel ile enine ve boyuna alınan ölçülerinden yararlanarak karpuzun kaç kilo geleceğini tahmin edecek şekilde derin öğrenme metotlarından oluşan YZ mimarileri kullanılmaktadır. Farklı yörelerde yetiştirilen karpuzların yoğunlukları farklı olduğundan özellikle Diyarbakır yöresi ve etrafındaki tarlalarda yetiştirilen karpuz numunelerinden veriler elde edilmektedir.

1.1. Yöresel Diyarbakır Karpuzu

Diyarbakır karpuzu boyutlarıyla dünyaca ünlüdür ve latince adı *Citrullus lanatus* türüne ait olan bir karpuz türüdür. Yuvarlak oval şekilli, alacalı karpuz olarak sınıflandırılmaktadır. Diyarbakır ve çevresinde yetişen bu karpuz türü Sürme, Pembe, Beyaz Kış, Kara Kış ve Ferik olarak bilinen çeşitleriyle yaygın olarak yetiştirilmektedir. Ancak günümüzde bu çeşitlerden sadece Sürme yetiştirilmektedir. Sürme karpuzlar çok büyük olmalarıyla ünlüdür (Anonim, 2022).

Diyarbakır karpuzunun diğer karpuzlardan ayırt edilen özellikleri şunlardır.

- Meyve şekilleri; Yuvarlak veya oval şekilde olabilir. Dış kabuğun rengi parlak ve yeşil arka plan üzerinde koyu yeşil geniş uzunlamasına dilimler halinde çizgilidir. Bu çizgiler arka plan rengine karşı düzenli ve oldukça koyu görünmektedir.
- Tohumlar siyah ve küçüktür. Meyvedeki toplam tohum sayısı değişmekle birlikte ortalama 650-750 tohumdur.
- Diyarbakır Karpuzu tek yıllık ve sürüngendir.

Çeşitli zamanlarda Diyarbakır Karpuz Festivalleri yapılmaktadır. Festivalde Diyarbakır Karpuzu'nun ağırlıklarına göre çeşitli düzeylerde yarışmalar yapılmakta olup

karpuz yetiřtiricilerine ödöller verilmektedir. řekil 1.1’de Diyarbakır Karpuzu gösterilmektedir.



řekil 1.1. Diyarbakır Karpuzu

1.2. Derin öęrenme

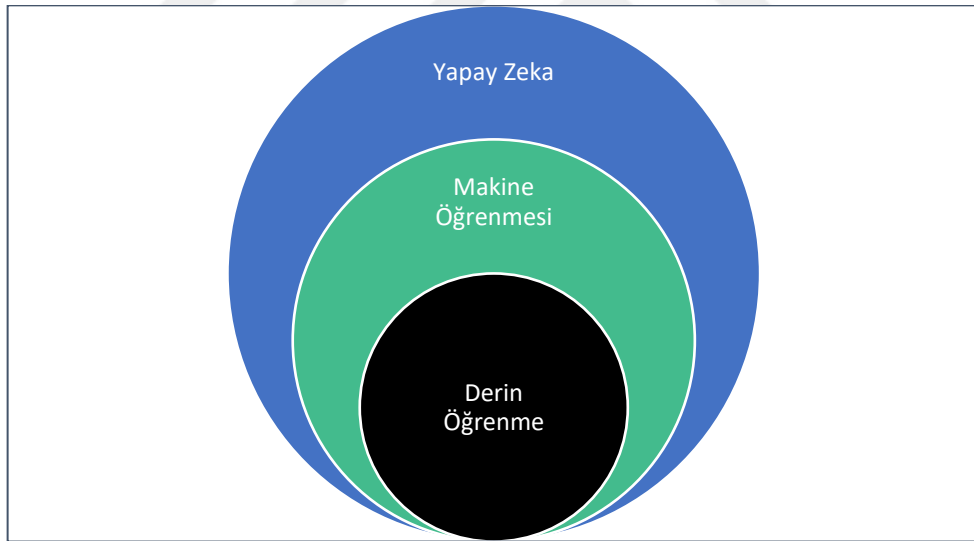
Yapay zeka günümüz toplumunun birçok alanında yeniden canlanarak önemli bir teknoloji haline gelmektedir. Yapay zekayı pratik ve ekonomik hale getiren, yüksek performanslı bilgi işlem teknolojisinin gelişmesi sayesinde. Bilgisayarla görme, konuşma, tanıma, makine çevirisi, sürücüsüz arabalar yapay zekanın örnek ve başarılı uygulama alanlarıdır.

Makine öğrenimi (MÖ), insanların nasıl öğrendiğini taklit etmek için verilere ve algoritmalara odaklanan ve makineler tarafından verilen kararların doğruluğunu kademeli olarak artıran bir YZ dalıdır. MÖ, geleceęi tahmin etmede geçmiş verilerden yinelenmeli olarak öğrenmek için farklı algoritmalar kullanır. Fiziksel veya teorik modellere dayalı geleneksel karar verme tekniklerinden farklı olarak MÖ, veriye dayalıdır.

Denetimsiz öğrenme, denetimli öğrenme ve pekiřtirmeli öğrenme olmak üzere üç tür makine öğrenimi vardır. Denetimsiz öğrenmede, veriler önceden tanımlanmış etiketler veya kategorilerle ilişkilendirilmezken öğrenme ise verileri benzer özellik veya özelliklere sahip sınıflar halinde gruplandırma sürecidir. Denetimli öğrenmede, her veri bir etiketle ilişkilendirilir ve öğrenme, çıktı etiketi ile gerçek etiket arasındaki uyumsuz-

luğu azaltmak için bir işlevi genelleştirme sürecidir. Pekiştirmeli öğrenmede ise nesnel veya nesnel bir duruma sahip bir durum makinesi ve durum makinesini atlamak için bir yöntem vardır; tüm verilerin etiketi yoktur, ancak hedef duruma ulaşmak için durum makinesinin durumunu bir eğitim hedefiyle değiştirir (Ngo, 2021).

DÖ, çeşitli görevleri gerçekleştirmek için tasarlanmış çok çeşitli sinir ağı mimarilerini kapsayan önemli bir yeni makine öğrenimi alanıdır. Tıbbi görüntüleme alanında, makine mühendisliği alanında ve birçok alanda görüntü sınıflandırılmasında tercih edilen bir alandır. Görüntüler için kullanılan en popüler ağ mimarisi, evrişimli sinir ağıdır (CNN). Geleneksel makine öğrenimi, algoritmanın öğrendiği özelliklerin belirlenmesini ve hesaplanmasını gerektirirken, derin öğrenme yöntemleri ise yeni veriler için tahminler yaparak bu tahminlerin özelliklerinin uygun ağırlıklarını öğrenir (Erickson ve ark., 2017). Şekil 1.2.'de derin öğrenmenin yapay zekada yeri gösterilmektedir (Anonim, 2022). Derin öğrenmenin makine öğrenmesinin bir parçası olduğu görülmektedir. Makine öğrenmesinin de yapay zekanın alt dalı olduğu görülmektedir.



Şekil 1.2. Yapay zekada derin öğrenmenin yeri

1.3. Derin Öğrenme Uygulamaları

Derin öğrenme, yağmur, deprem ve tsunamiler hakkında hava durumu tahminleri yapmak için yaygın olarak kullanılır. Gerekli önlemlerin alınmasına yardımcı olur. Derin öğrenme ile makineler konuşmayı anlayabilir ve gerekli çıktıyı sağlayabilir. Makinelerin kendisine beslenen görüntülerdeki insanları ve nesneleri tanımasını sağlar.

Derin öğrenme modelleri ayrıca, reklam verenlerin gerçek zamanlı teklif verme ve hedefli görüntülü reklamcılık gerçekleştirmek için verilerden yararlanmasına yardımcı olur (Anonymous, 2022).

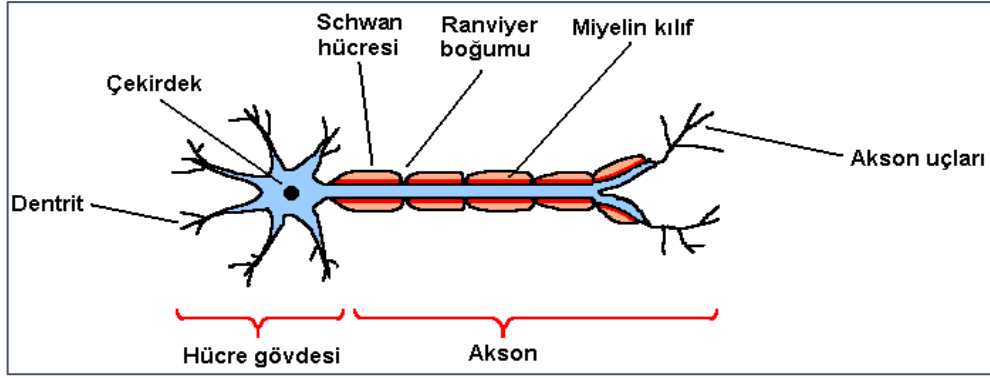
Günümüzde nesne tanıma ve sahnelerin havadan görüntü sınıflandırması derin öğrenme teknikleri kullanılarak tarımsal uygulamalarda önemli bir rol kazanmıştır. Bu tür uygulamalarda, drone'lar havadan görüntü elde etmek için düşük maliyetli bir platform sağlarken, derin öğrenme öncelikle bitki sayımı ve tanımlaması için kullanılır. Çeşitli uygulamalar bu amaç için derin öğrenme tekniklerini kullanılmıştır. Bitki verimini en üst düzeye çıkarmak ve mahsullerin durumunu izlemek için güçlü sistemler derin öğrenme ile denetlenmektedir (Carrio ve ark., 2017).

1.3.1. Derin öğrenmenin özellikleri

Derin öğrenmenin özelliklerini anlamak için makine öğrenmesi ile karşılaştırılması yapılmalıdır. Aşağıda maddeler halinde bu özellikler belirtilmiştir. Makine öğrenimi yalnızca yapılandırılmış ve yarı yapılandırılmış veri kümeleriyle çalışırken derin öğrenme ise hem yapılandırılmış hem de yapılandırılmamış verilerle çalışır. Derin öğrenme algoritmaları karmaşık işlemleri verimli bir şekilde gerçekleştirebilirken makine öğrenme algoritmaları yapamaz. Makine öğrenimi algoritmaları, kalıpları çıkarmak için etiketli örnek verileri kullanırken, derin öğrenme büyük hacimli verileri girdi olarak kabul eder ve bir nesneden özellikler çıkarmak için girdi verilerini analiz eder. Veri sayısı arttıkça makine öğrenmesi algoritmalarının performansı azalırken modelin performansını korumak için derin bir öğrenmeye ihtiyaç duyulur (Anonymous, 2022).

1.4. Yapay Sinir Ağları

İnsan sinir sistemi nöron adı verilen hücrelerden oluşur. Bu tür hücrelerin (nöronlar) birbirine bağlanması, sinir ağını oluşturur. İnsan sinir sisteminde yaklaşık 1011 nöron ve birbirleriyle yaklaşık 10000 bağlantı vardır. YSA (Yapay Sinir Ağı), yapay nöronların beyin ağı ile benzer bir şekilde bağlandığı doğal sinir ağının bir taklididir (Kukreja ve ark., 2016). Şekil 1.3'de bir sinir hücresinin yapısı gösterilmektedir (Abacı, 2022).

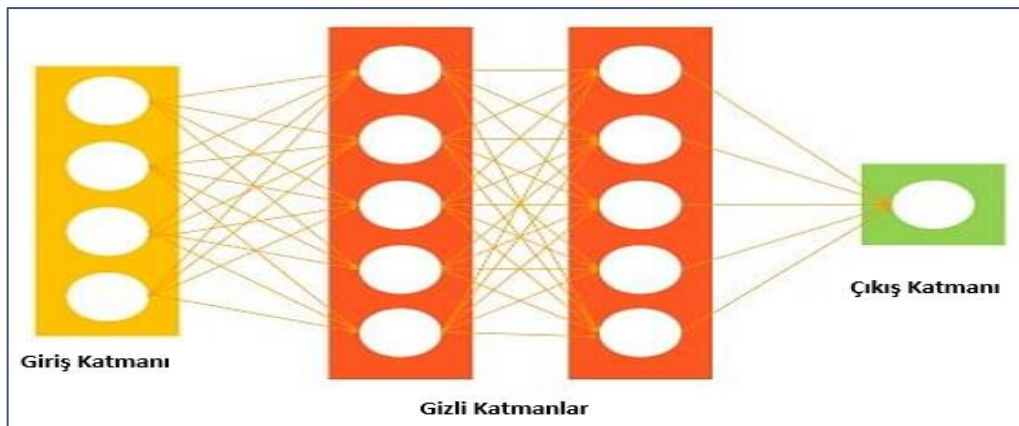


Şekil 1.3. Sinir hücresi yapısı

Bir yapay sinir ağı, nöron adı verilen işlem birimlerinden oluşur. Yapay bir nöron, doğal nöronun yapısını ve davranışını kopyalamaya çalışır. Bir nöron, girdilerden (dendritler) ve bir çıktıdan (akson yoluyla sinaps) oluşur. Nöronun aktivasyonunu belirleyen bir işlevi vardır (Kukreja ve ark., 2016).

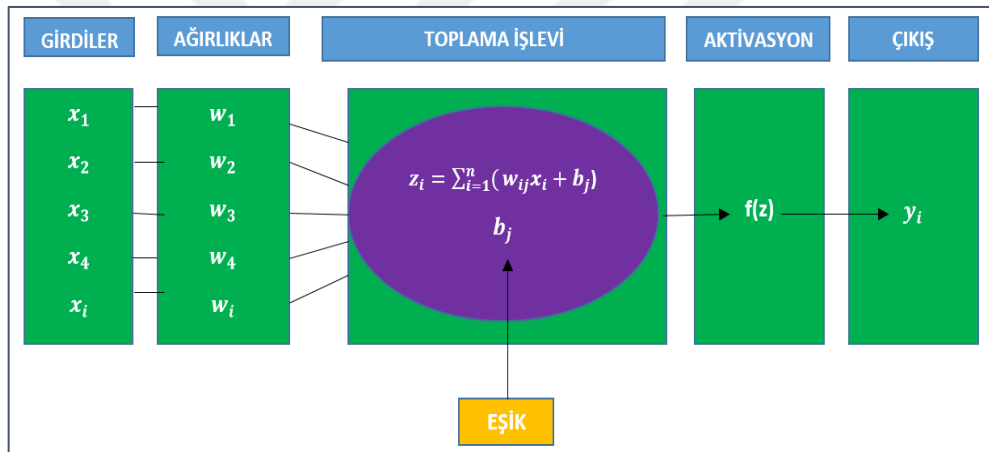
1.4.1. Yapay sinir hücresi ve genel yapısı

Bir YSA'nın amacı, girdi değerleri ile çıktı değerleri arasında bağlantı kurarak hesaplama yapmaktır. YSA'nın iki aşamalı bir çalışma modu vardır. Eğitim, bazı girdi ve çıktı veri setleri kullanılarak gerçekleştirilir. YSA bu verilerin bir genellemesini yapar. YSA girdiler, çıktılar ve gizli katmanlar olmak üzere üç katmandan oluşur ve her katmanda çok sayıda nöron bulunmaktadır (Tumer ve Edebalı, 2015). Şekil 1.4'de yapay sinir ağının katmanları gösterilmektedir (Anonymous, 2022).



Şekil 1.4. Yapay sinir ağının katmanları

Bir nöron, Şekil 1.5’de gösterildiği gibi; girdiler, ağırlıklar, toplama fonksiyonu, aktivasyon (transfer) fonksiyonu ve çıktı olmak üzere beş ana unsurdan oluşmaktadır. Girdi bilgileri, dış ağlardan veya diğer nöronlardan gelir. Bazı durumlarda, nöronlar geri bildirimle kendi kendine girdi oluşturabilir. Bir nöron genellikle birçok eşzamanlı girdi alır. Her girdinin, girdiyi işleme ögesinin toplama işlevi üzerinde ihtiyaç duyduğu etkiyi veren kendi görelî ağırlığı vardır. Ağırlıklar, yapay nöron tarafından kaydedildiği şekliyle giriş sinyalinin yoğunluğunu belirleyen ağ içindeki uyarlanabilir katsayılarıdır. Bir girişin bağlantı gücünün bir ölçüsüdür. Bu güçlü yönler, çeşitli eğitim setlerine yanıt olarak ve bir ağı özel topolojisine göre veya öğrenme kuralları aracılığıyla değiştirilebilir. Bir işleme elemanının işlemindeki ilk adım, tüm girdilerin ağırlıklı toplamını hesaplamaktır. Toplama işlevi işlemi, işleme ögelerinden gelen tüm girdileri toplar (Tümer ve Edebalı, 2015).



Şekil 1.5. Yapay sinir hücresinin yapısı

1.5. Aktivasyon Fonksiyonları

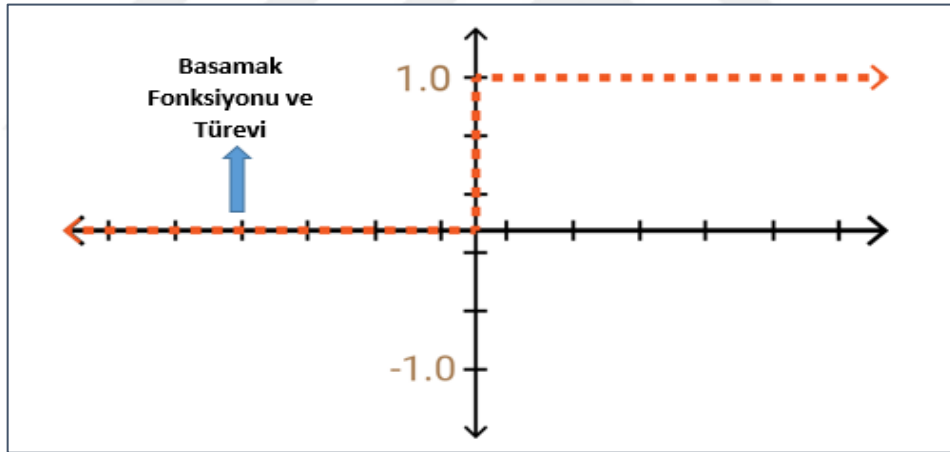
Bilimsel çalışmalarda en çok kullanılan aktivasyon fonksiyonları verilmiştir (Sharma ve ark., 2017).

- Birim basamak fonksiyonu
- Doğrusal fonksiyonu
- Sigmoid fonksiyonu
- Hiperbolik tanjant fonksiyonu

- ReLU fonksiyonu
- Leaky ReLU fonksiyonu
- Swish fonksiyonu
- Softmax fonksiyonu

1.5.1. Birim basamak fonksiyonu

Bir aktivasyon fonksiyonunda, dikkate alınması gereken en önemli şey eşik tabanlı sınıflandırıcıdır. Başka bir deyişle doğrusal dönüşümün değerinin nöronu aktive etmesi gerekip gerekmediğini veya aktivasyon fonksiyonuna girdinin daha büyük olması durumunda bir nöronun aktif hale geldiğini söylenebilir. Bir eşik değeri yoksa devre dışı bırakılır. Bu durumda çıktı bir sonraki katmana girdi olarak beslenmez (Sharma ve ark., 2017). Şekil 1.6'da basamak fonksiyonunun grafiği gösterilmektedir (Kızrak, 2019).



Şekil 1.6. Birim basamak fonksiyonu grafiği

Birim basamak fonksiyonu, var olan en basit etkinleştirme işlevi olarak Python'da basit if-else ifadeleriyle uygulanabilir. İkili sınıflandırıcı oluşturulurken genellikle birim basamak fonksiyonu kullanılır. Ancak, hedef taşıyıcıda çok sınıflı sınıflandırma durumunda birim basamak fonksiyon işlevi kullanılmaz. Ayrıca birim basamak fonksiyonunun gradyanı sıfırdır ve bu da geri yayılım adımında bir engelle neden olabilir. Yani $f(x)$ 'in x 'e göre türevini hesaplırsak sıfıra eşit olur (Sharma ve ark., 2017). Birim basamak fonksiyonu matematiksel olarak Denklem 1.1 ve Denklem 1.2'deki gibi tanımlanabilir.

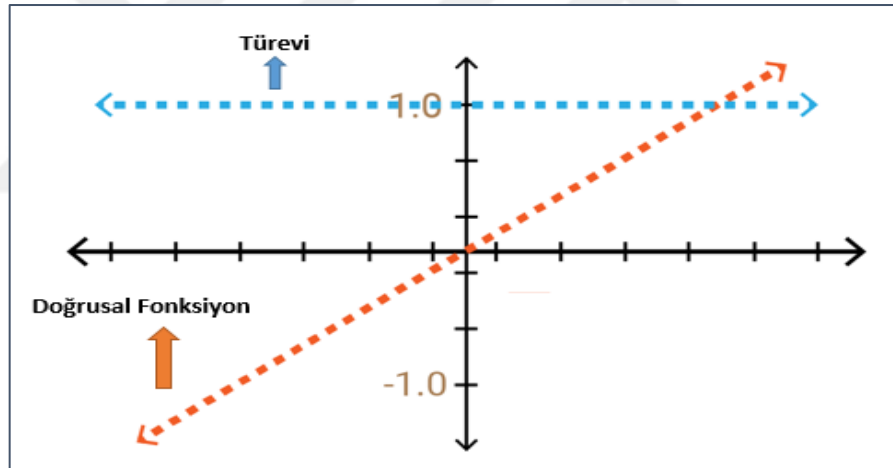
$$f(x) = 1, x \geq 0 \quad (1.1)$$

$$f(x) = 0, x < 0 \quad (1.2)$$

1.5.2. Doğrusal fonksiyonu

Doğrusal fonksiyonu, giriş ile doğru orantılıdır. Birim basamak fonksiyonunun dezavantajı sıfır gradyanına sahip olmasıdır. Çünkü birim basamak fonksiyonunda x'in hiçbir bileşeni yoktur. Bunu ortadan kaldırmak için doğrusal fonksiyon kullanılabilir (Sharma ve ark., 2017). Şekil 1.7'de doğrusal fonksiyonun grafiği gösterilmektedir (Kızrak, 2019). Doğrusal fonksiyonu Denklem 1.3'teki gibi ifade edilebilir.

$$f(x) = ax \quad (1.3)$$



Şekil 1.7. Doğrusal fonksiyonu grafiği

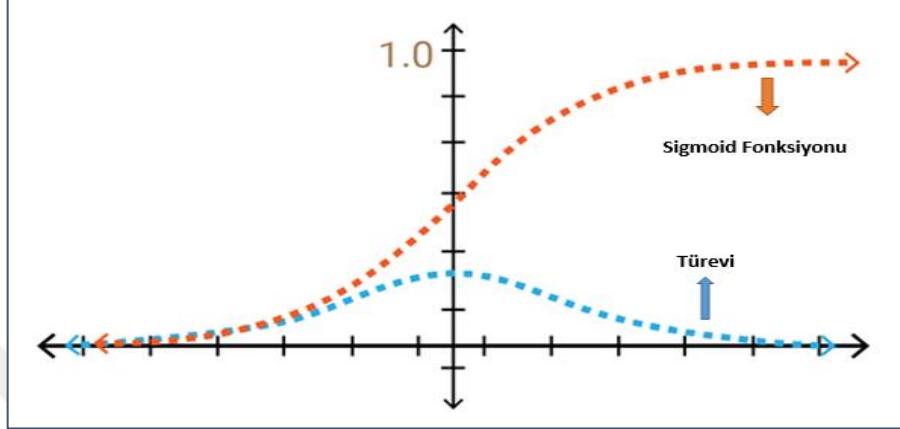
1.5.3. Sigmoid fonksiyonu

Doğrusal olmayan bir fonksiyon olduğu için en çok kullanılan aktivasyon fonksiyonlardan biridir. Sigmoid işlevi, 0 ile 1 aralığındaki değerleri dönüştürür. (Sharma ve ark., 2017). Şekil 1.8'de Sigmoid fonksiyonu grafiği gösterilmektedir (Kızrak, 2019). Sigmoid fonksiyonu Denklem 1.4'teki gibi ifade edilebilir.

$$f(x) = \frac{1}{1 + e^{-x}} \quad (1.4)$$

Sigmoid fonksiyonu sürekli türevlenebilir ve düzgün S şeklinde bir fonksiyondur. Fonksiyonun türevi Denklem 1.5'teki gibi ifade edilebilir.

$$f'(x) = 1 - \text{sigmoid}(x) \quad (1.5)$$



Şekil 1.8. Sigmoid fonksiyonu grafiği

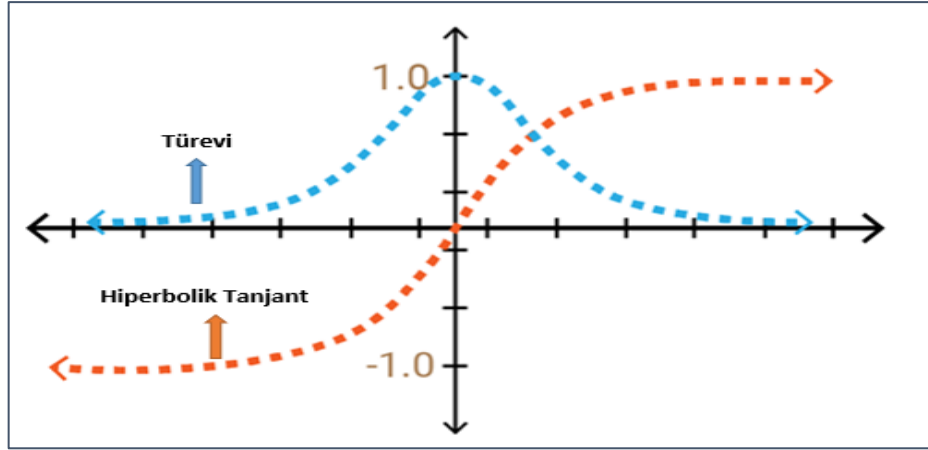
1.5.4. Hiperbolik tanjant fonksiyonu

Hiperbolik tanjant fonksiyonu sigmoid fonksiyonuna benzerken orijine göre simetriktir. Bir sonraki katmana girdi olarak beslenerek önceki katmanlardan farklı çıktı işaretleri ile sonuçlanır. (Sharma ve ark., 2017). Şekil 1.9'da hiperbolik tanjant fonksiyonunu grafiği gösterilmektedir (Kızrak, 2019). Hiperbolik tanjant fonksiyonunu Denklem 1.6'daki gibi ifade edilebilir:

$$f(x) = \tanh(x) = \frac{2}{1 + e^{-2x}} - 1 \quad (1.6)$$

Sigmoid fonksiyonuna benzediği için formülü Denklem 1.7'deki gibi yazılabilir.

$$f(x) = 2\text{sigmoid}(2x) - 1 \quad (1.7)$$

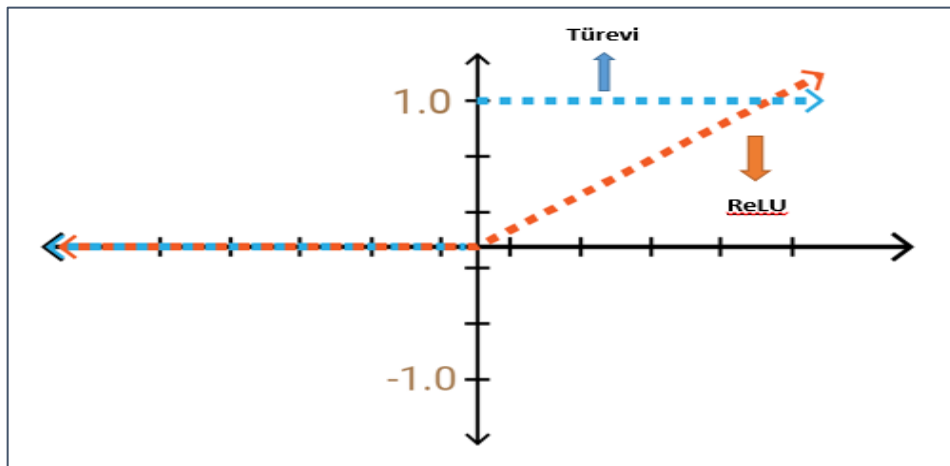


Şekil 1.9. Hiperbolik tanjant fonksiyonu grafiği

1.5.5. ReLU (Rectified Linear Unit) fonksiyonu

ReLU, rektifiye edilmiş liner birimi anlamına gelerek yapay sinir ağlarında yaygın olarak kullanılan doğrusal olmayan bir aktivasyon fonksiyonudur. ReLU fonksiyonunu kullanmanın avantajı, tüm nöronların aynı anda etkinleştirilmemesidir. Bir nöronun yalnızca doğrusal dönüşümün çıktısı sıfır olduğunda devre dışı bırakıldığı anlamına gelmektedir (Sharma ve ark., 2017). Şekil 1.10'da ReLU fonksiyonunun grafiği gösterilmektedir (Kızrak, 2019). ReLU fonksiyonu Matematiksel olarak Denklem 1.8'deki gibi ifade edilebilir.

$$f(x) = \max(0, x) \quad (1.8)$$



Şekil 1.10. ReLU fonksiyonu grafiği

1.5.6. Leaky ReLU fonksiyonu

Leaky ReLU, ReLU fonksiyonunun doğaçlama bir versiyonudur. x 'in negatif değerleri için ReLU fonksiyonlarının değerini sıfır olarak tanımlamak yerine, x 'in son derece küçük doğrusal bileşeni olarak tanımlanır (Sharma ve ark., 2017). Şekil 1.11'de Leaky ReLU fonksiyonunun grafiği gösterilmektedir (Kızrak, 2019). Leaky ReLU fonksiyonu Matematiksel olarak Denklem 1.9 ve Denklem 1.10'daki gibi ifade edilebilir.

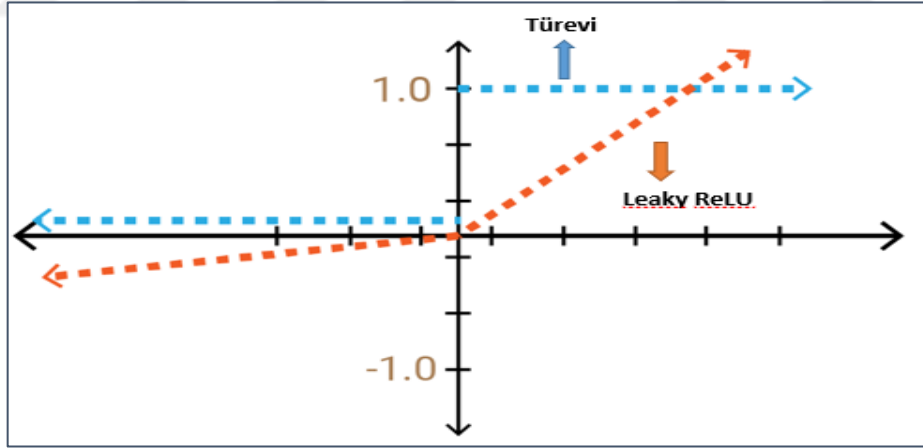
$$f(x) = 0.01x, x < 0 \quad (1.9)$$

$$f(x) = x, x \geq 0 \quad (1.10)$$

Ayrıca matematiksel olarak Denklem 1.11 ve Denklem 1.12'deki gibi de tanımlanabilir.

$$f(x) = x, x \geq 0 \quad (1.11)$$

$$f(x) = ax, x < 0 \quad (1.12)$$



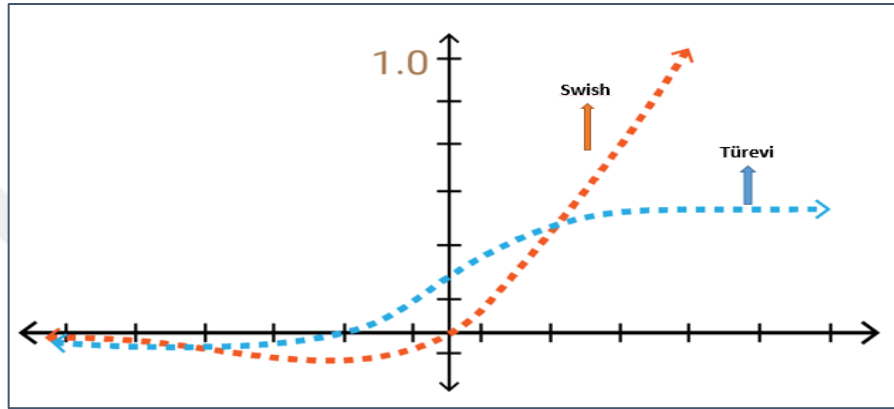
Şekil 1.11. Leaky ReLU fonksiyonu grafiği

1.5.7. Swish (A Self-Gated/Kendinden Geçitli) fonksiyonu

Swish fonksiyonu, hibrit bir aktivasyon fonksiyonu elde etmek için sigmoid fonksiyonu ve giriş fonksiyonunun kombinasyonu tarafından önerilen ilk bileşik aktivasyon fonksiyonlarından biridir (Nwankpa ve ark., 2018). Bununla birlikte Swish fonksiyonunun son yıllarda geliştirilmesi, kapsamlı ve pekiştirmeli öğrenmenin işbirliği

yapılan araştırma tekniklerinde iyi bir aktivasyon fonksiyonudur. CNN mimarisinde swish fonksiyonu ReLU üzerinden kullanılmasıyla görüntü sınıflandırmada iyileştirme sağlayabilir (Jinsakul ve ark., 2019). Şekil 1.12’de swish fonksiyonunun grafiği gösterilmektedir (Kızırak, 2019). Swish fonksiyonu matematiksel olarak Denklem 1.12’deki gibi ifade edilebilir.

$$f(x) = x \cdot \text{sigmoid}(x) = \frac{x}{1+e^{-x}} \quad (1,29) \quad (1.13)$$



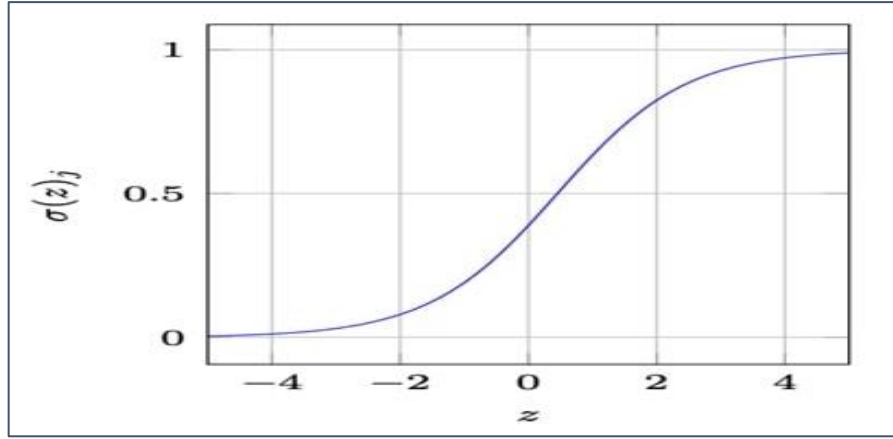
Şekil 1.12. Swish fonksiyonu grafiği

1.5.8. Softmax fonksiyonu

Softmax fonksiyonu, çoklu sigmoid fonksiyonlarının bir birleşimidir. Bir sigmoid fonksiyonunun 0 ile 1 aralığında değerler döndürüldüğünden dolayı belirli bir sınıfın veri noktalarının olasılıkları olarak ele alınabilir. Softmax fonksiyonu, ikili sınıflandırma için kullanılan sigmoid fonksiyonlarından farklı olarak çok sınıflı sınıflandırma problemlerinde kullanılabilir (Sharma ve ark., 2017). Softmax fonksiyonu Denklem 1.14’teki gibi ifade edilebilir:

$$G(z)_j = \frac{e^{z_j}}{\sum_{k=1}^K e^{z_k}} \quad j=1 \text{ için}, \dots, K \quad (1.14)$$

Burada z nörona girdi olan vektör ve k çok sınıflı sınıflandırma problemindeki sınıf sayısıdır. Şekil 1.13’de Softmax aktivasyon fonksiyonunun grafiği gösterilmektedir (Çağlar, 2022).



Şekil 1.13. Softmax fonksiyonu grafiği

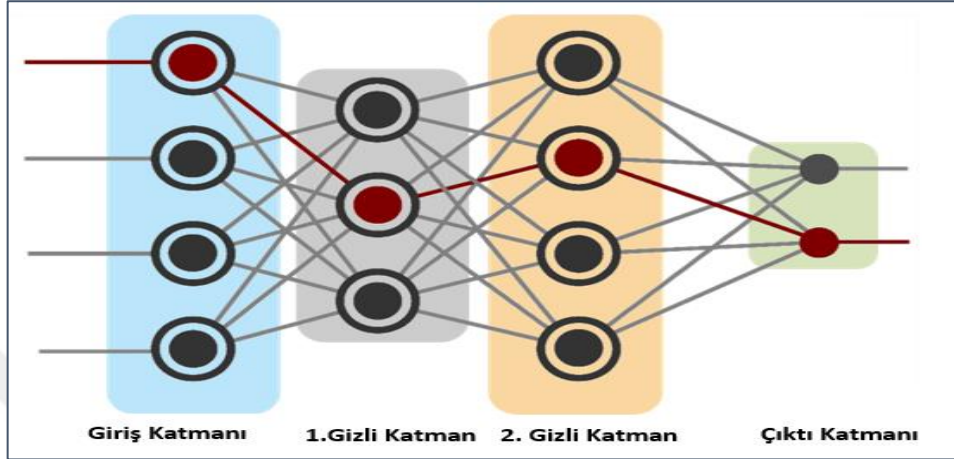
1.6. Yapay Sinir Ağlarının Sınıflandırılması

Yapay sinir ağları genellikle geri beslemeli ağlar ve ileri beslemeli ağlar olarak iki sınıfta incelenmektedir (Ataseven, 2013). Geri beslemeli sinir ağı, segmentasyonu kaldırma ve örüntü tanıma gibi görevlere uygulanabilir. Geri beslemeli sinir ağı uygulama alanları arasında matematiksel kanıtlar, sismik veri uydurma, tıp, bilim, mühendislik, sınıflandırma, fonksiyon tahmini ve zaman serisi tahmini vb. bulunur. Geri beslemeli sinir ağlarında veya geri yayılımında, düğümler arasındaki bağlantılar sırayla koordineli bir grafik üretir. Bu grafik, bir zaman dizisinde dinamik davranışı göstermek için sinir ağlarının geri beslemesine izin verir.

İleri beslemeli bir sinir ağı, insan nöron işleme birimlerine benzer katmanlar halinde organize edilmiş bir makine öğrenimi sınıflandırma algoritmasıdır. İleri beslemeli bir katmandaki her birim, katmanlardaki diğer tüm birimlerle ilişkilidir. Bu katmanların birimlerle olan bağlantıları eşit değilken her bağlantı farklı bir ağırlık veya kuvvete sahip olabilir. Ağ bağlantılarının ağırlıkları, ağ bilgisinin potansiyel miktarını ölçer. Ayrıca, sinir ağı birimleri düğümler olarak bilinir. Ağdaki bilgi işleme, giriş birimlerinden veri girişini içerir ve ağdan geçerek çıkış birimlerine ulaşana kadar bir katmandan diğerine akar. İleri beslemeli bilgi yalnızca bir yönde akarak giriş düğümlerinden varsa gizli düğümlere ve ardından çıkış düğümlerine iletilir. İleri beslemeli sinir ağları tek katmanlı ve çok katmanlı algılayıcılardır (Abiodun ve ark., 2018).

Çok katmanlı algılayıcılar (MLP), girdi için boyutların sayısı ve çıktı için boyutların sayısı olan bir veri kümesi üzerinde eğitim vererek bir fonksiyonu öğrenen deneysel bir öğrenme algoritmasıdır. Bir dizi özellik ve bir hedef verildiğinde, sınıflandırma

veya regresyon için doğrusal olmayan bir fonksiyon tahminini öğrenebilir. Lojistik regresyondan farklı olarak girdi ve çıktı katmanları arasında gizli katmanlar olarak adlandırılan bir veya daha fazla doğrusal olmayan katman olabilir (Anonymous, 2022). Şekil 1.14’de MLP yapısı gösterilmektedir (Anonim, 2018).



Şekil 1.14. MLP yapısı

1.7. Derin Öğrenme Platformları

1.7.1. Torch

Torch, C'deki bir uygulama ile LUA dili kullanılarak geliştirilirken Torch'un Python uygulamasına PyTorch denmektedir.

1.7.2. TensorFlow

TensorFlow, makine öğrenimi için uçtan uca açık kaynaklı bir kütüphanedir. Araştırmacıların yeni makine öğrenimi teknolojilerinden ve geliştiricilerin kolayca oluşturulup geliştirmelerine olanak tanıyan kapsamlı ve esnek bir kitaplık ve topluluk kaynağı ekosistemine sahiptir. Google'ın TensorFlow'u şu anda dünyanın en popüler öğrenme kitaplığıdır. Vektörler veya n boyutlu matrisler olan tensör kavramına dayanmaktadır.

1.7.3. Keras

Keras, Python ile yazılan TensorFlow üzerinde çalışan bir derin öğrenme uygulamasıdır. Hızlı denemeye izin vermek amacıyla geliştirilerek fikirden sonuca mümkün olduğunca çabuk giden iyi bir araştırmamanın anahtarlarından biridir.

1.7.4. DL4J

Java için Derin Öğrenme (DL4J), Java ve Scala için yazılmış ilk derin öğrenme kitaplığıdır. Hadoop ve Apache Spark ile koordineli çalışmaktadır (Anonymous, 2022).

1.8. Derin Öğrenmede Kullanılan Algoritma Türleri

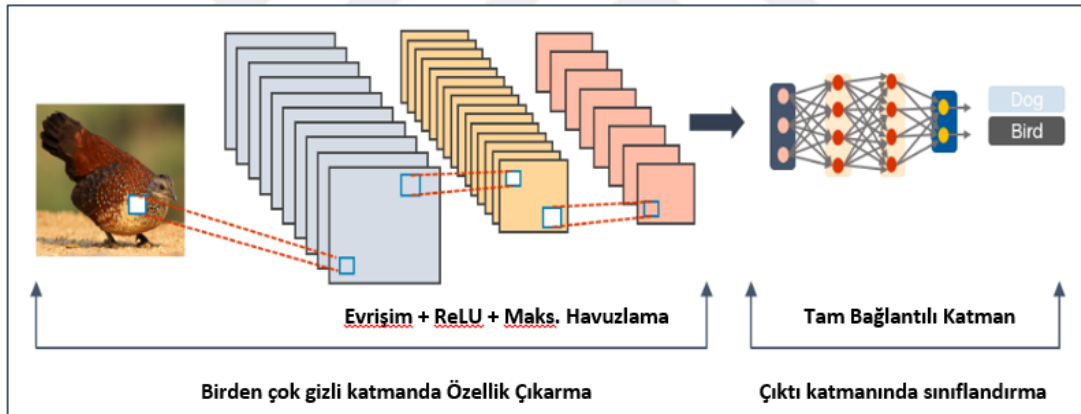
Gelişen teknoloji sayesinde derin öğrenme algoritmalarının sayısı her geçen gün artmaktadır (Anonymous, 2022). Bu algoritmalar:

- Evrişimli Sinir Ağları (Convolutional Neural Networks - CNNs)
- Uzun Kısa Süreli Bellek Ağları (Long Short Term Memory Networks - LSTMs)
- Tekrarlayan Sinir Ağları (Recurrent Neural Networks - RNNs)
- Üretken Düşman Ağları (Generative Adversarial Networks - GANs)
- Radyal Temel Fonksiyon Ağları (Radial Basis Function Networks - RBFNs)
- Çok Katmanlı Algılayıcılar (Multilayer Perceptrons - MLPs)
- Kendi Kendini Düzenleyen Haritalar (Self Organizing Maps - SOMs)
- Derin İnanç Ağları (Deep Belief Networks - DBNs)
- Kısıtlı Boltzmann Makinaları (Restricted Boltzmann Machines - RBM)
- Otomatik kodlayıcılar

1.8.1. Evrişimli sinir ağları (CNN)

Evrişimli sinir ağları (CNN), temel olarak görüntü tanıma için kullanılan ancak bununla sınırlı olmayan son teknoloji makine öğrenme algoritmalarıdır. Diğer sinir ağları çalışmalarının aksine verileri sırayla yönlendiren ve dönüştüren özellikle evrişimli ve havuzlama katmanları olmak üzere bir dizi farklı katman türünü birleştirmektedir.

Şekil 1.15'te CNN yapısı gösterilmektedir (Anonymous, 2022). İlk evrişim katmanları, kenarlar veya köşeler gibi küçük temel özellikleri algılayarak yeni kanallar eklemektedir. Havuzlama katmanları çözünürlüğü azaltırken kanal sayısına değiştirmez. Sonraki evrişim katmanlarının daha önce tespit edilen temel özellikleri veri kümesine özgü daha karmaşık daha büyük özelliklerle yeniden birleştirmesine olanak tanımaktadır. Evrişimli katmanları filtreleri öğrenilebilir ağırlık matrislerini kullanarak filtrelerin özelliklerini tanıırken havuzlama katmanları da çekirdeklerinde bir ortalama veya maksimum işlevini kullanmaktadır. Hem filtreler hem de çekirdekler her seferinde adım adım dönen bir pencere olarak giriş üzerinde hareket ettirilir. Girdi verileri, azalan çözünürlüklere karşı koymak için sıfırlarla doldurulabilir. Özellik haritalarını giriş üzerinde taşıyarak parametreleri yeniden kullanarak ağırlıkların sayısını azaltmaktadır. Daha derin topolojilere izin vererek konumlarından bağımsız özellikleri tespit etmektedir. CNN'nin özellik tanıma bölümünü, istenen çıktıyı yaymak için bir sınıflandırıcı, klasik olarak bir tam bağlantılı ağ ve softmax sınıflandırıcı izlemektedir (Strauch, 2019).



Şekil 1.15. CNN yapısı ve katmanları

1.9. Segmentasyon İçin Derin Öğrenme

Derin öğrenme, son yıllarda görüntü segmentasyonu için en yaygın kullanılan yaklaşımdır. Buna karşılık DÖ tabanlı algoritmalar, nesne algılama ve segmentasyon için verilerden karmaşık özellikleri otomatik olarak keşfetmede iyi performans göstermektedir. DÖ tabanlı algoritmaların diğer görüntü analizi uygulamalarını kolaylaştırır. Gelişmiş bilgisayar donanımından ve eğitim için artan mevcut verilerden yararlanan DÖ tabanlı segmentasyon algoritmaları geleneksel yöntemlerden kademeli olarak daha iyi

performans gösterirken arařtırmalarda daha fazla popülerlik kazanmaktadır (Chen ve ark., 2020).

Görüntü Segmentasyonu, bir görüntüyü benzer özelliklere sahip pikseller içeren ayrı ve farklı bölgelere ayırma işlemidir. Görüntü segmentasyonu sahne anlama, tıbbi görüntü analizi, robotik algı, video gözetimi, artırılmış gerçeklik ve görüntü sıkıştırma gibi uygulamalarla görüntü işleme ve bilgisayarla görmede önemli bir çalışma alanıdır. Son zamanlarda, derin öğrenme modellerinin çeşitli uygulamalarındaki başarısı nedeniyle, derin öğrenme modellerini kullanarak görüntü segmentasyonu yaklaşımlarını geliřtirmeye yönelik önemli miktarda çalışma yapılmaktadır (Minaee ve ark. 2021).

1.9.1. Eşik

Eşik, segmentasyonun en basit ve yaygın yöntemidir. Görüntü önce gri ölçekli forma getirilerek ve T eřiğı kullanılarak ikili görüntüye dönüřtürölür (Labs, 2018). Daha sonra bu T'nin altındaki tüm gri seviye deęerleri siyah (0) olarak sınıflandırılarak T'nin üzerindeki beyaz (1) olacaktır (Kumar, M. J., ve ark., 2014). Eřiik deęerini Denklem 1.15'teki gibi ifade edilebilir.

$$T=T [i, j, p (i, j), f (i, j)] \quad (1.15)$$

Burada: T, eřiik deęeridir. i, j, eřiik deęer noktasının koordinatlarıdır. p (i, j), f(i,j) gri seviyeli görüntü piksellerinin noktalarıdır.

1.9.2. Kenar bazlı segmentasyon

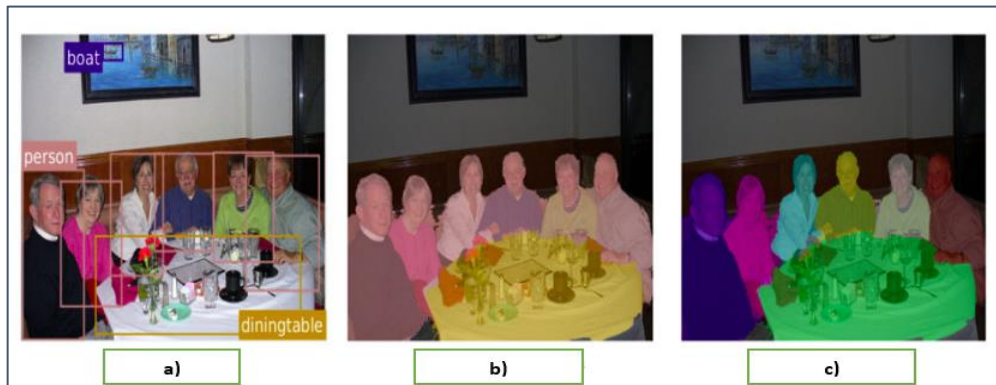
Kenar, birçok görüntü işleme uygulamasında çok önemli bir rol oynamaktadır. Nesnenin bir taslağını saęlarlar. Kenar, gri deęeri farklı olan iki bölge arasındaki sınırda uzanan bir dizi baęlantılı pikseldir. Kenardaki bu piksellere kenar noktaları denir. Bir kenar, tipik olarak, görüntü fonksiyonunun türevinin hesaplanmasıyla çıkarılır. Kenar algılama teknięi, görüntü segmentasyon teknięinin yapısal tekniklerinden biridir (Kumar, M. J., ve ark., 2014).

1.9.3. Bölge bazlı segmentasyon

Bir görüntünün R ile gösterilen bir bölgesi, gri seviye veya doku gibi bazı kriterlere göre görüntünün bağlantılı homojen bir alt kümesi olarak tanımlanır. Bir görüntüdeki bölgeler, benzer özelliklere sahip bir grup bağlantılı pikseldir. Bölge bazlı yaklaşımda, her piksel belirli bir nesneye veya bölgeye atanır. Bölge bazlı segmentasyon algoritmaları, kenar bazlı segmentasyon yöntemine kıyasla nispeten basittir. Kenar bazlı yöntemler, bir görüntüyü kenarlara yakın yoğunluktaki hızlı değişikliklere dayalı olarak bölümlere ayırırken, bölge bazlı yöntemler, bir görüntüyü önceden tanımlanmış bir dizi kritere göre benzer bölgelere ayırır. Bölge bazlı segmentasyonda bir nesneye karşılık gelen pikseller birlikte gruplandırılır ve işaretlenir. Bölge bazlı segmentasyon ayrıca uygun eşikleme tekniklerinin kullanılmasını gerektirir (Kumar, M. J., ve ark., 2014).

1.10. Semantik segmentasyon

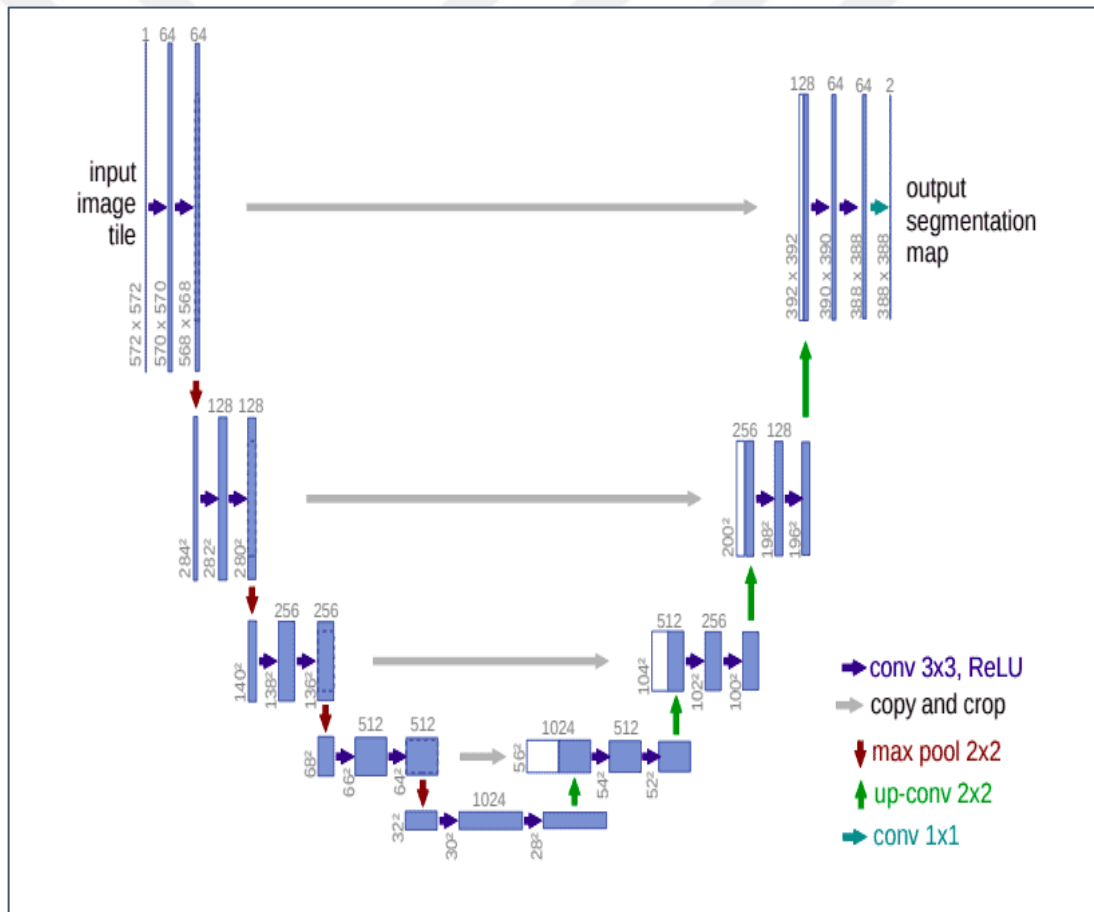
Bir görüntüdeki her pikseli bir sınıf etiketine bağlama işlemine Semantik segmentasyon denmektedir. Etiket ise kedi, çiçek, aslan vb. olabilir. Semantik segmentasyon, piksel düzeyinde görüntü sınıflandırması olarak düşünülebilir. Bu nedenle Semantik segmentasyonda görüntünün her pikseli belirli bir sınıf etiketi ile ilişkilendirilmek zorundadır. Şekil 1.16'da semantik segmentasyonun aşamaları gösterilmektedir. Altı kişilik bir imajın olduğu bir örnekte nesne algılayan altı kişiyi tanımlayan ve etraflarında sınırlayıcı kutular oluşturarak onlara tek bir kişi etiketi vermektir. Semantik segmentasyon daha da geliştirilerek tanımlanan her kişi üzerinde bir maske oluşturarak hepsine tek bir kişi etiketi vermektir (Mwiti, 2022).



Şekil 1.16. a) Nesne algılama b) Semantik segmentasyon c) Örnek segmentasyon

1.10.1. U-Net

U-Net mimarisi, aşağı örnekleme daraltma yolu ve yukarı örnekleme genişletme yoluna sahip U şeklinde, simetrik evrişimli bir ağıdır. Ortaya çıkan parçalı çıktı görüntüsünün boyutu, ham girdi görüntüsünden çok daha küçüktür (Shah, 2021). U-net mimarisi Şekil 1.17’de U-Net mimarisi gösterilmektedir (Mwiti, 2022). Her mavi kutu, çok kanallı bir özellik haritasına karşılık gelmektedir. Kanal sayısı kutunun üst kısmında belirtilmiştir. X-y boyutu kutunun sol alt kenarında verilmiştir. Beyaz kutular kopyalanan özellik haritalarını temsil ederken oklar farklı işlemleri göstermektedir (Xiancheng ve ark., 2018).



Şekil 1.17. U-Net mimarisi

Mimari, “U” harfinin sol kısmına kodlayıcı, sağ kısmına kod çözücü adı verilen iki yoldan oluşmaktadır. Kodlayıcı, girdiyi alan ve bir özellik haritası, vektör ve tensör çıktısı veren bir ağ içermektedir. Özellik vektörü, girişi temsil eden bilgileri, özellikleri

tutarken kod çözücü aynı zamanda kodlayıcıdan özellik vektörünü alan ve gerçek girdiye veya amaçlanan çıktıya en yakın eşleşmeyi veren bir ağıdır. Genellikle kodlayıcı ile aynı ağ yapısına sahip ancak zıt yönlüdür (Pallawi, 2019).

U-net avantajları

- Esnektir ve herhangi bir rasyonel görüntü maskeleye görevi için kullanılabilir.
- Yüksek doğruluk sağlar.
- Tamamen bağlı katmanlar içermez.
- Görüntüleri hızlı ayıklar.
- Sınırlı veri içeren senaryolarda çok güçlü segmentasyon aracı olduğu kanıtlanmıştır.
- Farklı biyomedikal segmentasyon uygulamalarında çok iyi performanslar elde etmeyi başarır.

U-net dezavantajları

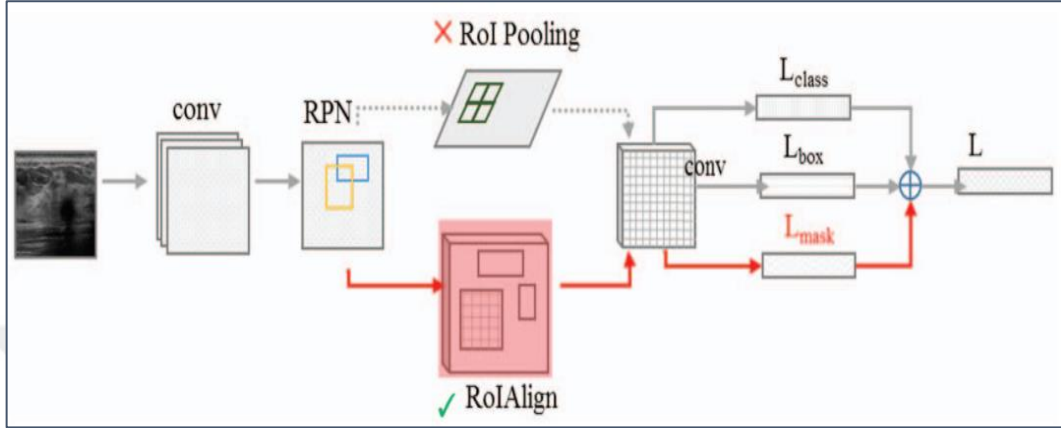
- Daha büyük görüntüler için yüksek GPU belleği gerekir.
- Çok katmanda eğitimi önemli miktarda zaman alır.
- Önceden eğitilmiş modeller yaygın olarak bulunmaz (Ronneberger ve ark., 2015).

1.10.2. Mask R-CNN

Faster R-CNN iki aşamadan oluşur. Bölge öneri ağı (RPN) olarak adlandırılan ilk aşama, aday nesne sınırlama kutuları önerir. Özünde Faster R-CNN olan ikinci aşama, RoIPool kullanarak her bir aday kutudan öznelikler çıkarır ve sınıflandırma ve sınırlayıcı kutu regresyonu gerçekleştirir. Her iki aşama tarafından kullanılan özellikler daha hızlı çıkarım için paylaşılabilir (He ve ark., 2017).

Şekil 1.18'de Mask R-CNN mimari yapısı gösterilmektedir (Chiao ve ark., 2019). Mask R-CNN, sınırlayıcı kutu tanıma için mevcut dal ile paralel olarak bir nesne maskesini tahmin etmek için ek bir dalın dahil edilmesiyle Faster R-CNN'den farklıdır. Diğer bir önemli fark, Mask R-CNN'nin, Şekil 1.18'de gösterildiği gibi, Faster R-CNN'de niceleme hatasına eğilimli RoIPool katmanını değiştirmek için RoIAalign adı verilen daha doğru bir özellik çıkarma katmanını kullanmasıdır. CNN'nin tasarımı, sınıflandırma ile sınırlıdır ve segmentasyon amaçlı değildir. Bu nedenle, özellik çıkarımı için kaba uzamsal niceleme gerçekleştirmek için RoIPool kullandığından, ağ girişleri ve

çıkışları arasında pikselden piksele hizalama için verimli değildir. Bu nedenle, Mask R-CNN, RoIPool'u, herhangi bir yanlış hizalamanın üstesinden gelmek için tam uzamsal konumları sadık bir şekilde koruyan, RoIAlign adı verilen basit, niceleme içermeyen bir katmanla değiştirir (Adeola, 2020).



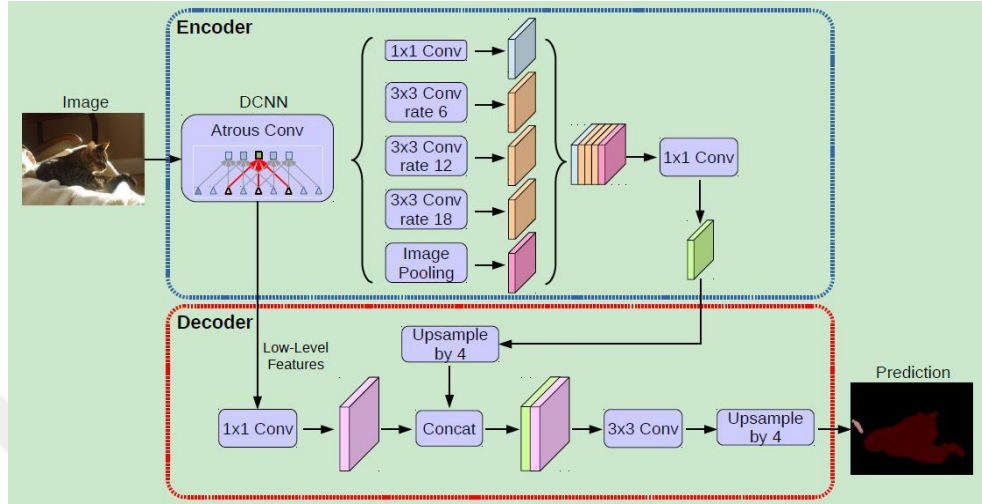
Şekil 1.18. Mask R-CNN mimarisi

Mask R-CNN modeli, görüntü sınıfı için sınırlayıcı kutu segmentasyon modülüne paralel olarak bir nesne maskesi tahmin modülünü içeren nesne eş görünümü düzeyinde tanıma için genel bir çerçevedir. İyileştirilmiş sınırlayıcı kutu, çıktı sınırlayıcı maskeyi iyileştirmek için önceki bilgileri kullanır. Bölümlere ayrılmış çıktı, sınırlayıcı kutuya ek olarak maskenin tahminidir. Maske hedefi, bir ilgilenilen bölge (RoI) ile bununla ilişkili yer gerçeği maskesi arasındaki kesişme noktasıdır. Bir RoI, en az 0,5'lik bir temel gerçeklik kutusuyla kesişim açık birleşimine (IoU) sahipse pozitif olarak kabul edilir, aksi takdirde negatif olarak kabul edilir. "Maske kaybı" yalnızca pozitif ROI'lerde tanımlanır (Adeola, 2020). Mask R-CNN'deki kayıp fonksiyonu L , $L_{class} + L_{box} + L_{mask}$ değeri minimize edilmiş ve sinir ağı modeli olarak eğitim verisi üzerinde kayıp fonksiyonunun minimize edilmesi yoluyla en uygun modeli kullanır (Chiao ve ark., 2019).

1.10.3. DeepLab

DeepLab yöntemi, derin evrişimli sinir ağları (DCNN) ve tamamen bağlı bir koşullu rastgele alan (CRF) kombinasyonunu kullanır. DCNN'yi anlamsal segmentasyona uygularken karşılaşılan üç ana zorluğun üstesinden gelir: (1) azaltılmış özellik

çözünürlüğü, (2) birden çok ölçekte nesnelerin varlığı ve (3) DCNN değişmezliğinden kaynaklanan azaltılmış yerelleştirme doğruluğudur. Şekil 1.19’da DeepLab mimarisi gösterilmektedir (Anonymous, 2022).



Şekil 1.19. DeepLab mimarisi

Model atröz evrişimler kullanır; örneklenmiş filtreler ile evrişimlerdir. Atröz evrişimler parametre sayısını veya hesaplama miktarını artırmaya gerek kalmadan filtre alanının genişletilmesini sağlamaktadır Birden fazla ölçekte nesnelerin varlığı, atröz uzamsal piramit havuzu (ASPP) kullanılarak çözülür.

DeepLab yönteminde, bir özellik katmanı, evrişim gerçekleşmeden önce birden çok oranda yeniden örneklenir. Bu, görüntü bağlamının birden çok ölçekte yakalanmasıyla sonuçlanır. Son sorun (DCNN değişmezliğinden kaynaklanan azaltılmış yerelleştirme doğruluğu), tam bağlı bir CRF kullanılarak modelin ince ayrıntıları yakalama yeteneği geliştirilerek çözülür. Etkili hesaplamanın yanı sıra ince kenar ayrıntılarını yakalama yeteneği nedeniyle tam bağlantılı bir ikili CRF kullanılır (Chen ve ark., 2017).

2. KAYNAK ARAŞTIRMASI

Abu-Mahfouz (2005) yaptığı çalışmada kararlı durum koşullarında çalışan bir dişli kutusunun arıza, aşınma ve hasarının tespiti ve sınıflandırılması için üç YSA mimarisi ile eğitim yapmıştır. İleri beslemeli geri yayılım (FFBP), fonksiyonel bağlantı ağı (FLN) ve öğrenen vektör parçalama (LVQ) uygulanmasının sonuçlarına genel bir bakış sunmuştur. Bunların dışında şanzımana kasıtlı olarak gevşek anahtar, tek diş yan aşınması ve tam diş kırılması (eksik diş) gibi üç yapay kusur eklenerek kapsamlı deneylerden toplanan titreşim sinyalleri, YSA'ları beslemek için özellik vektörleri olarak kullanılan zaman ve frekans alanı tanımlayıcıları kullanılarak analiz edilmiştir. Bu çalışma için FLN'nin daha hızlı öğrendiğini ve operasyonda FFBP veya LVQ'dan daha doğru olduğunu göstermektedir. LVQ algoritması, FFBP'den daha hızlı yakınsama oranı sergiler ancak yanlış sınıflandırmalardan daha fazla zarar görmüştür.

Denil ve ark. (2013) yaptıkları çalışmada bazı derin öğrenme modellerinin değişkenlerinin atamasında önemli bir fazlalık olduğunu tespit etmiştir. Her özellik için verilen ağırlık değerlerinden, diğer kalan değerleri doğru tahmin edebildiğini göstermiştir. Ayrıca çalışmada diğer değişkenlerin hiç öğrenilmeden değerlerinin tahmin yoluyla bulunabileceğini ifade etmiştir. Eğitim ve test olmak üzere değişkenlerin öğretilmesiyle geri kalan mimarinin en iyi durumda %95 oranında tahmin edebildiğini göstermiştir.

Yamamoto ve ark. (2014) domates bitkisinin hasattan önce meyvelerinin durumuna bakılarak verimini tespit eden bir yöntem geliştirilerek sınıflandırılmıştır. RGB dijital kamera kullanarak bir bitki üzerinde olgun, olgunlaşmamış ve genç meyveler dahil olmak üzere tek tek bozulmamış domates meyvelerinin görüntüleri alınmıştır. Alınan görüntülerden sonra meyvelerin segmentasyonu yapılmıştır. Yapılan sonuçlara göre meyve tespiti geliştirilen yöntemle 0.80 geri çağırma elde ettiğini ve hassasiyetin ise 0.88 olduğunu göstermiştir.

Ronneberger ve ark. (2015) yaptıkları çalışmada U-Net mimarisini oluşturmuşlardır. Biyomedikal alanında başarıyla uygulanan model birçok çalışma için öncü olmuştur. Mikroskopi ile kaydedilen cam üzerindeki HeLa hücrelerini görüntülerini kaydederek ve segmentasyon uygulayarak maskelerini oluşturmuştur. Görüntüleri pikselleyerek segmentasyon tahmini yapmıştır.

Sladojevic ve ark. (2016) yaptıkları çalışmada görüntü sınıflandırmada CNN'ler büyük bir ilerleme kaydettiğini açıklamışlardır. Bitkilerin yaprak şekline bakılarak has-

ta-lıklarının tespit edilmesinde derin öğrenme metotlarının kullanılmasıyla daha verimli bit-kilerinin elde edilmesi dijital tarım açısından mümkün olmuştur. Çalışmada geliştirilen modelle bitki yapraklarını çevresinden ayırt etme yeteneği ile sağlıklı yapraklardan 13 farklı bitki hastalığını tanıyabilmektedir. Bitki hastalıklarının tanınması için bu yöntem ilk kez önerilmiştir. Görüntülerden hastalık tanıma modeli ile CNN kullanılarak Berkley Vision and Learning Centre tarafından geliştirilen Caffee derin öğrenme çerçevesi kullanılmıştır. Yapılan deneysel çalışmada yaprak hastalıklarını ayrı sınıf testleri için ortalama %91-%98 arasında doğru sonuçlar elde edilmiştir.

Kim ve Lim (2017) yaptıkları çalışmada dijital görüntülerin kalitesini insan odaklı bakış açısıyla değerlendirmek gerektiğini belirtmiştir. Çalışmada İnsan Görsel Sisteminin (HVS) davranışını görüntü kalitesi değerlendirmesi (IQA) veri tabanlarını kullanarak Derin görüntü kalitesi değerlendirmesi (DeepQA) adlı yeni bir CNN tabanlı, bir dizi tam referanslı görüntü kalitesi değerlendirme (FR-IQA) modeli önermişlerdir. Sonuç olarak deneyler yoluyla tahmin edilen görsel duyarlılık haritalarının insan öznel görüşleri ile uyumlu olduğunu göstermişlerdir. Çalışmanın sonunda DeepQA, FR-IQA modelleri arasında en gelişmiş tahmin doğruluğunu elde eden model olduğunun sonucuna varılmıştır.

Nasiri ve ark. (2017) yaptıkları çalışmada makine mühendisliğinde yaygın olarak kullanılan YZ yöntemlerinin mühendislik problemlerini çözmek için yeni teknikler ile çözülmesi ve geliştirilmesini incelemişlerdir. Makine mühendisliğinde birçok YZ tekniği ve yaklaşımı mevcut olmasına rağmen, kırılma mekaniği alanında mevcut çalışmaları, sistemleri ve uygulamaları gözden geçirmeyi amaçlayan herhangi bir çalışma bulunmamaktadır. Çalışmada kırılma mekaniği alanında kullanılan Bayes ağı, YSA, genetik algoritmalar (GA), bulanık mantık ve vaka tabanlı muhakeme gibi beş yapay zeka yönteminin son durumu incelenmiştir. Kırılma mekaniği bu beş YZ ile açıklamaya çalışılmıştır.

Wang ve ark. (2017) araştırmada, gıdaların otomatik şekilde özelliklerini belirlemek için derin öğrenme yöntemi kullanılarak görüntülerden bir sınıflandırma yapılmıştır. Çalışmada PlantVillage veri setindeki elma çürükleri görüntüleri kullanılmıştır. Hastalık ciddiyetini teşhis edebilmek için CNN tercih edilmiştir. Eğitilen ağda performans ve sistematik şekilde görüntüden hastalığın belli olması amaçlanmıştır. En iyi model olarak VGG16 mimarisi seçilerek veriler test edilmiştir. Sonuç olarak %90,4 oranında

doğruluk sağlandığı gözlemlenmiştir. Önerilen derin öğrenme modelinin, bitkilerin hastalık kontrolünde modern tarım için bir öncülük yapması istenmiştir.

Çoşkun ve ark. (2017) yaptıkları çalışmada DÖ ile ilgili kavramlar üzerinde çalışmışlardır. Makine Öğrenimindeki en son yaklaşımlardan biri olan derin öğrenmenin temel kavramlarına tarihçesi ve güncel uygulamaları açısından genel bir bakış sunulmuştur. Derin öğrenme, el yazısı tanıma, görüntü tanıma, nesne algılama vb. gibi birçok alanda büyük başarılar göstermiştir. CNN, RNN, RBM ve otomatik kodlayıcılar gibi tipik derin öğrenme algoritmalarının kavramları ve mekanizmaları açıklanmıştır.

Kamilaris ve ark. (2018) yaptıkları çalışmada, DÖ tekniklerini kullanan 40 araştırma çalışması değerlendirilmiştir. DÖ ile tarım alanında yaşanabilecek sorunları ortadan kaldıracak ifade edilmiştir. DÖ ile tarımsal ürünlerin veri analizinin yapılabileceği gösterilmiştir. Son olarak derin öğrenmenin yüksek doğruluk sağladığını ve yaygın olarak kullanılan mevcut görüntü işleme tekniklerinden daha iyi performans gösterdiği ifade edilmiştir.

Ferentinos, (2018) yaptığı çalışmada sağlıklı ve hastalıklı bitkilerin basit yaprak görüntüleri kullanılarak bitki hastalık tespiti ve teşhisinde CNN modeli kullanılmıştır. Modellerin eğitimi için 58 farklı bitkiye ait yaprak ve hastalık verilerinden yararlanılmıştır. Modelin en hastalıklı bitkiyi tespit etmede neredeyse %100 yakın performans gösterdiği sonucuna varılmıştır.

Adhikari ve ark. (2018) çalışmada tarım sektörünün en büyük sorunlarından olan bitki hastalıklarının belirlenmesi için çözüm bulma amaçlanmıştır. Hastalıkların tespit edilmesi ve önlenmesi tarımla uğraşanlar için hayati önemdedir. Bu çalışmada domates bitkisi kullanılarak bitki hastalıkları otomatik olarak sınıflandırma ve tespit etmek amaçlanmıştır. OpenCV görüntü işleme kullanılarak CNN modeli ile hastalık türlerini tahmin eden bir model oluşturulmuştur. Veriler çevrimiçi bir kaynaktan toplanmıştır. Domates bitkilerinde görülen birkaç hastalık tespit edilmiştir.

Koirala ve ark. (2019) yaptıkları çalışmada meyve tespiti konusunda gelecekte yapılabilecek çalışmalar için alternatif kaynaklar sunulmuştur. DÖ görüntü işleme tekniklerinin yeni kullanıcılarına yardımcı olmak ve yön vermek için çeşitli yollar önerilmiştir. Meyvelerin görüntülerinden ağırlık veya ağaç maksimum taşıma kapasitesinin belirlenmesi hakkında halka açık kaynakların olması savunulmuştur.

Arsenoviç ve ark. (2019) yaptıkları çalışmada yaprak görüntülerini içeren en büyük veri setlerinden biri olacak yeni bir veri seti oluşturulmuştur. Görüntüler, çeşitli

hava koşullarında, farklı açılarda ve gündüz saatlerindeki gibi farklı arka planla çekilmiştir. Veri setindeki görüntü sayılarını artırmak için geleneksel büyütme yöntemleri ve GAN kullanılmıştır. Bitki hastalıklarını karmaşık bir arka planda ve tek bir yaprakta birden fazla hastalığın tespitinde kontrollü bir ortamda eğitimin ve gerçek yaşam durumlarında kullanımın etkisi test edilmiştir. Gerçek bir çevreye odaklanan bitki hastalıkları sınıflandırması için bir sinir ağının yeni iki aşamalı mimarisi önerilmiştir. Eğitilen model %93,67'lik bir doğruluk elde etmiştir.

Salem ve ark. (2019) yaptıkları çalışmada bitki hastalıkları teşhisinin tarım alanında bitki büyümesini erken tespit etmesine yardımcı olduğu ifade edilmiştir. DÖ mimarileri zamanla gelişmeye başladığından, araştırmacılar tarafından görüntü tanıma ve sınıflandırmada ya uygulanmıştır. Görüntü algılama, segmentasyon ve sınıflandırma için AlexNet'in sunulmasından sonra birçok son teknoloji DÖ modeli kullanılmıştır.

Verma ve ark. (2019) yaptıkları çalışmada bitki hastalıklarının düzgün sınıflandırılması ve tanımlanmasının, tahmin modellerinde hayati öneme sahip olduğu ifade edilmiştir. Görüntülerden bitki hastalıklarının tanımlanabileceğini belirten çalışmada görüntü sınıflandırma uygulamalarından CNN kullanılmıştır. Bu çalışmada PlantVillage veri setini kullanarak domates yaprağı görüntüleri alınmıştır. Veri setinde Inception ResNet v2, VGG16, VGG19, ResNet50 ve Xception olmak üzere beş CNN modeli uygulanarak eğitilmiştir. Beş modeli karşılaştırıp en iyi performansı gösteren modelin ResNet50 olduğu sonucuna varılmıştır. Domates bitki hastalıklarını sınıflandırıp başarıyla açıklayan bir uygulama geliştirilmiştir.

Buayai ve ark. (2019) yaptıkları çalışmada domuz ağırlıklarını otomatik bir şekilde görüntülerinden tahmin edilmesi incelenmiştir. Geleneksel yöntemlerle tartımlar yapılan domuzların ağırlıklarında oluşan belirsizliği ortadan kaldırmak için Makine Öğrenimi yöntemi kullanılmıştır. Oluşturulan bir sistemle domuzların hareketleri görüntüler aracılığıyla kaydedilmiştir. Geliştirilen bir yazılımla, kamera kalibrasyonu, özellik çıkarma işlemi için domuz vücut sınırı tespiti ve domuz ağırlığı tahmini için yapay sinir ağı ile eğitilmiştir. Deneysel sonuçlara göre %2.84'lük düşük bir mutlak ortalama hataya, yaklaşık 5 saniyelik kısa bir model eğitim süresine ve 7.18 milisaniyelik kısa bir tahmin süresine sahip yaklaşımın faydaları ve pratikliği gösterilmiştir.

Ibtehaz ve Rahman (2020) yaptıkları çalışmada klasik U-Net modeli yerine daha gelişmiş bir model üzerinde çalışılmıştır. Geliştirilen yeni mimari MultiResUnet ile tıbbi görüntüler eğitilmiştir. İki mimari karşılaştırılıp değerlendirilmiştir. Beş farklı veri

kümesi ile çalışılmıştır. Performans yüzdeleri %10,15, %5,07, %2,63, %1,41 ve %0,62 oranında bulunulmuştur. MultiResUnet modelinin klasik U-Net modeline göre daha başarılı sonuçlar verdiği ifade edilmiştir. Tıbbi görüntü segmentasyonunda derin öğrenme modellerinin başarılı çalışmaları hastalık tespit ve önlemede öncü rol oynayacağı öngörülmüştür.

Singh ve ark. (2020) yaptıkları çalışmada, submillisievert göğüs ve abdominal bilgisayarlı tomografi'nin (BT) derin öğrenme rekonstrüksiyonunda (DÖR) ve yinelemeli rekonstrüksiyonunda (IR) görüntülerinden görüntü kalitesini kullanarak lezyon tespiti-nin bulunması incelenmiştir. Çalışmada 59 yetişkin insandan görüntüler alınmıştır. Oluşturulan veri setleri yorumlanarak görüntü kalitesi ve lezyon tespiti uzmanlar tarafından incelenmiştir. Yapılan çalışma derin öğrenme ile sağlıklı bireylerin belirlenmesine öncülük etmiştir.

Villacrés ve Auat Cheein (2020) yaptıkları çalışmada, kiraz algılama, sınıflandırma ve sayma için derin öğrenme stratejilerinin, özellikle de Faster R-CNN'nin kullanılabilişliği gösterilmiştir. Görüntü veri tabanı, üç aylık bir süre içinde elde edilerek 5 satırı kaplayarak günün farklı saatlerinde (farklı aydınlatma koşulları) 15.000'e kadar RGB görüntüsü elde edilmiştir. Kamera bir traktöre monte edilerek 25 deneme sürüşü yapılarak verilerin toplanması sağlanmıştır. Kirazları saymada başarı oranı % 85 olarak bulunmuştur.

Kumari ve ark. (2020) çalışmada, mango meyvesinin görüntülerinden kalitesinin belirlenebileceği bir uygulama geliştirilmiştir. MÖ kullanılarak gelişmiş bulanık tabanlı K-ortalamlar kümeleme algoritması segmentasyonun verimliliğini artırmak için tasarlanmıştır. Önerilen segmentasyon yöntemi, K-ortalamlar ve Bulanık C-ortalamlar kümeleme yöntemleri ile karşılaştırılmıştır. Görüntüleri A Sınıfı iyi kalite mango, B Sınıfı ortalama kalite mango ve C Sınıfı düşük kaliteli mango olarak üç sınıfa ayrılmıştır. Sonuç olarak modelin, çeşitli kusurlu ve sağlıklı mango görüntülerini incelemekte ve önerilen yöntemin mevcut yöntemlerle karşılaştırıldığında en yüksek doğruluğa sahip olduğu kanıtlanmıştır.

Lin ve Guo (2020) yaptıkları çalışmada, görüntü segmentasyonu ve CNN modelinin entegrasyonuna dayalı olarak İnsansız hava araçları (UAV) görüntüleri kullanılarak sorgum salkımı algılama ve sayma boru hattı geliştirilmiştir. Çalışmada RGB kameralı bir UAV kullanılmıştır. Görüntüler kayıt altına alınmış ve eğitim veri seti oluşturulmuştur. Eğitim veri kümeleri kullanılarak bir U-Net CNN modeli oluşturulmuştur.

Sorgum salkımları, algoritma aracılığıyla tahmin edilen bir maske ile tespit edilmiş ve sayılmıştır. Algoritma, derin öğrenme prosedürü için Tensorflow kütüphanesi ile Python ve sorgum salkımı sayımı işlemi için OpenCV kütüphanesi kullanılarak uygulanmıştır. Eğitim Doğruluğu görüntülerinin sayısı ile birlikte genel bir artış eğilimine sahip olduğunu göstermiştir. Algoritma %95,5 doğruluk ve 2,5 ortalama karekök hatası (RMSE) ile 1000 eğitim görüntüsü için en iyi performansı göstermiştir. Görüntü segmentasyonu ve U-Net CNN modelinin entegrasyonunun sorgum salkımı sayımı için doğru ve sağlam bir yöntem olduğunu ve sorgum yetiştirme verimliliği ile doğru verim tahmini için bir fırsat sunduğu gösterilmiştir.

Kanjana Devi ve Rathamani (2020) yaptıkları çalışmada, meyve hastalıklarının tarımla uğraşan insanlar için verim kaybı ve maliyetten kayıp olduğu açıklanmıştır. Manuel sistemlerin, meyve hastalıklarının belirlenmesinde yeteri kadar başarı gösterilmediği ifade edilmiştir. Yenilikçi ve önceden tahmin yoluyla meyve hastalıklarının önlenileceği ve meyve hastalıklarının belirlenmesinde görüntü segmentasyonu uygulamaları ile üstesinden gelerek çözümlere fayda sağlayabileceği öngörülmüştür.

Yanes ve ark. (2020) yaptıkları çalışmada, bitkilerin büyüme sürecini izleyerek bitkiler hakkında tahminler yürütülebileceği ifade edilmiştir. Bunun için Aquaponik tekniği ile bilgisayarlı görme teknolojisi birleştirilerek bitki büyüme sürecine katkı sağlanabileceği bir metot önerilmiştir. Büyüyen bitkilerin boyutunu tahmin etme ve görüntü segmentasyonu kullanılarak mahsullerin büyüklüğü ve taze ağırlıkları hakkında fikir söylenilmiştir. Çalışmada küçük taşlı marul için ekinlerin boyutu 30 mm (%18,7) ve taze ağırlıklığı 0.5 g (%8.3) toplam hatayla tahmin edilmiştir.

Çıtlak ve Demirpolat (2021) yaptıkları çalışmada, duman borulu kazanların ısı transferini DÖ yöntemi ile artış değerleri incelenmiştir. 125.000 kcal/h ısı kapasiteli 42 milimetre çapında, 230 milimetre baca çapında ve 65 milimetre su giriş ve çıkış çaplarına sahip katı yakıtlı bir kazanda 4 farklı tip yakıt kullanılmıştır. Kazanların ısı transferinin iyileştirilmesi hedeflenmiştir. Yapılan deney sonuçları damper yardımı ile fanın debisini değiştirmiş ve Reynolds sayısı 18000 ile 28000 arasında hesaplanmıştır. Hesaplamalar sonucunda elde edilen ısı transferi artış değerleri için MÖ algoritmalarından destek vektör makinesi (SVM) ve karar ağacı (M5P model ağacı) kullanılarak tahminler elde edilmiştir.

Wang ve ark. (2021) çalışmada çiftlik hayvanlarının vücut ağırlıklarının görüntülerden tahmin edilmesi incelenmiştir. Hayvanların vücut ölçüleri alınarak özellikleri

kayıt altına alarak sonraki görüntülerden tahmin edilebileceği belirtilmiştir. Gelenek-sel tartma ve temassız elektro-optik sensörlerin bilgisayarla görme (CV) ile karşılaştırılma-sını yaparak zamandan tasarruf ve maliyetten kazanç elde edilebileceği belirtmiştir. MÖ ve DÖ, 2B ve 3B görüntüler gibi YZ alanlarının eş zamanlı ilerlemesiyle BW tahminleri için biyometrik ve morfometrik özellikler ile CV tabanlı ve MÖ/DÖ tabanlı BW tah-min yöntemlerinin bir incelemesini yapılmıştır. Güçlü yanları, zayıf yönleri ve endüstri-ye uygulanabilirlik potansiyellerinin tartışılması istenmiştir.

Atila ve ark. (2021) yaptıkları çalışmada, görüntülerin analizi için güvenilir so-nuçlar elde etmek için manuel olarak öznelite çıkarımının kusursuz olması istenilen klasik makine öğrenmesi yerine ön işlemeye gerek duymadan ve başarılı bir sınıflan-dırma yapabilen bir yöntem kullanılmıştır. Bitki yaprak hastalıkları sınıflandırmasında EfficientNet derin öğrenme mimarisi önerilmiş ve bu modelin performansı diğer derin öğren-me modelleriyle karşılaştırılmıştır. Modelleri eğitmek için PlantVillage veri seti kullanılmıştır. Test veri setinde elde edilen sonuçlar, EfficientNet mimarisinin B5 ve B4 modellerinin orijinal ve artırılmış veri setlerinde doğruluk için sırasıyla %99.91 ve %99.97 ve hassasiyet için sırasıyla %98.42 ve %99.39 ile diğer derin öğrenme modelle-rine göre en yüksek değerleri elde ettiğini göstermiştir.

Kienbaum ve ark. (2021) yaptıkları çalışmada, mısır koçanlarının mahsul veri-mini artırmak için fenotiplerinin önemli olduğu belirtilmiştir. Veri seti olarak Peru'dan yerel mısır çeşitliliği görüntü olarak kullanılmıştır. Mısır koçanı görüntülerini CNN gibi derin öğrenme yöntemleri kullanılarak segmentasyonu yapılmıştır. Klasik görüntülerle fenotipleme karşılaştırılmıştır. Verimli fenotipleme için derin öğrenmenin yararlı oldu-ğu gösterilmiştir.

André Sant'Ana ve ark. (2021) yaptıkları çalışmada, bilgisayarla görme teknolo-jisi ve makine öğrenimini birleştirerek çiftliklerdeki hayvanların tartım sürecinin kolay-laştırılması incelenmiştir. Koyunların ağırlık kontrolü yapılarak görüntüleri alınmıştır. Yapılan çalışma bir veri seti oluşturularak sekiz makine öğrenme tekniği ile regresyonu yapılmıştır. Başarılı bir şekilde görüntüler kullanılarak koyun ağırlığı tahmin edilmiştir.

Chang ve ark. (2021) yaptıkları çalışmada yüksek verimli fenotipleme platformu (HTPP) ile bir bitkinin yaşam döngüsü verileri incelenerek verim tahmini yapılmıştır. HTTP verileri ile mahsullerin verim performansı arasında bir ilişki kurmak ve MÖ kul-lanarak temel zaman aralıklarını seçmek için birden çok DÖ ağı kullanılmıştır. Görüntü-ler belli aralıklarla alınarak büyüme süreci izlenmiştir. Yapraklı bitkilerin veriminin ve

dikey tarımda tüketilen toplam biyokütlenin anlaşılması sağlanmıştır. Zaman aralıklarının analizi ile gelecekte daha fazla kullanılacağı belirtilmiştir.

Lu ve ark. (2022) çalışmada, soya fasulyesi görüntülerini kullanarak verimini ve ağırlığını tahmin etmek için bir derin öğrenme algoritması oluşturulmuştur. Çalışma ekim ortamında gerçekleştirerek soya fasulyesi büyüme sezonu boyunca gözlemlenmiştir. Tarla içini tahmin edecek bir yöntem önererek Genelleştirilmiş Regresyon Sinir Ağı (GRNN) ile birleştirilen bir derin öğrenme metodu kullanılması belirtilmiştir. Fasulye kapsüllerini tanımak için tanıma hassasiyeti yüksek dedektörler kullanılmıştır. Bölgesel tabanlı evrişimli sinir ağı (R-CNN) mimarisi kullanılarak hem tanıma hassasiyeti hem de hızı dikkate alınarak YOLOv3 algoritması seçilmiştir. Ekilen yerlerdeki soya fasulyesi bakla türlerinin ağırlıkları, tohum sayıları ve yaprak şekilleri belirlenerek son olarak veri seti oluşturularak görüntüler eğitilmiştir. Ortalama doğruluk oranı %97,43'e kadar çıkarılarak görüntüler tahmin edilmiştir. Böylelikle sinir ağı, bitki soya fasulyesindeki verimi yerinde mükemmel bir doğrulukla tahmin edebilmiştir

3. MATERYAL ve YÖNTEM

3.1. Verilerin Elde Edilmesi

Tez çalışmasında 5000 adet Diyarbakır karpuzunun fotoğrafının çekilmesi ve ağırlıklarının ölçülmesi için veri seti oluşturulmuştur. Bu veri setinin oluşturulması için marketler, gıda hali ve pazar yerleri belirlenmiştir. Şekil 3.1’de Diyarbakır karpuzunun fotoğraf çekimi için gidilen yerler gösterilmektedir.



Şekil 3.1. Diyarbakır Karpuzu'nun satıldığı yerlerden kareler

Karpuz çekimi için gidilen yerlerde fotoğraflar 16 MP kamerası olan bir telefon ve selfie çubuğu ile 65 cm mesafeden çekilmiştir. Şekil 3.2’de karpuz fotoğraflarının çekim şekli gösterilmektedir.



Şekil 3.2. 576 numaralı karpuzun fotoğrafının çekim esnası

Beyaz ve siyah örtü ile arka plan oluşturulmuştur. Arka plan oluşturularak görüntüde yalnızca karpuz görüntüsünün kolayca algılanması amaçlanmaktadır. Şekil 3.3. de fotoğrafı çekilen karpuzların arka plan gösterimleri gösterilmektedir.



Şekil 3.3. Beyaz örtüde çekilen 243 ve siyah örtüde çekilen 504 numaralı karpuz

Fotoğrafi çekilen karpuzlar daha sonra hassas tartım terazileri ile ağırlıkları kilogram cinsinden tartılarak not edilmiştir. Tartımı yapılan her bir karpuzla numara verilmiştir. Şekil 3.4. de tartılan bir karpuz gösterilmektedir.



Şekil 3.4. 595 numaralı karpuz ve 600 numaralı karpuzların hassas tartım terazisi ile tartılması

3.2. Verilerin İşlenmesi

Oluşturulan veri setinin eğitimi için Python program dili seçilmiştir. Python için Anaconda dağıtım sistemi tercih edilmiştir. Ayrıca Matlab programı da görüntülerin piksel sayılarını karpuz alanının bulunmasında kullanılmıştır. Tez çalışmasında kullanılan programlar aşağıda ifade edilmiştir.

3.2.1. Anaconda

Anaconda, paket yönetimini ve dağıtımını basitleştirmeyi amaçlayan bilimsel hesaplama (veri bilimi, makine öğrenimi uygulamaları, büyük ölçekli veri işleme, tahmine dayalı analitik, vb.) için Python ve R programlama dillerinin bir dağıtımıdır. Dağıtım, Windows, Linux ve macOS için uygun veri bilimi paketlerini içermektedir (Anonymous, 2022).

3.2.2. Spyder

Spyder, Python ile yazılmış ve bilim insanları, mühendisler ve veri analistleri tarafından ve onlar için tasarlanmış Python için ücretsiz ve açık kaynaklı bir bilimsel ortamdır. Veri madenciliği, etkileşimli yürütme, bilimsel bir dizinin derin ve güzel görselleştirmesini test etme ile tam özellikli bir geliştirme aracının gelişmiş düzenleme, analiz, hata ayıklama ve profil oluşturma özelliklerinin benzersiz bir kombinasyonunu sunmaktadır (Anonymous, 2022).

3.2.3. Python

Python, çok çeşitli uygulamalar için kullanılabilen genel amaçlı bir programlama dilidir. Okunabilirliği ve anlaşılmasını kolaylaştırmak için tasarlanmış diğer yapısal unsurları nedeniyle yeni başlayanlar için harika bir dil olan Python, temel kullanımla sınırlı değildir. Aslında, dünyanın en karmaşık uygulamalarından ve web sitelerinden bazılarına güç sağlamaktadır (Anonymous, 2022).

3.3. Python Kütüphaneleri

Derin öğrenme ile verilerin analizini yapmak için Python kütüphanesine ihtiyaç vardır. Bu tez çalışmasında verilerin analizini yapmak için aşağıdaki paket programlar gereklidir.

3.3.1. Numpy

NumPy, Python'da bilimsel hesaplama için temel bir pakettir. Çok boyutlu bir dizi nesnesi, çeşitli türetilmiş nesneler (maskelenmiş diziler ve matrisler gibi) ve diziler üzerinde matematiksel, mantıksal, şekil işleme, sıralama, seçme, ayrık Fourier dönüşümleri dahil olmak üzere hızlı işlemler için çeşitli rutinler sağlayan bir Python kitaplığıdır (Anonymous, 2022). Tez çalışmasında Numpy verilerin sisteme yüklerken kullanılmıştır.

3.3.2. Pandas

Pandas, veri analizi ve veri işleme için kullanılan Python programlama dilindeki güçlü açık kaynak kitaplıklarından biridir. Bir veri tabanından veya diğer formlardan (CSV, JSON, Excel, vb.) gelen veriler gibi herhangi bir tablo verisiyle çalışmak için pandas kütüphanesi en iyi araçtır (Anonymous, 2022). Tez çalışmasında verilerin yüklenmesinde kullanılmıştır.

3.3.3. Matplotlib

Matplotlib, Python'da statik, dinamik ve etkileşimli görselleştirmeler oluşturmak için kapsamlı bir kitaplıktır. Matplotlib kolay şeyleri kolay, zor şeyleri mümkün kılmaktadır. Verileri grafik şekline getirip karşılaştırmaları sağlamaktadır (Anonymous, 2022). Tez çalışmasında verilerin sonuçlarını grafik ortamına aktarmak için kullanılmıştır.

3.3.4. Seaborn

Seaborn, matplotlib tabanlı bir Python veri görselleştirme kütüphanesidir. İlgi çekici ve bilgilendirici istatistiksel grafikler çizmek için bir arayüz sağlamaktadır (Anonymous, 2022). Tez çalışmasında verilerin birbiri ile karşılaştırılıp görsel grafik elde edilmesinde kullanılmıştır.

3.3.5. OpenCV

OpenCV (Open Source Computer Vision Library), açık kaynaklı bir bilgisayarlı görü ve makine öğrenimi yazılım kitaplığıdır. Nesnelerin görüntülerini algılayıp düzenlemektedir. Bilgisayarlı görü uygulamaları için ortak bir altyapı sağlamak ve ticari ürünlerde makine algısının kullanımını hızlandırmak amacıyla oluşturulmuştur (Anonymous, 2022). Tez çalışmasında U-Net mimarisinin oluştururken karpuz görüntülerini sisteme yüklemek için kullanılmıştır.

3.3.6. Os

Os modülü, işletim sistemine bağlı özellikleri kullanmak için taşınabilir bir yol sağlamaktadır (Anonymous, 2022). Tez çalışmasında fonksiyonların kullanılması ve Windows işletim sistemini tanınması için kullanılmıştır.

3.3.7. Scikit Learn

Scikit Learn, makine öğrenmesinde kullanılan ve verileri modellemeye odaklanan bir kütüphanedir. Yalnızca modellemeye odaklanırken verileri yüklemeye ve manipüle etmeye odaklanmamaktadır (Kumar, 2021). Tez çalışmasında verilerin sınıflandırılması ve algoritmalara ayrılmasını sağlarken kullanılmıştır.

3.3.8. Scikit-image (Skimage)

Scikit-image, görüntü işleme için bir algoritmalar koleksiyonudur (Anonymous, 2022). Tez çalışmasında U-Net görüntülerini işletmesine yardımcı olarak kullanılmıştır.

3.3.9. TensorBoard

TensorBoard, makine öğrenimi iş akışı sırasında ihtiyaç duyulan ölçümleri ve görselleştirmeleri sağlamaya yönelik bir araçtır. Kayıp ve doğruluk gibi deney ölçümlerinin izlenmesine, model grafiğinin görselleştirilmesine, yerleştirmelerin daha düşük boyutlu bir alana yansıtılmasına ve çok daha fazlasına olanak tanımaktadır (Anonymous, 2022). Tez çalışmasında U-Net görüntü sonuçlarının elde edilmesinde kullanılmıştır.

3.3.10. Tqdm

Tqdm, programın yürütülmesinin arkasındaki işlemeyi gösteren ilerleme çubukları eklemek için kullanılmaktadır (Kharwal, 2021). Tez çalışmasında programın işleyişini sağlamaktadır.

3.3.11. Random

Random, çeşitli dağıtımlar için sözde rastgele sayı üreteçlerini uygulamaktadır (Anonymous, 2022). Tez çalışmasında görüntüleri rastgele sayı ile başlatmak için kullanılmıştır. Random seed eğitim modelinde 42 olarak seçilmiştir.

3.4. Matlab

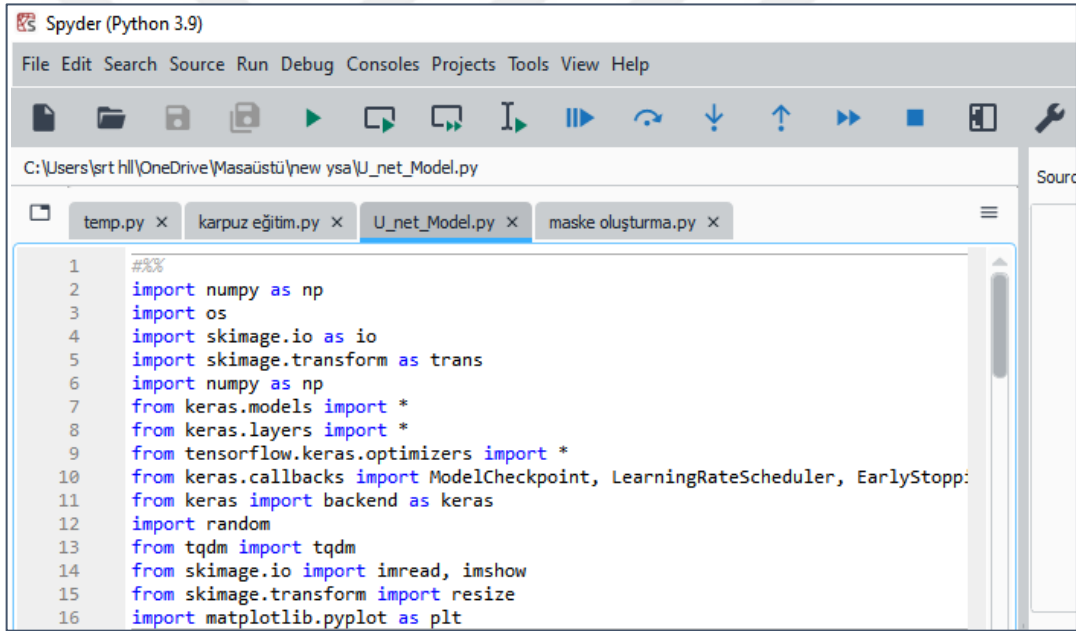
Matlab, yinelemeli tasarım ve analiz süreçlerine uygun bir masaüstü ortamını, matrisleri ve matris matematiğini doğrudan temsil eden bir programlama dili ile birleştiren bir yazılım programıdır. Yürütülebilir bir not defterinde kod, çıktı ve biçimlendirilmiş metni birleştiren komut dosyaları oluşturmak için canlı düzenleyici içermektedir (Anonymous, 2022). Tez çalışmasında karpuz görüntülerinin tüm fotoğrafta elde edilen piksel alanını bulmak için kullanılmıştır.

3.5. U-Net mimarisi ile görüntü alanını belirleme

Tez çalışmasındaki görüntüler tam evrişimli ağda U-Net ile semantik segmentasyonu yapılarak eğitilmiştir. Görüntülerde piksellerin eğitilmesiyle nesne belirlenmiştir. Belirlenen nesne daha sonra piksel alan metoduyla kilogramı tahmin edilmektedir.

3.5.1. Gerekli kütüphanelerin yüklenmesi

Kütüphaneleri içe aktararak (import) verilerin çalışması için Spyder programı çalıştırılmıştır. Gerekli kodlar Spyder’da yazılmıştır. Şekil 3.5’te kütüphanelerin yükleme kodları gösterilmektedir.



```

1  ###
2  import numpy as np
3  import os
4  import skimage.io as io
5  import skimage.transform as trans
6  import numpy as np
7  from keras.models import *
8  from keras.layers import *
9  from tensorflow.keras.optimizers import *
10 from keras.callbacks import ModelCheckpoint, LearningRateScheduler, EarlyStopping
11 from keras import backend as keras
12 import random
13 from tqdm import tqdm
14 from skimage.io import imread, imshow
15 from skimage.transform import resize
16 import matplotlib.pyplot as plt
  
```

Şekil 3.5. Python kütüphanelerinin yükleme kodları

3.5.2. Maske oluşturma

Karpuz görüntülerinin arka planları alınarak görüntüde sadece karpuz şeklinin kalması hedeflenmiştir. İnternet üzerinde (zyro, 2022) adlı sitede görüntülerin arka planları silinmiştir. Şekil 3.6’da arka planı silinmiş bir karpuz görüntüsü yer almaktadır.



Şekil 3.6. 50 numaralı karpuzun arka planının silinmesi

Karpuzların arka planlarının silinmesi ardından ortaya çıkan karpuz şekillerinin Spyder programında maskeleri alınmıştır. Bu aşamada karpuz şeklinin alanının belirlenmesi hedeflenmektedir. Programda kullanılan maske oluşturma kodu Şekil 3.7’de gösterilmektedir.

```

Spyder (Python 3.9)
File Edit Search Source Run Debug Consoles Projects Tools View Help

C:\Users\art hll\OneDrive\Masaüstü\new ysa\maske oluşturma.py

temp.py x karpuz eğitim.py x U_net_Model.py x maske oluşturma.py* x

1 import cv2
2 import matplotlib.pyplot as plt
3
4 img = cv2.imread("zyro-image (50).png")
5
6 print(img.size)
7 def rescale(frame, scale=1):
8     width = int(frame.shape[1] * scale)
9     height = int(frame.shape[0] * scale)
10
11     boyut = (width, height)
12     return cv2.resize(frame, boyut, interpolation=cv2.INTER_AREA)
13
14 resized_image = rescale(img)
15
16 resim=cv2.cvtColor(resized_image, cv2.COLOR_BGR2GRAY)
17
18
19 (thresh, blackAndWhiteImage) = cv2.threshold(resim, 0, 255, cv2.THRESH_BINARY)
20
21 plt.imshow(blackAndWhiteImage)
22 plt.show()
23 cv2.resize(blackAndWhiteImage, (3000, 2250))
24 cv2.imwrite("resim50.png", blackAndWhiteImage)
25

```

Şekil 3.7. Maske oluşturma kodu

Kodlar çalıştırıldığında karpuz görüntülerinin maske görüntüleri oluşmuştur. Şekil 3.8’ de arka planı silinen bir karpuzun maske görüntüsü yer almaktadır.



Şekil 3.8. 50 numaralı karpuzun silinen arka planının maske görüntüsü

Maskeler elde edildikten sonra U-Net mimarisinde kullanılmak üzere eğitim (train) dosyasına kaydedilmiştir. U-Net mimarisinde karpuz görüntüsünün algılanmasında kullanılmıştır.

3.5.3. Verilerin hazırlanması

Bu bölümde rastgele (random) sayı üretici ile verilen değerle veriler başlatılmıştır. Giriş resminin boyutları 128x128x3 olarak tanımlanmıştır. Maskeler de 128x128 olarak belirlenmiştir. Test için 400 karpuz görüntüsü, eğitim için ise 1000 karpuz görüntüsü ve 1000 maske görüntü belirlenmiştir. Eğitim ve test görüntüleri yeniden boyutlandırılarak okunmuştur ve yeni liste oluşturulmuştur. Maske yapılarak karpuz şeklini algılanması ve mimaride tanımlanması amaçlanmaktadır. Görüntülerin listesi oluşturularak test ve eğitim için hazır hale getirilmiştir.

3.5.4. U-Net modelinin oluşturulması

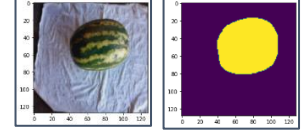
U-Net mimarisi ile biyomedikal alanında yapılan başarılı sonuçlar nedeniyle bu tez çalışmasında tercih edilmiştir. Az görüntü ile çalışabilmesi ve performansının iyi olması nedeniyle yapılan eğitimler yüksek başarı oranı yakalmaktadır. U-Net modeli oluşturulurken iki kısımda model oluşturulmuştur. U'nun sol kısmında kodlayıcı kısımdır. Sağ kısmı ise kod çözücü kısımdır. Şekil 3.9'da U-Net modelinin akış şeması gösterilmektedir.

Giriş Görüntüsü ve Maskesi

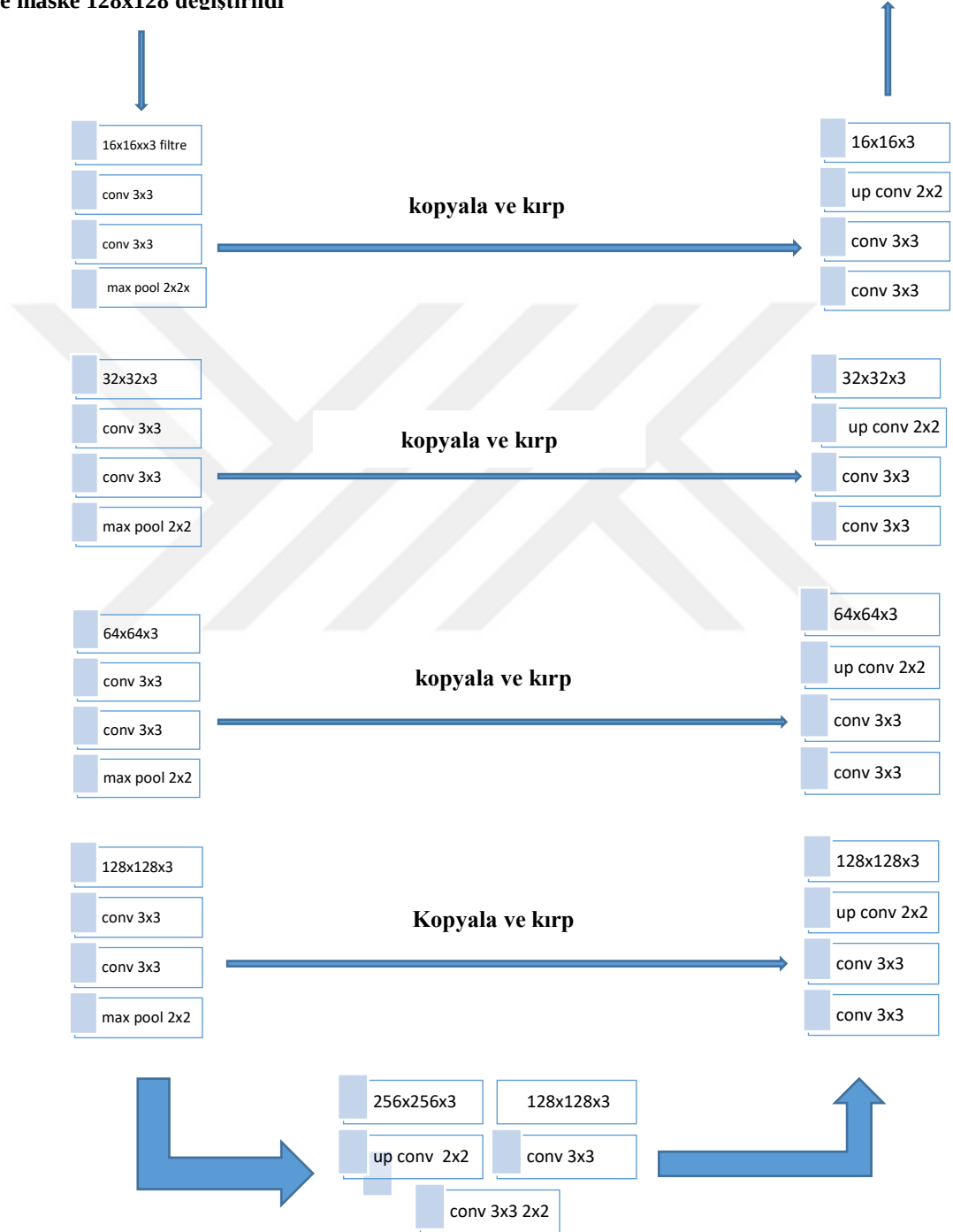


Giriş görüntüleri 128x128x3
ve maske 128x128 değiştirildi

Çıkış görüntüsü ve Segmentasyon



Segmentasyon 128x128x3



Şekil 3.9. U-Net modelinin akış şeması

Boyutlandırılan girdi görüntüleri ve maskeleri model için hazır hale getirilmiştir. Model 23 Evrişim katmanı içermektedir. 19 evrişim katmanı ve 4 transpoze evrişim katmanı vardır. Giriş görüntüleri 128x128x3 olarak verilmiştir. Görüntülerdeki pikseller 0-1 arasında normalleştirilerek 255 parçaya bölünmüştür. U-Net modeline geçmeden önce aşağıda kavramların bilinmesi gereklidir:

Conv2D: Evrişim katmanları görüntüleri filtreleyerek görüntünün boyutunu küçültür ve özelliklerini belirler.

Max-pooling2D: Mevcut verilerin en iyi şekilde işlenmesini sağlar. Görüntülerin parametre sayısını azaltarak boyutları küçültür.

Dropout: Bırakma işlevi, aşırı öğrenmeden (overfitting) kaçınmak için bazı nöronları unutmak için uygulama sırasında kullanılır.

Padding: Görüntülerin kenarlarını doldurmaya yarar.

Conv2DTranspose: Giriş maskesine benzer bir maske üretmek için evrişim katmanına ters yönde bir dönüşüm sağlar.

Concatenate: Bir girdi listesini birleştiren katmandır. Girdi olarak birleştirme ekseninde dışında tümü aynı şekle sahip bir tensör listesi alır ve tüm girişlerin birleştirilmesi olan tek bir tensör döndürür.

Şekil 3.10'da U-Net kodlayıcı kısımda yapılan işlemlerin kodları gösterilmektedir. Şekil 3.10 incelendiğinde Filtre sayısı 16 ve boyutu (3,3) seçilerek evrişim katmanını belirlenmiştir. Evrişim katmanında matris filtre sayısı azaltılarak iyileştirme yapılmıştır. Aktivasyon fonksiyonu ReLU olarak seçilmiştir. Ağırlık başlatma yöntemi he_normal ve padding (doldurma) same seçilerek Dropout 0.1 belirlenerek geri geçiş engellenmiştir. Maksimum havuzlama katmanı boyutu 2x2 belirlenmiştir. Maksimum havuzlama katmanında görüntülerin parametre sayısı azaltılarak boyut azaltılmıştır.

Sonra ki adımlarda evrişim katmanı filtre sayısı ikiye katlanarak 64, 128, 256 olmuştur. Dropout 0,1, 0.2 ve 0.3 arasında değişmiştir. Maksimum havuzlama katmanları 2x2 olarak devam etmiştir.

Burada c1, c2, c3, c4 ve c5 ifadeleri evrişim katmanlarını temsil ederken p1, p2, p3 ve p4 ifadeleri ise maksimum havuzlama katmanının temsil etmektedir.

```

86
87 #Build Model
88 inputs=Input((IMG_WIDTH, IMG_HEIGHT, IMG_CHANNELS))
89
90 s=Lambda(lambda x: x/255)(inputs)
91
92 c1=Conv2D(16, (3,3), activation="relu", kernel_initializer="he_normal", padding="same")(s)
93 c1=Dropout(0.1)(c1)
94 c1=Conv2D(16, (3,3), activation="relu", kernel_initializer="he_normal", padding="same")(c1)
95 p1=MaxPooling2D((2,2))(c1)
96
97 c2=Conv2D(32, (3,3), activation="relu", kernel_initializer="he_normal", padding="same")(p1)
98 c2=Dropout(0.1)(c2)
99 c2=Conv2D(32, (3,3), activation="relu", kernel_initializer="he_normal", padding="same")(c2)
100 p2=MaxPooling2D((2,2))(c2)
101
102
103 c3=Conv2D(64, (3,3), activation="relu", kernel_initializer="he_normal", padding="same")(p2)
104 c3=Dropout(0.2)(c3)
105 c3=Conv2D(64, (3,3), activation="relu", kernel_initializer="he_normal", padding="same")(c3)
106 p3=MaxPooling2D((2,2))(c3)
107
108 c4=Conv2D(128, (3,3), activation="relu", kernel_initializer="he_normal", padding="same")(p3)
109 c4=Dropout(0.2)(c4)
110 c4=Conv2D(128, (3,3), activation="relu", kernel_initializer="he_normal", padding="same")(c4)
111 p4=MaxPooling2D((2,2))(c4)
112
113 c5=Conv2D(256, (3,3), activation="relu", kernel_initializer="he_normal", padding="same")(p4)
114 c5=Dropout(0.3)(c5)
115 c5=Conv2D(256, (3,3), activation="relu", kernel_initializer="he_normal", padding="same")(c5)
116

```

Şekil 3.10. U-Net modelinin kodlayıcı kısmı kodları

Şekil 3.11’de kod çözücü kısmın kodları gösterilmektedir. Şekil 3.11 incelendiğinde modeli oluştururken ederken iki yeni katman da sürece dahil olmuştur. Conv2Dtranspose katmanı filtre sayısını küçülterek yeni bir maske oluşturmaktadır. Filtre sayısı 128, 64, 32 ve şeklinde gitmektedir. Concatenate birleştirme katmanı ardı ardına gelen girişleri birleştirmiştir. Havuzlama katmanının bu bölümde yoktur. Olmayışının sebebi çıktının girdi boyutunu kopyalamasını istenilmesidir ve burada girdi bir görüntü ve onun maskesi olurken beklenen çıktı da bir maske olmak zorundadır.

Burada u6, u7, u8 ve u9 ifadeleri transpoze evrişim katmanını temsil ederken c5, c6, c7, c8 ve c9 ifadeleri ise evrişim katmanlarını temsil etmektedir.

```

118 # expansive path
119
120 u6=Conv2DTranspose(128, (2,2), strides=(2,2), padding="same")(c5)
121 u6=concatenate([u6,c4])
122 c6=Conv2D(128, (3,3), activation="relu", kernel_initializer="he_normal", padding="same")(u6)
123 c6=Dropout(0.2)(c6)
124 c6=Conv2D(128, (3,3), activation="relu", kernel_initializer="he_normal", padding="same")(c6)
125
126
127 u7=Conv2DTranspose(64, (2,2), strides=(2,2), padding="same")(c6)
128 u7=concatenate([u7,c3])
129 c7=Conv2D(64, (3,3), activation="relu", kernel_initializer="he_normal", padding="same")(u7)
130 c7=Dropout(0.2)(c7)
131 c7=Conv2D(64, (3,3), activation="relu", kernel_initializer="he_normal", padding="same")(c7)
132
133
134 u8=Conv2DTranspose(32, (2,2), strides=(2,2), padding="same")(c7)
135 u8=concatenate([u8,c2])
136 c8=Conv2D(32, (3,3), activation="relu", kernel_initializer="he_normal", padding="same")(u8)
137 c8=Dropout(0.2)(c8)
138 c8=Conv2D(32, (3,3), activation="relu", kernel_initializer="he_normal", padding="same")(c8)
139
140
141 u9=Conv2DTranspose(16, (2,2), strides=(2,2), padding="same")(c8)
142 u9=concatenate([u9,c1], axis=3)
143 c9=Conv2D(128, (3,3), activation="relu", kernel_initializer="he_normal", padding="same")(u9)
144 c9=Dropout(0.2)(c9)
145 c9=Conv2D(128, (3,3), activation="relu", kernel_initializer="he_normal", padding="same")(c9)
146
147 outputs=Conv2D(1, (1,1), activation="sigmoid")(c9)
148
149 model=Model(inputs=[inputs], outputs=[outputs])
150 model.compile(optimizer="adam", loss="binary_crossentropy", metrics=["accuracy"])
151 model.summary()

```

Şekil 3.11. U-Net modelinin kod çözücü kısmı kodları

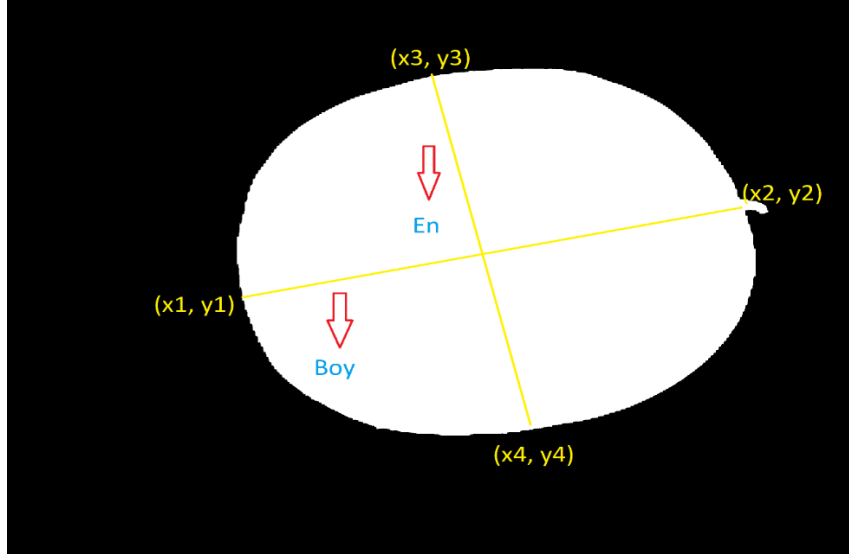
Çıkış katmanında sigmoid aktivasyon fonksiyonu kullanılmıştır. Optimize olarak Adam kullanılmıştır. Adam'da her ağırlık için bir öğrenme oranı belirlendiği için görüntüler tekrar tekrar uyarlanmaktadır.

3.6. Piksel İle Karpuzların En ve Boylarının Belirlenmesi

Karpuz görüntülerinin ağırlıklarının tahmininin yapılması için görüntülerden karpuzun kapladığı piksel boyutu ile en ve boy belirlenebilmektedir. Karpuz görüntüsünün U-Net mimarisinde elde edilen maske görüntüsü ile en ve boyun belirlenmesi Windows Paint programında yapılmıştır. Şekil 3.12'de piksellerle en ve boyun nasıl yapıldığı gösterilmektedir. Elde edilen veriler yapay sinir ağında kullanılmıştır. En ve boy hesabı aşağıdaki bağıntılar ile hesaplanmıştır:

$$\text{Boy} = \frac{\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}}{\text{piksel boyu}} \quad (3.1)$$

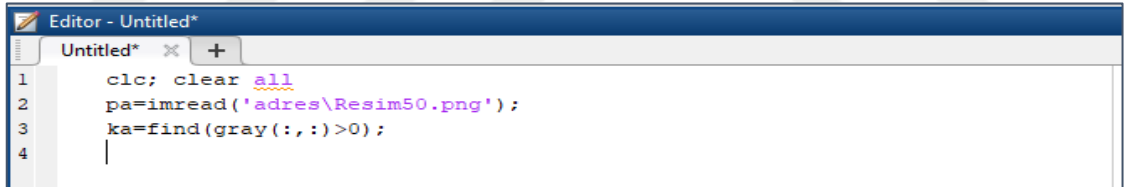
$$\text{En} = \frac{\sqrt{(x_3 - x_4)^2 + (y_3 - y_4)^2}}{\text{piksel eni}} \quad (3.2)$$



Şekil 3.12. Paint'de piksel yardımıyla karpuzların en ve boyunun belirlenmesi

3.7. Piksel alanının bulunması

Maske görüntülerinden sonra Matlab uygulamasında karpuz şeklinin tüm şekilde kapladığı alan bulunmuştur. Şekil 3.13'de kullanılan kod gösterilmektedir.



Şekil 3. 13. Piksel ile alan bulma kodu

Kapladığı alanı oranlarsak karpuz ağırlığı tahmin edebilir. Karpuz şeklinin kapladığı orana A, tüm alana pa (piksel) ve karpuz şeklinin kapladığı alana ka dersek aşağıdaki bağıntı elde edilmiş olur:

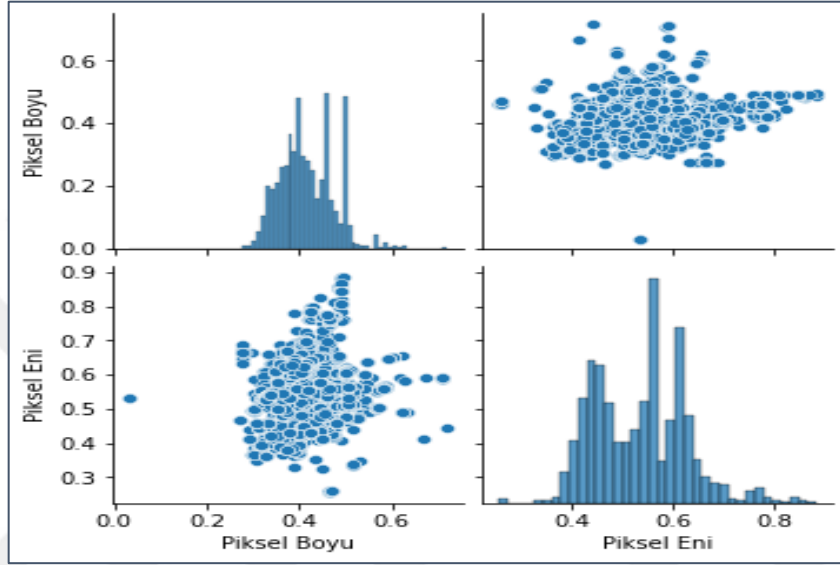
$$A = \frac{ka}{pa} \quad (3.3)$$

50 numaralı karpuz için bu bağıntıyı kullanırsak:

$$A = \frac{2184690}{3000*2250} = 0,323657778 \quad (3.4)$$

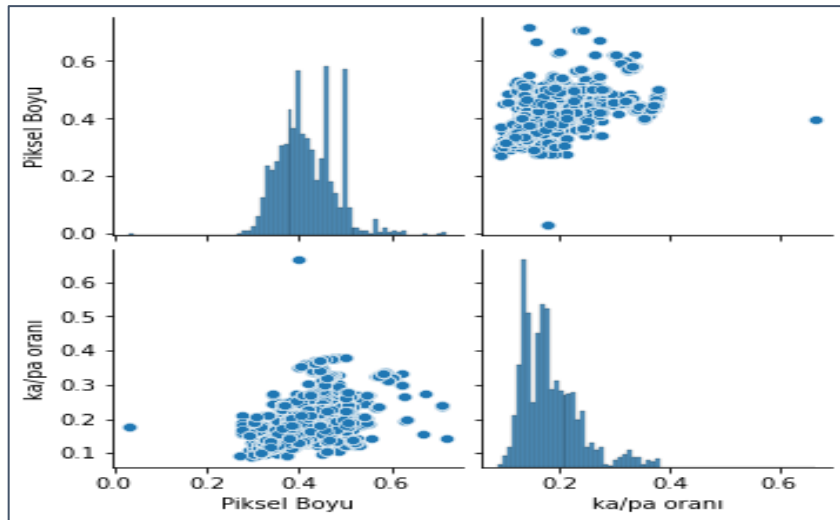
bulunur.

Veriler elde edildikten sonra tablo haline getirilmiştir. Fotoğraflar içinde ayrı bir klasör oluşturulmuştur. Veriler grafikler halinde birbiri ile karşılaştırılmıştır. Şekil 3.14'te karpuz görüntülerinden elde edilen piksel boyu ile piksel eninin birbirleriyle karşılaştırılması gösterilmektedir. Veriler birbirleriyle uyumlu bir biçimde dağılmaktadır.



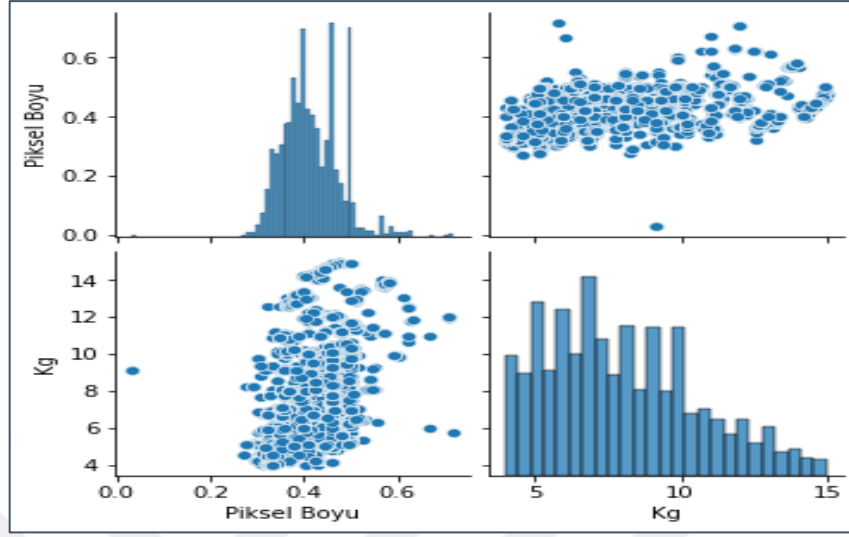
Şekil 3.14. Piksel boyu ile piksel eninin karşılaştırılması.

Şekil 3.15'te piksel boyunun ka/pa oranına göre karşılaştırılması grafikte gösterilmektedir.



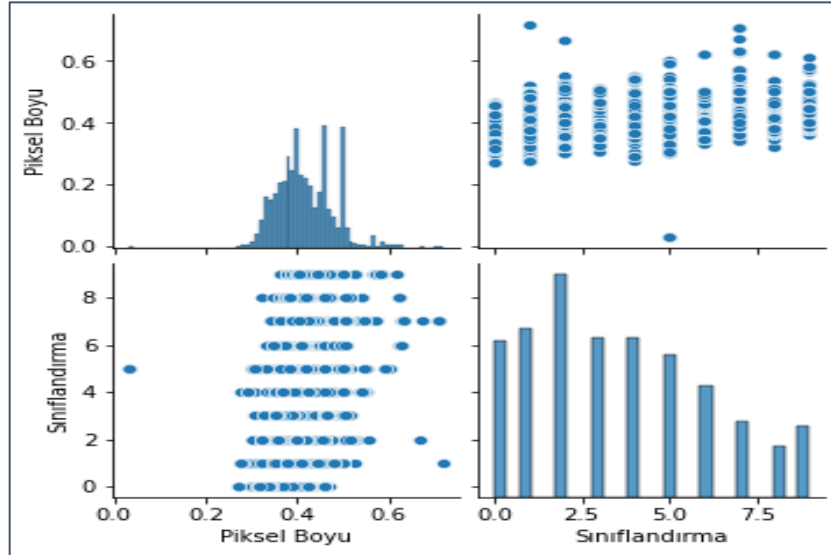
Şekil 3.15. Piksel boyu ile ka/pa oranının karşılaştırılması

Şekil 3.16’da piksel boyu ile karpuzların ağırlıklarını içeren grafik gösterilmektedir. Kg’lar piksel boyuna göre uyumlu bir biçimde kümelenmiştir.



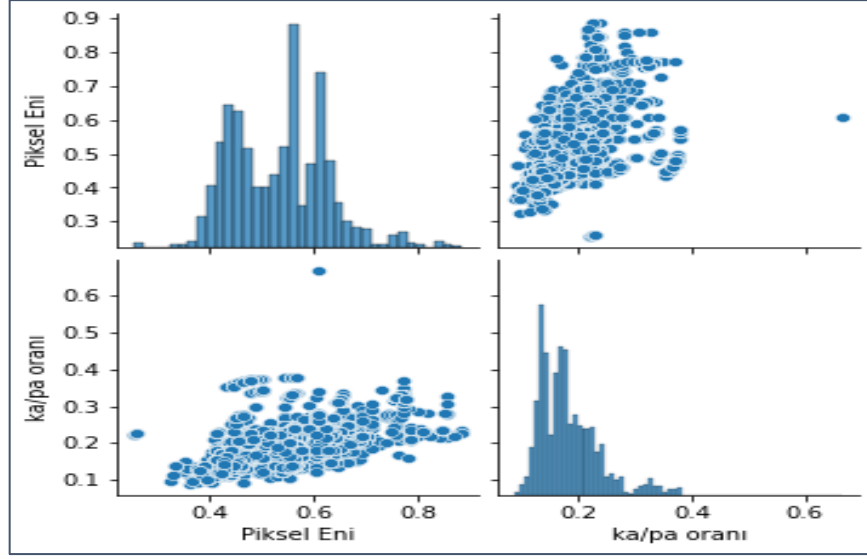
Şekil 3.16. Piksel boyu ile kg'ların karşılaştırılması

Şekil 3.17’de piksel boyunun karpuz ağırlıklarının sınıflandırması ile karşılaştırıldığı grafik gösterilmektedir.



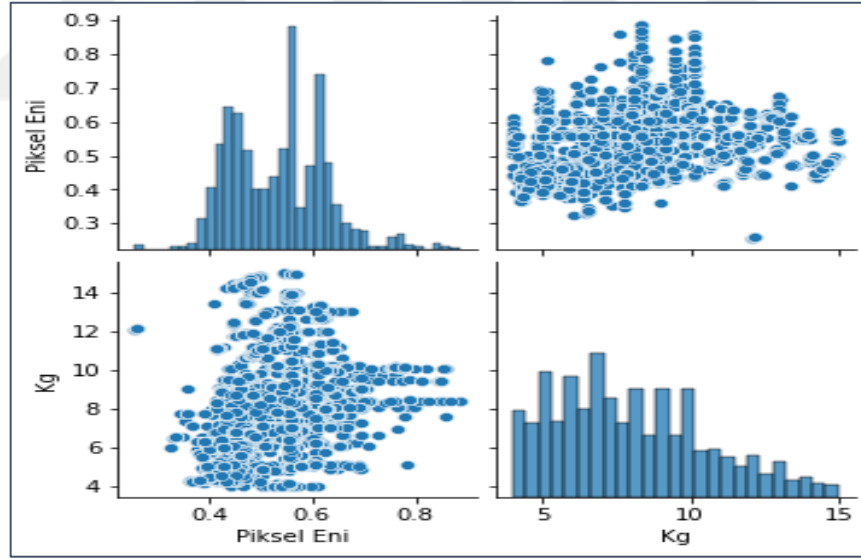
Şekil 3.17. Piksel boyu ile sınıflandırmanın karşılaştırılması

Şekil 3.18’de piksel eni ka/pa oranının karşılaştırıldığı grafik gösterilmektedir.



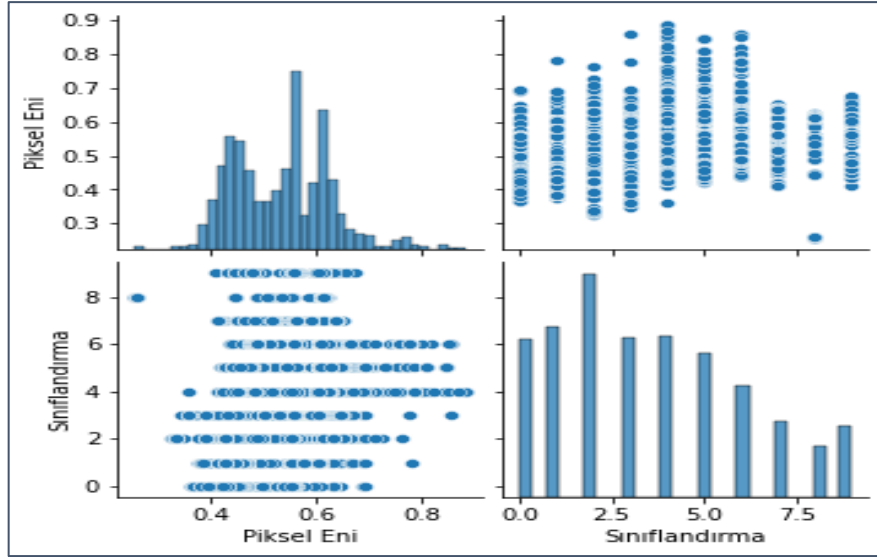
Şekil 3.18. Piksel eni ka/pa oranının karşılaştırılması

Şekil 3.19’da piksel eni ile karpuz ağırlıklarının karşılaştırıldığı grafik gösterilmektedir.



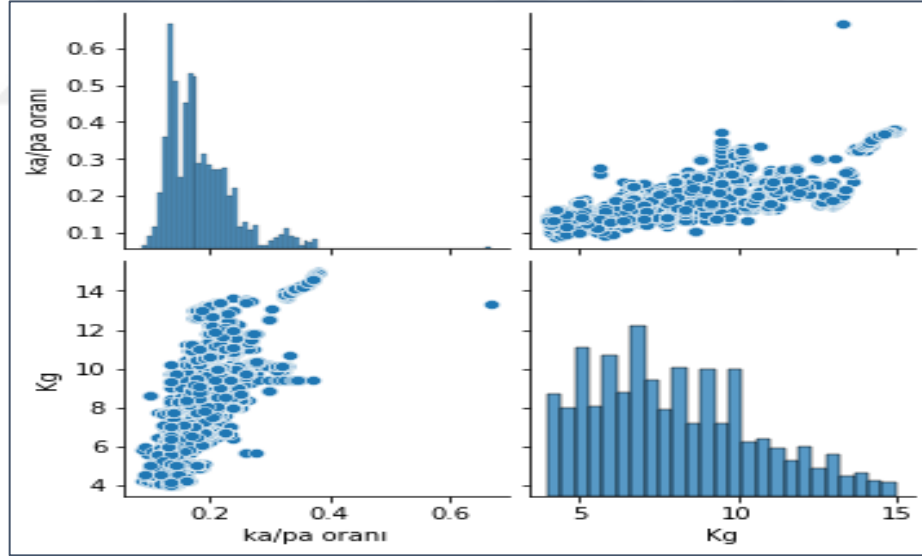
Şekil 3.19. Piksel eni ile kg'ların karşılaştırılması

Şekil 3.20’de piksel eni ile sınıflandırmanın birbiri ile karşılaştırıldığı grafik gösterilmektedir.



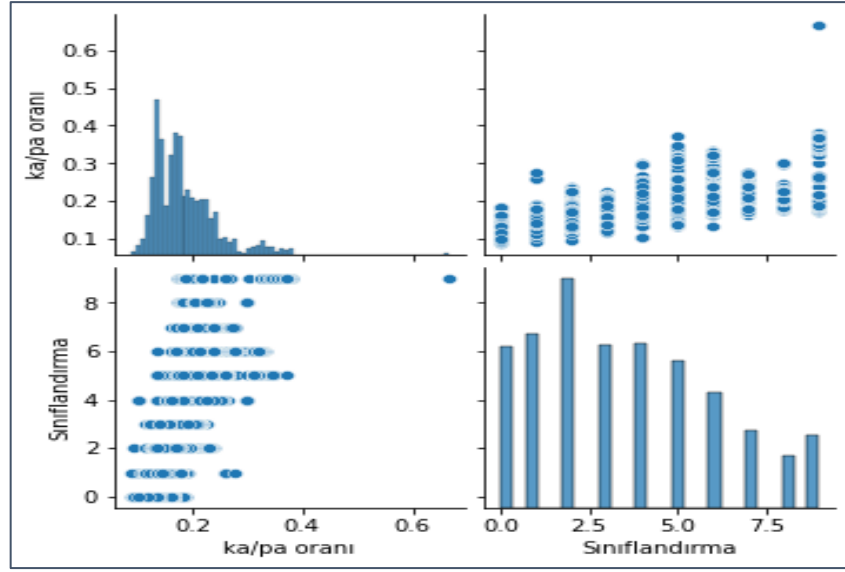
Şekil 3.20. Piksel eni ile sınıflandırmanın karşılaştırılması

Şekil 3.21’de ka/pa oranı ile karpuz ağırlıklarının karşılaştırıldığı grafik gösterilmektedir.



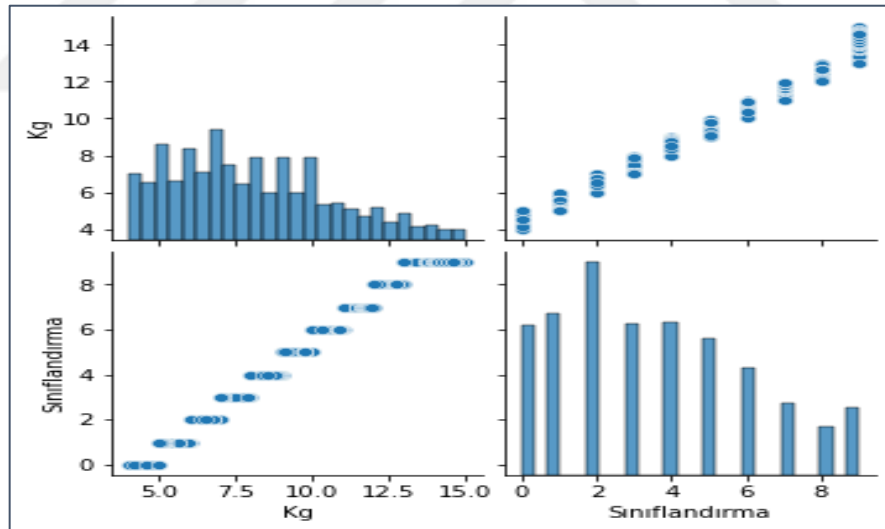
Şekil 3.21. ka/pa oranı ile kg'ların karşılaştırılması

Şekil 3.22’de ka/pa oranının sınıflandırma ile karşılaştırıldığı grafik gösterilmektedir.



Şekil 3.22. ka/pa oranının sınıflandırma ile karşılaştırılması

Şekil 3.23'te kg'ların sınıflandırma ile karşılaştırıldığı grafik gösterilmektedir. Şekil 3.23 incelendiğinde verilerin birbiri ile uyum içinde oldukları görülmektedir.



Şekil 3.23. Kg'lar ile sınıflandırmanın karşılaştırılması

3.8. Verilerin Yapay Sinir Ağları İle Eğitilmesi

Karpuz ağırlıklarına göre sınıflandırma işlemi yapılmıştır. Çizelge 3.1'de ağırlıklara karşılık gelen sınıflandırma işlemi gösterilmektedir. Sınıflandırma 10 ayrı sınıfa ayrılarak yapay sınır ağlarında çıkış katmanı olmuştur.

Çizelge 3. 1. Karpuz ağırlıklarına göre sınıflandırma işlemi

Karpuz Ağırlıkları (kg)	Sınıflandırma (0-9)	Karpuz Sayısı
4 – 4,999	0	594
5 – 5,999	1	641
6 – 6,999	2	844
7 – 7,999	3	602
8 – 8,999	4	606
9 – 9,999	5	542
10 – 10,999	6	423
11 – 11,999	7	288
12 – 12,999	8	191
13 – 14,999	9	269

Sınıflandırma işleminden sonra yapay sinir ağında verileri eğitmek için giriş ve çıkış katmanları belirlenmelidir. Giriş katmanında; boy, en, ka/pa oranı ve bulunmaktadır. Çıkış katmanında ise sınıflandırma bulunmaktadır. Çizelge 3.2’de yapay sinir ağında kullanılan verilerin sırası verilmiştir.

Çizelge 3. 2. Kullanılan verilerin sıralandırılması

Boy (Piksel Boyu)	En (Piksel eni)	ka/pa oranı	Sınıflandırma
0,372403604	0,445519810	0,139808444	0
0,385702246	0,454783085	0,147930963	1
0,380525952	0,489560750	0,156355111	2
0,472652356	0,470154164	0,171111111	3
0,399590102	0,560533784	0,180710667	4
0,370586360	0,588517670	0,207366197	5
0,410002223	0,632022333	0,225555555	6
0,500002233	0,61002234	0,259990526	7
0,423666657	0,44556989	0,288002222	8
0,5711527	0,558889777	0,329933338	9
0,441111223	0,502222256	0,343333333	9
...	...	...	...

Giriş ve çıkış belirlendikten sonra yapay sinir ağlarında kullanılan parametreler belirlenir. En iyi sonuç için yapay sinir ağları 9 farklı şekilde eğitilmiştir. En iyi sonuç 1024 birime ayrılan ve 4 gizli katman kullanılan, aktivasyon fonksiyonu ReLu ve çıkış katmanı softmax seçilen parametre olmuştur. Gizli katmanlar her biri 1024 birime ayrılmış olup dropout 0.25 olarak belirlenmiştir. Çıkış katmanı ise 10 birime ayrılmıştır

Kayıp (loss) fonksiyonu olarak kategorik çapraz entropi, optimizier ise adam, metrik olarak da doğruluk (accuracy) kullanılmıştır. Model de girilen veri boyutu (batch size) 1 olarak belirlenmiştir. Eğitim tur sayısı (epoch) Sayısı 100 ve 3000 iteras-

yon şeklinde model çalıştırılmıştır. Şekil 3.24'te modelin tamamlama ve çalıştırma kodları gösterilmektedir.

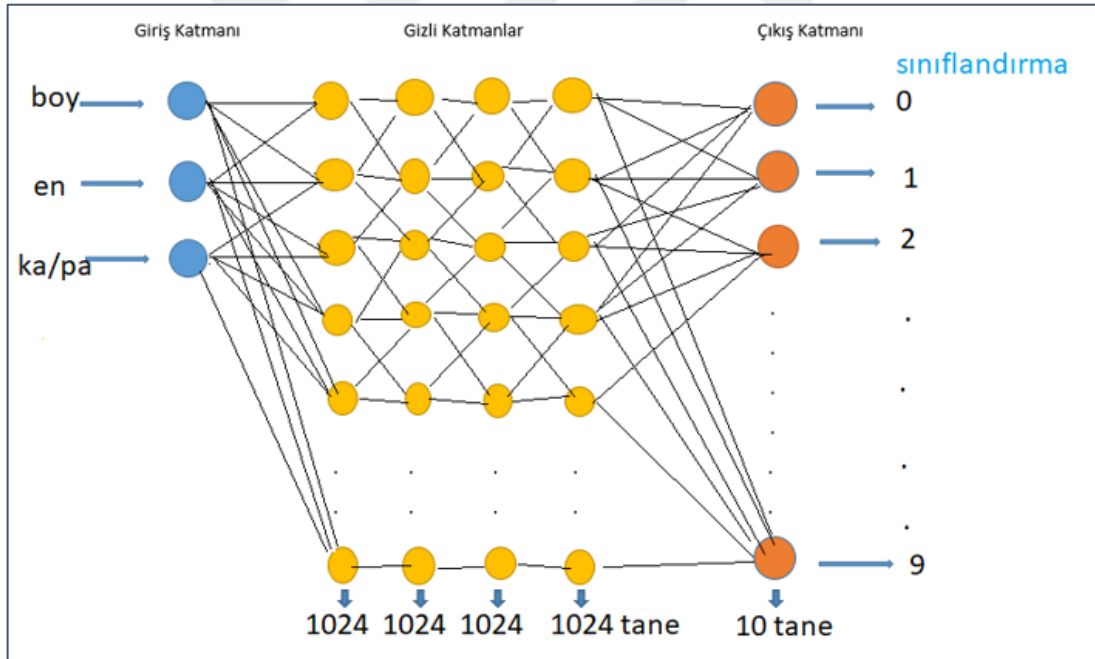
```

61  %% modeli compile etme
62  model.compile(loss="categorical_crossentropy", optimizer=("adam"), metrics=["accuracy"])
63
64
65  %% model çalıştırma
66  hist=model.fit(x_train, y_train, validation_data=(x_test,y_test),batch_size=1, epochs=100)

```

Şekil 3.24. YSA modelinin tamamlama ve çalıştırma kodları

Çalışmada kullanılan YSA modeli Şekil 3.25'te gösterilmektedir. Şekil 3.25 incelendiğinde giriş katmanında 3 farklı giriş bulunmaktadır. Gizli katmanlar 4 tane olup 1024 birime ayrılmış nöron içermektedir. Çıkış katmanı ise 10 sınıflandırmaya ayrılmıştır.



Şekil 3.25. YSA'nın modeli

4. ARAŞTIRMA BULGULARI ve TARTIŞMA

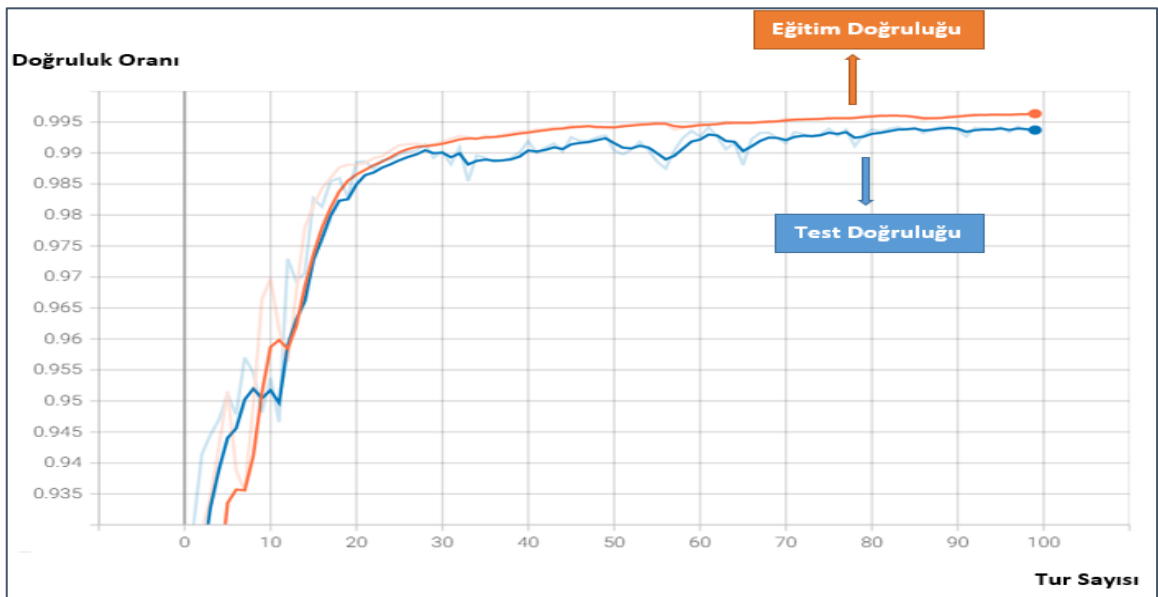
5000 Diyarbakır karpuzu görüntüsünün fotoğrafları çekilip hassas tartım terazileri ile ağırlıkları tartılmıştır. Her karpuz görüntüsü numaralandırılıp veriler kayıt altına alınmıştır. Görüntüler için U-net mimarisi ile çalışma yapılmıştır. Görüntülerden elde edilen verilerden ise YSA ile eğitim yapılmıştır.

4.1. U-Net Sonuçları

U-Net mimarisinin görüntüler üzerindeki yüksek doğruluk sağlamasından dolayı görüntüleri hızlı ayıklamaktadır. Bu yüzden U-Net mimarisi tez çalışmasında tercih edilmiştir. Karpuz görüntülerinin şeklinin algılanmasında yaptığı başarılı segmentasyonlardan dolayı eğitim sonuçları beklenenden iyi bir sonuç elde etmiştir.

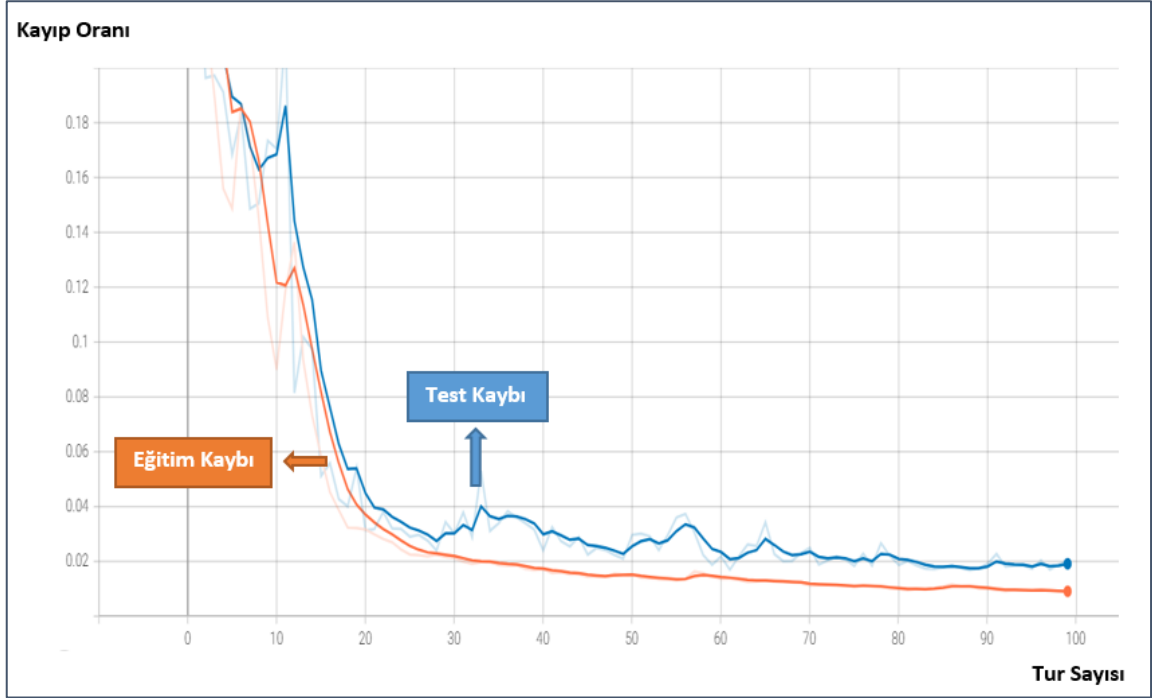
U-net sonuçlarına göre eğitim doğruluğu %99,65, eğitim kayıp oranı %0,88 olmuştur. Test doğruluğu %99,35 olurken test kaybı ise %2.01 oranında çıkmıştır.

Şekil 4.1’de tur sayısına göre doğruluk yüzdeleri gösterilmektedir. Görüntülerin tur sayısı ile orantılı olarak başarı yüzdeleri artmaktadır. 60 tur sayısından sonra model sabit bir şekilde yüzdesini koruyarak çalışmasının bitirmiştir. Çalışma sonucunda eğitilen görüntüler %99,65 oranında eğitim doğruluğu ile büyük bir yüzde yakalamıştır.



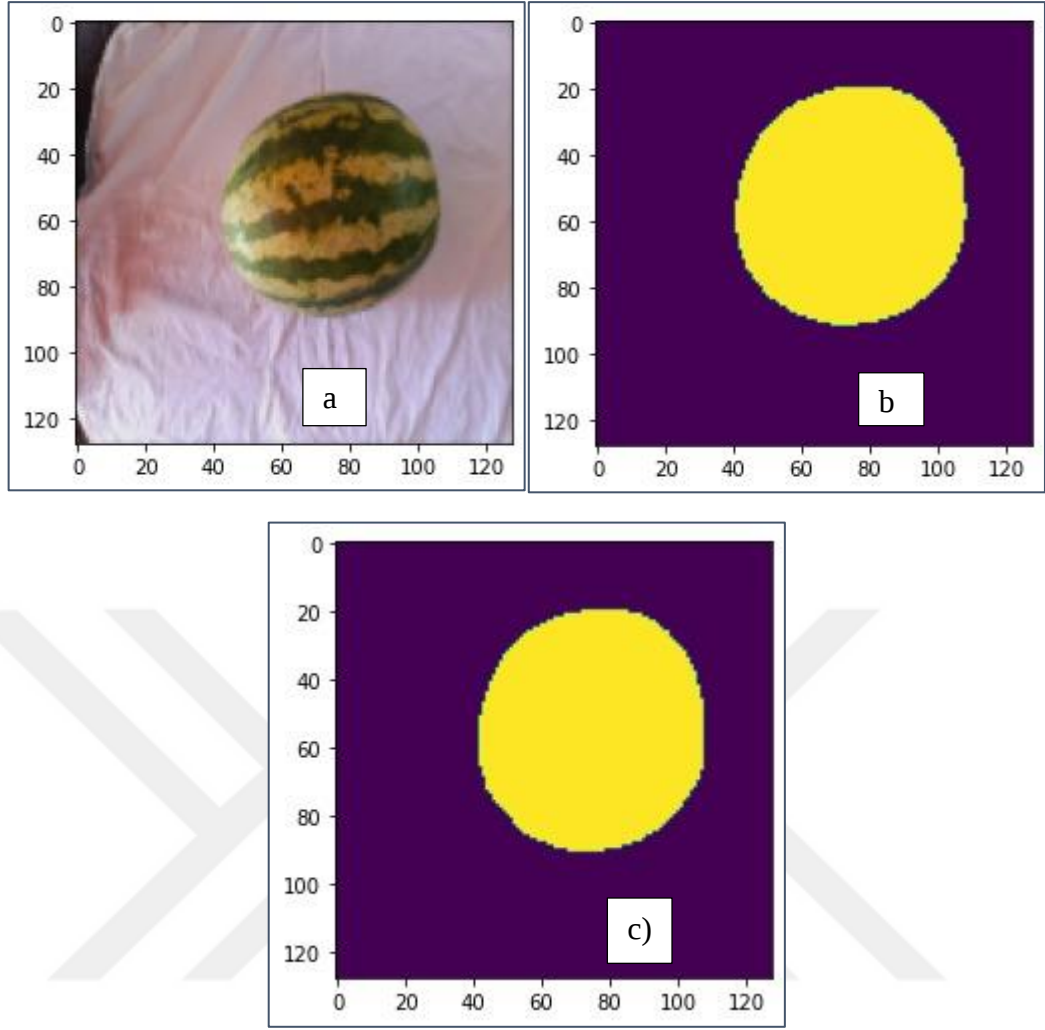
Şekil 4.1. Tur sayısına göre doğruluk grafiği

Şekil 4.2’ deki tur sayısına göre kayıp grafiğini incelediğimizde sıfıra yaklaşan bir hassasiyetle görüntülerin analiz sonuçları başarılı bir şekilde eğitimin yapıldığını göstermektedir. %0,88 oranında eğitim kayıp oranı elde edilmiştir. Tur sayısına göre büyük bir başarı ile görüntüler segmentasyonunu tamamlamıştır.



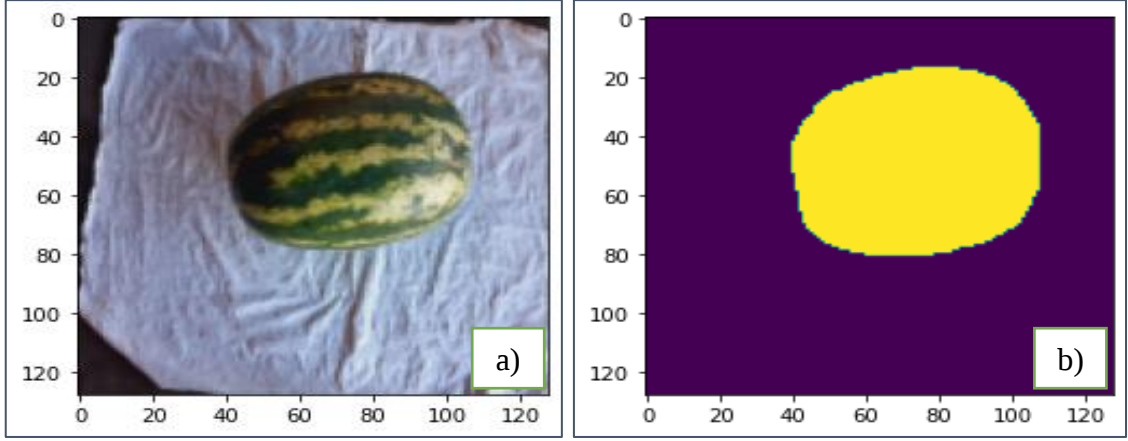
Şekil 4.2. Tur sayısına göre kayıp grafiği

Şekil 4.5’te U-Net modelinin tahmin ettiği karpuz segmentasyon görüntüsü gösterilmektedir. Şekil 4.5’teki görüntüler incelendiğinde modelin yaptığı tahminin başarılı olduğu görülmektedir. Segmentasyon görüntüsünün, orijinal görüntü ile maske görüntüsüyle karşılaştırıldığında modelin başarılı bir şekilde karpuz görüntüsünü ortaya çıkardığı tespit edilmiştir. Tüm karpuzlarda segmentasyon tamamlanmıştır. Görüntüler büyük bir oranda karpuz şeklini algılamıştır.

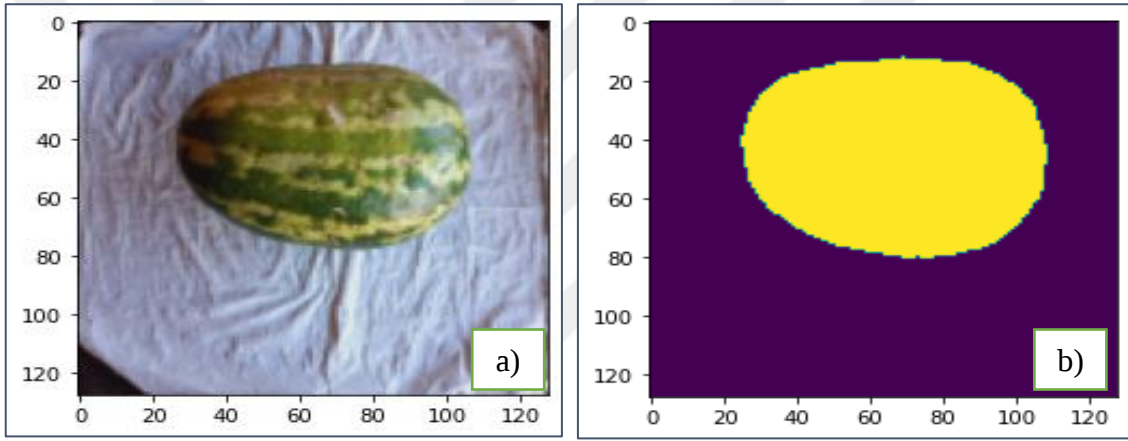


Şekil 4.3. Orjinal görüntü boyutu 128x128 olarak değiştirilmiş b) Maskesi c) U-Net mimarisinin tahmin ettiği segmentasyon

Şekil 4.4 ve Şekil 4.5'te U-Net modelinin yaptığı diğer tahminler gösterilmektedir. Şekil 4.4 ve Şekil 4.5 incelendiğinde karpuz şeklinin başarılı bir biçimde segmentasyon görüntülerini tahmin ettiği görülmektedir.



Şekil 4.4. a) 80 numaralı karpuzun orijinal görüntü boyutu 128x128 olarak değiştirilmiş b) U-Net mimarisinin tahmin ettiği segmentasyonu



Şekil 4.5. a) 95 numaralı karpuzun orijinal görüntü boyutu 128x128 olarak değiştirilmiş b) U-Net mimarisinin tahmin ettiği segmentasyonu

4.2. Yapay Sinir Ağları Sonuçları

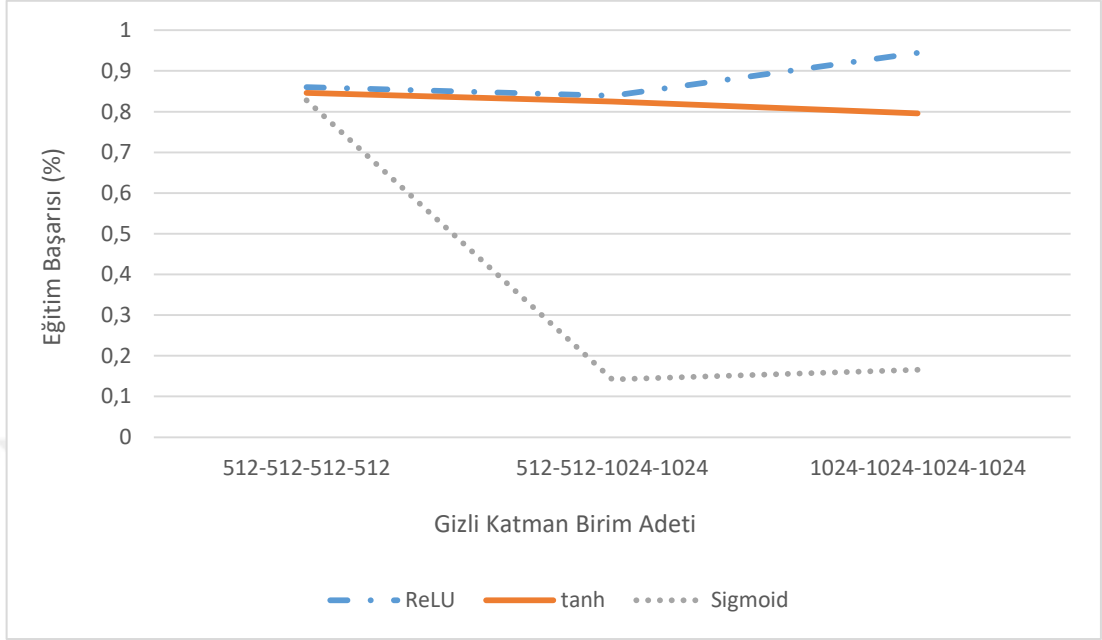
Yapay sinir ağı ile eğitilen veriler yüzde olarak büyük bir başarı elde etmiştir. U-Net görüntülerinden elde edilen veriler yapay sinir ağında eğitilmiştir. Çizelge 4.1. de en iyi sonuç elde edilinceye kadar yapay sinir ağlarında eğitimi yapılan parametreler gösterilmektedir. 9 farklı yapay sinir ağı eğitimi yapılmıştır. Çıkış katmanında softmax aktivasyon fonksiyonu tercih edilmiştir. Softmax aktivasyon fonksiyonunun kullanılması-
nın nedeni ikiden fazla sınıflandırma işlemlerinde kullanılmasıdır. Gizli katmanlarda en iyi sonucun bulunması için yapay sinir ağlarında en çok kullanılan ReLU, sigmoid ve tanh olmak üzere 3 farklı aktivasyon fonksiyonu kullanılmıştır.

Çizelge 4. 1. Eğitim Sonuçları

Deneme	Gizli katman sayısı ve birim sayıları	aktivasyon fonksiyonu	Çıkış fonksiyonu	Eğitim Sonuçları
1	4 tane [512-512-512-512]	ReLU	Softmax	Eğitim Kaybı: 0.2598 Eğitim Doğruluğu: 0.8600 Test Kaybı: 0.2759 Test Doğruluğu: 0.8880
2	4 tane [512-512-512-512]	tanh	Softmax	Eğitim Kaybı: 0.5767 Eğitim Doğruluğu: 0.8563 Test Kaybı: 0.5887 Test Doğruluğu: 0.8080
3	4 tane [512-512-512-512]	Sigmoid	Softmax	Eğitim Kaybı: 0.4510 Eğitim Doğruluğu: 0.8283 Test Kaybı: 0.2443 Test Doğruluğu: 0.8735
4	4 tane [512-512-1024-1024]	ReLU	Softmax	Eğitim Kaybı: 0.4591 Eğitim Doğruluğu: 0.8390 Test Kaybı: 0.3813 Test Doğruluğu: 0.8140
5	4 tane [512-512-1024-1024]	tanh	Softmax	Eğitim Kaybı: 1.0663 Eğitim Doğruluğu: 0.8247 Test Kaybı: 0.7377 Test Doğruluğu: 0.8105
6	4 tane [512-512-1024-1024]	Sigmoid	Softmax	Eğitim Kaybı: 2.2805 Eğitim Doğruluğu: 0.1420 Test Kaybı: 2.2919 Test Doğruluğu: 0.0840
7	4 tane [1024-1024-1024-1024]	ReLU	Softmax	Eğitim Kaybı: 0.1571 Eğitim Doğruluğu: 0.9443 Test Kaybı: 0.2535 Test Doğruluğu: 0.9260
8	4 tane [1024-1024-1024-1024]	tanh	Softmax	Eğitim Kaybı: 1.2142 Eğitim Doğruluğu: 0.7957 Test Kaybı: 1.2133 Test Doğruluğu: 0.7950
9	4 tane [1024-1024-1024-1024]	Sigmoid	Softmax	Eğitim Kaybı: 2.2289 Eğitim Doğruluğu: 0.1657 Test Kaybı: 2.2354 Test Doğruluğu: 0.1645

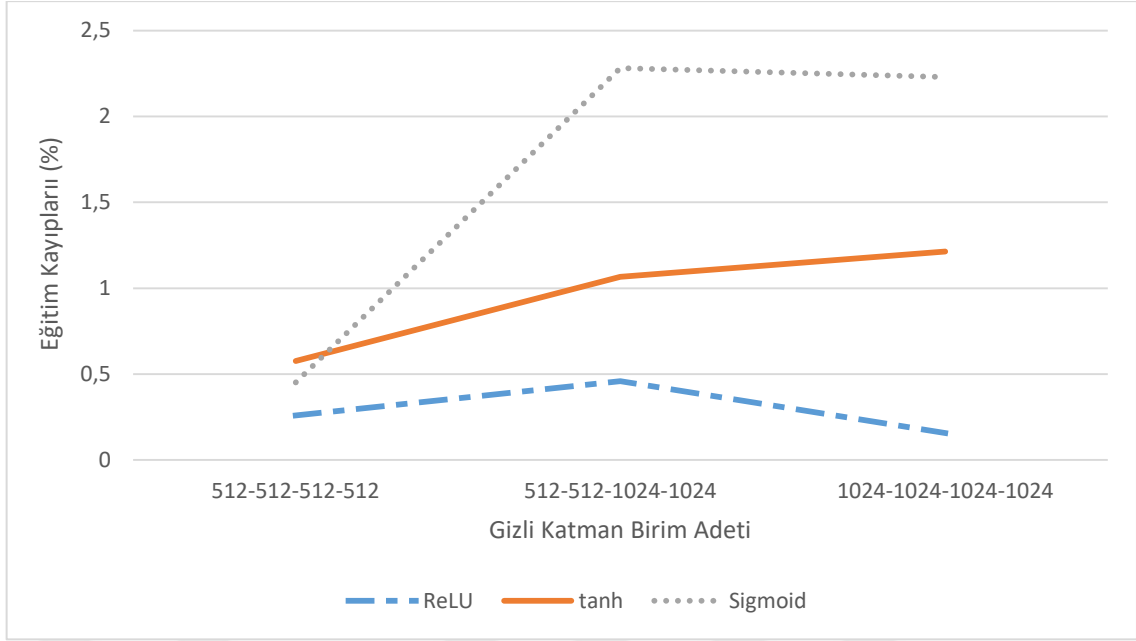
Eğitilen ağlar içinde aktivasyon fonksiyonu olarak en başarıyı ReLU aktivasyon fonksiyonu göstermiştir. ReLU aktivasyon fonksiyonunu en büyük başarısı nöronların aynı anda etkinleştirilmemesidir. Bu yüzden oranları hesaplarken çok verimli olmaktadır. Birim sayısı olarak dört gizli katmanı 1024 alındığında en iyi sonucu göstermiştir. Şekil 4.6’da aktivasyon fonksiyonlarının gösterdikleri eğitim başarısı karşılaştırılmıştır.

Birim sayısı artıkça ReLU artarken, tanh fonksiyonu hafif bir azalış göstermiştir. Sigmoid fonksiyonu ise sert bir azalış halinde değer kaybetmiştir.



Şekil 4.6. Aktivasyon fonksiyonlarının eğitim başarıları

Aktivasyon fonksiyonlarının eğitim boyunca gösterdikleri kayıp oranları şekil 4.7’de gösterilmektedir. Şekil 4.7’deki grafik incelendiğinde kayıp oranlarına göre en büyük başarıyı ReLU aktivasyon fonksiyonu göstermiştir. ReLU fonksiyonu istikrarlı bir şekilde değeri düşük tutmuştur. En kötü performansı sigmoid aktivasyon fonksiyonu göstermiştir. Gizli katman birim sayısı artıkça kayıp oranı da buna paralel artış göstermiştir. Tanh aktivasyon fonksiyonu ise hafif artışlarla sigmoid fonksiyonundan sonra kötü bir performans göstermiştir. ReLU fonksiyonun başarıları bir kez daha kanıtlanmıştır.



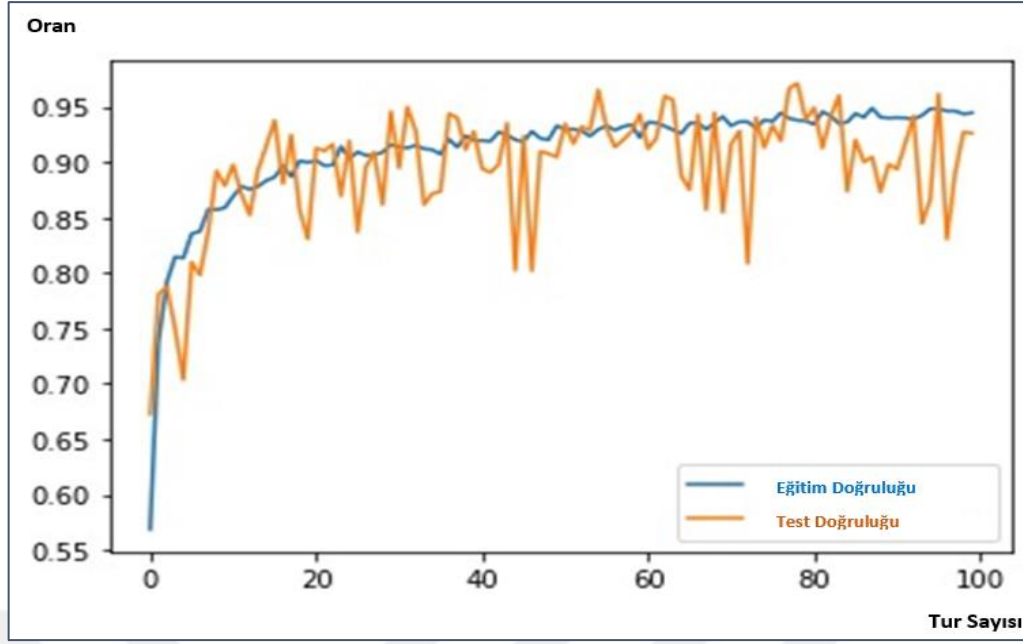
Şekil 4.7. Aktivasyon fonksiyonlarının eğitim kayıp oranları

4.3. En İyi Sonucun Değerlendirilmesi

Yapay sinir ağlarında yapılan dokuz deneme içinde Çizelge 3.3'te 7 numaralı deneme en iyi sonucu vermiştir. Hem test doğruluk yüzdesi hem de eğitim doğruluk yüzdesi başarı olarak yüksek çıkmıştır.

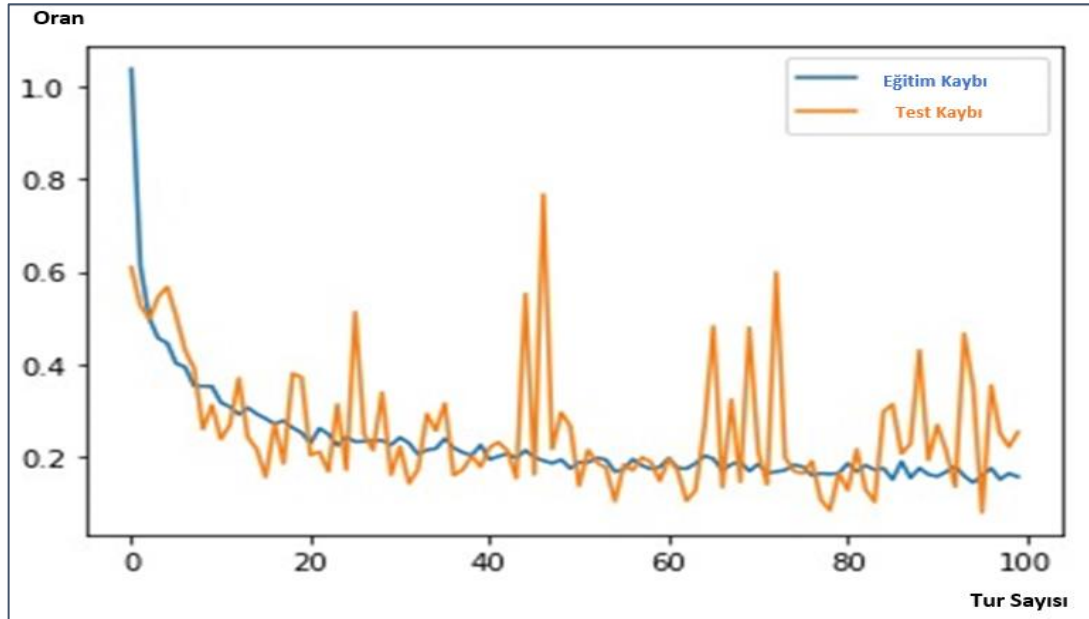
Yapay sinir ağında elde edilen test sonuçları; test kayıp oranı %25,35 ve test doğruluk oranı ise %92,59 bulunmuştur. Eğitim için kayıp oranı %15,71 ve doğruluk oranı ise %94,43 bulunmuştur.

Şekil 4.8'de verilen grafikte eğitim doğruluğu ve testler için doğruluk oranı verilmiştir. Sonuçlar incelendiğinde eğitim yüzdesinin yüksek olması testin başarıyla sonuçlandığını göstermektedir. Karpuz görüntülerinden elde edilen verilerin doğruluk oranı %94,43 olarak büyük bir hassasiyetle verileri tahmin etmiştir. Tur sayısı artıkça eğitim doğruluğu istikrarlı bir şekilde artarken test doğruluğu dalgalı bir şekilde azalış ve artış şeklinde ilerlemiştir. Son tur sayısında eğitim doğruluğu maksimum noktaya ulaşmıştır.



Şekil 4.8. Doğruluk başarıları

Kayıp grafiği ise şekil 4.9’da gösterilmektedir. Kayıp oranlarının düşük olması yapay sinir ağı eğitiminin başarıyla tahminler yaptığını göstermektedir. Tur sayısı artarken eğitim kayıp oranı istikrarlı bir şekilde düşüşe geçerken test kayıp oranı dalgalı bir şekilde düşüşünü tamamlamıştır. Kayıp oranlarının düşük olması doğru tahmin sayısını yükseltmiştir.

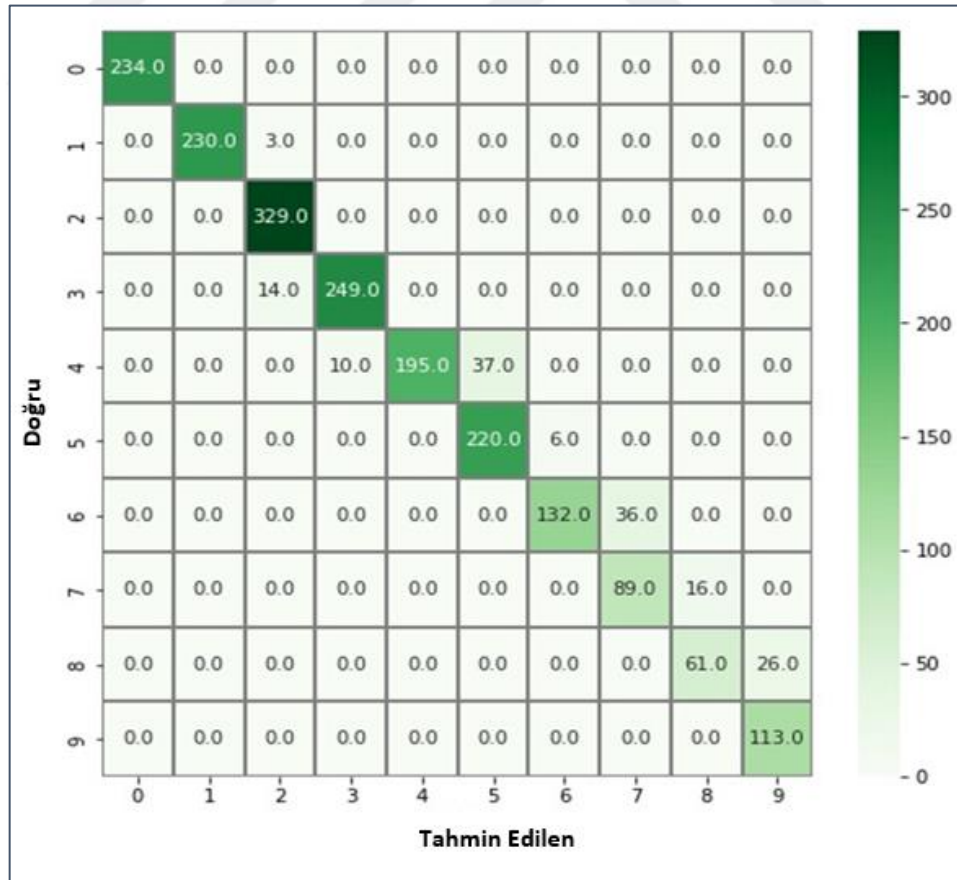


Şekil 4.9. Kayıp oranları

Eğitim yapılırken sınıflandırmaların doğruluk sayısı incelenmiştir. Şekil 4.10'da yapay sinir ağında eğitilen sınıflandırmanın tahmin edilen doğru görüntü ve yanlış görüntü sayısı verilmiştir. Şekil 4.10 incelendiğinde; 0 numaralı sınıflandırma 234, 1 numaralı sınıflandırma 230, 2 numaralı sınıflandırma 329, 3 numaralı sınıflandırma 249, 4 numaralı sınıflandırma 195, 5 numaralı sınıflandırma 220, 6 numaralı sınıflandırma 132, 7 numaralı sınıflandırma 89, 8 numaralı sınıflandırma 61 ve 9 numaralı sınıflandırma ise 113 görüntüyü doğru tahmin etmiştir.

2 numaralı sınıflandırmanın sonucunun yüksek çıkması karpuz sayısının fazla olmasından, 7, 8 ve 9 numaralı sınıflandırmalardaki sonucun düşük çıkması ise karpuz sayısının az olmasından kaynaklanmaktadır.

Yanlış görüntü sayısı incelendiğinde 1 numaralı satırda 3 yanlış, 3 numaralı satırda 14 yanlış, 4 numaralı satırda 47 yanlış, 5 numaralı satırda 6 yanlış, 6 numaralı satırda 36 yanlış, 7 numaralı satırda 16 yanlış ve 8 numaralı satırda 26 yanlış görüntü elde edilmiştir. Yanlış görüntü sayısının az olması eğitim sonuçlarının başarılı olduğu anlamına gelmektedir.



Şekil 4.10. Tahmin edilen doğru ve yanlış görüntü sayısı

5. SONUÇLAR ve ÖNERİLER

5.1. Sonuç

5000 Diyarbakır yöresi karpuzu ile yapılan saha çalışmasında elde edilen sonuçlar değerlendirilmiştir. Bu çalışmada görüntülerden elde edilen verilerle U-Net ve yapay sinir ağları ile eğitimi yapılan karpuzlarla ilgili olarak aşağıdaki sonuçlara ulaşılmıştır:

- U-Net mimarisinde segmentasyonu yapılan karpuz görüntüleri %99,65 ile büyük bir oranda görüntüleri doğru tahmin etmiştir.
- U-Net görüntü sonuçlarında karpuz şeklini başarılı bir şekilde ortaya çıkarabilmiştir.
- 9 farklı şekilde YSA eğitimi yapılmıştır. En iyi sonucu 1024 birimlere ayrılan 4 gizli katman ile eğitilen mimari elde etmiştir.
- 3 Aktivasyon fonksiyonunun karşılaştırıldığı yapay sinir ağında ReLU aktivasyon fonksiyonu mimaride en iyi başarıyı göstermiştir.
- Yapay sinir ağı ile yapılan eğitimde test doğruluk oranı %92,59 bulunmuştur. Eğitim doğruluk oranı ise %94,43 bulunmuştur. Görüntülerden elde edilen verilerin büyük bir kısmı doğru tahminde bulunmuştur.
- Karpuz sayısı fazla olan sınıflandırma doğru tahmin sayısını yüksek çıkarırken karpuz sayısı az olan sınıflandırmalar ise doğru tahmin sayısını düşük tahmin etmiştir.

5.2. Öneriler

Bu çalışma ile önemli Diyarbakır ve çevre illerinde yöresel yaz mevsimine özgü ürünü olan karpuzun ağırlığının çekilen görüntülerden yola çıkılarak tespit edilmesi hem üreticilerin hem de satıcıya ulaşana kadar geçen kişilerin işlerinin kolaylaşacağı düşünülmektedir. Son yılların popüler çalışma alanlarından biri olan DÖ yöntemlerinden yararlanarak karpuzların ağırlığını tespit etme yeni bir çalışma alanı olarak karşımıza çıkmaktadır. Karpuz tarlalarında, tezgahlarda ve pazarlarda satılmakta olan karpuzların yerinde ağırlıkların tahmin edilmesiyle dijital tarım alanında yapılan ve gelecekte yapılacak olan çalışmalara ışık tutacağı düşünülmektedir. Bu çalışmanın sonuçlanmasıyla birlikte insansız hava araçlarından veya drone tarzı görüntü ve video kaydeden cihazlardan alınacak görüntülerin hazırlanan DÖ mimarisiyle çalıştırıldığında tarlalarda

bulunacak karpuzların ağırlıklarının tespit edilmesi çalışmaları yürütülebilecektir. Böylece tarlada tahmini fiyatlarla satılan karpuzların bu görüntülerin işlenmesiyle elde edilen ağırlıklar baz alınarak üreticilerin doğabilecek mağduriyetlerin önüne geçebilecektir. Bu çalışmadan ilham alacak araştırmacılar farklı ürünler için ağırlık tahminleri üzerine çalışmalar yürüteceklerdir.



KAYNAKLAR

Abacı, A., 2022, Sinir hücrelerinin yapısı ve nöron çeşitleri [online], <http://www.biyodoc.com/Sinir-sistemi-sinir-hucrelerinin-yapisi-ve-noron-cesitleri.html> [Ziyaret Tarihi: 11 Mart 2022].

Abiodun, O. I., Jantan, A., Omolara, A. E., Dada, K. V., Mohamed, N. A., & Arshad, H., 2018, State-of-the-art in artificial neural network applications: A survey, *Heliyon*, 4(11), e00938

Abu-Mahfouz, I. A., 2005, A comparative study of three artificial neural networks for the detection and classification of gear faults, *International Journal of General Systems*, 34(3), 261-277.

Adeola, E. O., 2020, Knee Bone Segmentation from MRI Images Using a Deep Learning Model, Pace University.

Adhikari, S., Unit, D., Shrestha, B., & Baiju, B., 2018, Tomato Plant Diseases Detection System, vol. I, 81-86.

Anonim, 2018, Diyarbakır Karpuzu [online], <https://www.kulturportali.gov.tr/portal/diyarbakirkarpuzu> [Ziyaret Tarihi: 12 Mart 2022].

Anonim, 2018, Yapay Sinir Ağı Eğitimi – Çok Katmanlı Perceptron (Multi Layer Perceptron) [online], Devhunter, <https://devhunteryz.wordpress.com/2018/07/05/yapay-sinir-agi-egitimi-cok-katmanli-perceptronmulti-layer-perceptron/> [Ziyaret Tarihi: 20 Mart 2022].

Anonymous, 2022, About – OpenCV [online], <https://opencv.org/about/> [Ziyaret Tarihi: 01 Temmuz 2022].

Anonymous, 2022, Anaconda (Python distribution), [online], From Wikipedia [online], [https://en.wikipedia.org/wiki/Anaconda_\(Python_distribution\)#:~:text=Anaconda%20is%20a%20distribution%20of,and%20Travis%20Oliphant%20in%202012](https://en.wikipedia.org/wiki/Anaconda_(Python_distribution)#:~:text=Anaconda%20is%20a%20distribution%20of,and%20Travis%20Oliphant%20in%202012) [Ziyaret Tarihi: 01 Temmuz 2022].

Anonymous, 2022, Matplotlib: Visualization with Python [online], <https://matplotlib.org/> [Ziyaret Tarihi: 01 Temmuz 2022].

Anonymous, 2022, Get started with TensorBoard [online], https://tensorflow.google.cn/tensorboard/get_started?hl=en [Ziyaret Tarihi: 01 Temmuz 2022].

Anonymous, 2022, Google Semantic Segmentation Network Deep Lab v3 plus design ideas and architecture, [online], Programmer Sought, <https://programmersought.com/article/67484433091/> [Ziyaret Tarihi: 30 Mart 2022].

Anonymous, 2022, Home – Spyder IDE, [online], <https://www.spyder-ide.org/> [Ziyaret Tarihi: 01 Temmuz 2022].

Anonymous, 2022, Image processing in Python [online], <https://scikit-image.org/> [Ziyaret Tarihi: 01 Temmuz 2022].

Anonymous, 2022, Matlab - MathWorks - Matlab & Simulink [online], <https://www.mathworks.com/products/matlab.html> [Ziyaret Tarihi: 01 Temmuz 2022].

Anonymous, 2022, Neural network models (supervised) [online], https://scikit-learn.org/stable/modules/neural_networks_supervised.html [Ziyaret Tarihi: 17 Mart 2022].

Anonymous, 2022, os - Miscellaneous operating system interfaces [online], <https://docs.python.org/3/library/os.html> [Ziyaret Tarihi: 01 Temmuz 2022].

Anonymous, 2022, Random - Generate pseudo-random numbers [online], <https://docs.python.org/3/library/random.html> [Ziyaret Tarihi: 01 Temmuz 2022].

Anonymous, 2022, seaborn: statistical data visualization [online], <https://seaborn.pydata.org/> [Ziyaret Tarihi: 01 Temmuz 2022].

Anonymous, 2022, The Best Introduction to Deep Learning - A Step by Step Guide [online], Simplilearn, Purdue University, <https://www.simplilearn.com/tutorials/deep-learning-tutorial/introduction-to-deep-learning> [Ziyaret Tarihi: 10 Mart 2022].

Anonymous, 2022, What is NumPy?, [online], <https://numpy.org/doc/stable/user/whatisnumpy.html> [Ziyaret Tarihi: 01 Temmuz 2022].

Anonymous, 2022, What is Python? – What is Python Used For? [online], <https://codingcompiler.com/what-is-python/> [Ziyaret Tarihi: 10 Temmuz 2022].

Anonymous, 2022, What is Python? Executive Summary [online], <https://www.python.org/doc/essays/blurb/> [Ziyaret Tarihi: 01 Temmuz 2022].

Anonymous, 2022, What is Pandas in python? [online], <https://www.tutorialspoint.com/what-is-pandas-in-python#> [Ziyaret Tarihi: 01 Temmuz 2022].

Arsenovic, M., Karanovic, M., Sladojevic, S., Anderla, A., & Stefanovic, D., 2019, Solving current limitations of deep learning based approaches for plant disease detection, *Symmetry*, 11(7), 939.

Ataseven, B., 2013, Yapay Sinir Ağları İle Öngörü Modellemesi, *Öneri Dergisi*, 10 (39), 101-115.

Atila, Ü., Uçar, M., Akyol, K., & Uçar, E., 2021, Plant leaf disease classification using EfficientNet deep learning model, *Ecological Informatics*, 61, 101182.

Bhatia, V., 2021, U-NET Implementation from Scratch using TensorFlow, [online], Medium, <https://medium.com/geekculture/u-net-implementation-from-scratch-using-tensorflow-b4342266e406> [Ziyaret Tarihi: 20 Temmuz 2022].

Bora, D. J., 2017, Importance of image enhancement techniques in color image segmentation: A comprehensive and comparative study, arXiv preprint arXiv:1708.05081.

Buayai, P., Piewthongngam, K., Leung, C. K., & Saikaew, K. R., 2019, Semi-automatic pig weight estimation using digital image analysis, *Applied Engineering in Agriculture*, 35(4), 521-534.

Carrio, A., Sampedro, C., Rodriguez-Ramos, A., & Campoy, P., 2017, A review of deep learning methods and applications for unmanned aerial vehicles, *Journal of Sensors*.

Chandra, A. ve Matcha, N., 2020, A 2021 guide to Semantic Segmentation [online], <https://nanonets.com/blog/semantic-image-segmentation-2020/> [Ziyaret Tarihi: 26 Mart 2022].

Chang, S., Lee, U., Hong, M. J., Jo, Y. D., & Kim, J. B., 2021, Time-Series Growth Prediction Model Based on U-Net and Machine Learning in Arabidopsis, *Frontiers in plant science*, 12, 721512-721512.

Chen, C., Qin, C., Qiu, H., Tarroni, G., Duan, J., Bai, W., & Rueckert, D., 2020, Deep learning for cardiac image segmentation: a review, *Frontiers in Cardiovascular Medicine*, 25.

Chen, L. C., Papandreou, G., Kokkinos, I., Murphy, K., & Yuille, A. L., 2017, Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs, *IEEE transactions on pattern analysis and machine intelligence*, 40(4), 834-848.

Chiao, J. Y., Chen, K. Y., Liao, K. Y. K., Hsieh, P. H., Zhang, G., & Huang, T. C., 2019, Detection and classification the breast tumors using mask R-CNN on sonograms, *Medicine*, 98(19).

Coşkun, M. , Yıldırım, Ö. , Uçar, A. & Demır, Y., 2017, An overview of popular deep learning methods, *European Journal of Technique (EJT)*, 7 (2), 165-176 .

Çağlar, O.. 2022, Aktivasyon Fonksiyonu Nedir?[online], <https://www.elektrikport.com/haber-roportaj/aktivasyon-fonksiyonu-nedir/23511#ad-image-0> [Ziyaret Tarihi: 16 Mart 2022].

Çıtlak, A. & Demirpolat, A. B., 2021, Effects of Different Turbulators on Heat Transfer in Smoke Tube Boilers and Modeling of These Effects with Machine Learning Algorithms, *Journal of the Institute of Science and Technology*, 11 (1), 474-489.

Denil, M., Shakibi, B., Dinh, L., Ranzato, M. A., & De Freitas, N., 2013, Predicting parameters in deep learning, *Advances in neural information processing systems*, 26.

Devi, P. K., 2020, Image Segmentation K-Means Clustering Algorithm for Fruit Disease Detection Image Processing, *In 2020 4th International Conference on Electronics, Communication and Aerospace Technology*, 861-865.

Egmont-Petersen, M., de Ridder, D., & Handels, H., 2002, Image processing with neural networks—a review. *Pattern recognition*, 35(10), 2279-2301.

Erickson, B. J., Korfiatis, P., Akkus, Z., Kline, T., & Philbrick, K., 2017, Toolkits and libraries for deep learning, *Journal of digital imaging*, 30(4), 400-405.

Ferentinos, K. P., 2018, Deep learning models for plant disease detection and diagnosis, *Computers and electronics in agriculture*, 145, 311-318.

Fountas, S., Espejo-Garcia, B., Kasimati, A., Mylonas, N., & Darra, N., 2020, The future of digital agriculture: technologies and opportunities, *IT professional*, 22(1), 24-28.

He, K., Gkioxari, G., Dollár, P., & Girshick, R., 2017, Mask r-cnn, *In Proceedings of the IEEE international conference on computer vision*, 2961-2969.

Ibtehaz, N., & Rahman, M. S., 2020, MultiResUNet: Rethinking the U-Net architecture for multimodal biomedical image segmentation, *Neural Networks*, 121, 74-87.

Jinsakul, N., Tsai, C. F., Tsai, C. E., & Wu, P., 2019, Enhancement of deep learning in image classification performance using xception with the swish activation function for colorectal polyp preliminary screening, *Mathematics*, 7(12), 1170.

Kamilaris, A., & Prenafeta-Boldú, F. X., 2018, Deep learning in agriculture: A survey, *Computers and electronics in agriculture*, 147, 70-90.

Khairunniza-Bejo, S., Mustaffha, S., & Ismail, W. I. W., 2014, Application of artificial neural network in predicting crop yield: A review, *Journal of Food Science and Engineering*, 4(1), 1.

Kharwal, A., 2021, Tqdm in Python (Tutorial) [online], <https://thecleverprogrammer.com/2021/04/08/tqdm-in-python-tutorial/> [Ziyaret Tarihi: 01 Temmuz 2022].

Kızrak, A., 2019, Derin Öğrenme İçin Aktivasyon Fonksiyonlarının Karşılaştırılması [online], Medium, <https://ayyucekizrak.medium.com/derin-%C3%B6%C4%9Frenme-i%C3%A7in-aktivasyon-fonksiyonlar%C4%B1n%C4%B1n-kar%C5%9F%C4%B1la%C5%9Ft%C4%B1r%C4%B1lmas%C4%B1-cee17fd1d9cd> [Ziyaret Tarihi: 14 Mart 2022].

Kienbaum, L., Correa Abondano, M., Blas, R., & Schmid, K., 2021, DeepCob: precise and high-throughput analysis of maize cob geometry using deep learning with an application in genebank phenomics, *Plant Methods*, 17(1), 1-19.

Kim, J., & Lee, S., 2017, Deep learning of human visual sensitivity in image quality assessment framework, *In Proceedings of the IEEE conference on computer vision and pattern recognition*, 1676-1684.

Koirala, A., Walsh, K. B., Wang, Z., & McCarthy, C., 2019, Deep learning—Method overview and review of use for fruit detection and yield estimation, *Computers and electronics in agriculture*, 162, 219-234.

Kukreja, H., Bharath, N., Siddesh, C. S., & Kuldeep, S., 2016, An introduction to artificial neural network, *Int J Adv Res Innov Ideas Educ*, 1, 27-30.

Kumar, B., 2021, What is Scikit Learn in Python [online], <https://pythonguides.com/what-is-scikit-learn-in-python/> [Ziyaret Tarihi: 01 Temmuz 2022].

Kumar, M. J., Kumar, D. G. R., & Reddy, R. V. K., 2014, Review on image segmentation techniques, *International Journal of Scientific Research Engineering & Technology*, 3(6), 993-997.

Kumari, N., & Belwal, R., 2021, Hybridized approach of image segmentation in classification of fruit mango using BPNN and discriminant analyzer, *Multimedia Tools and Applications*, 80(4), 4943-4973.

Labs, C., 2018, Understanding U-Net Architecture For Image Segmentation [online], Medium, <https://medium.com/@CereLabs/understanding-u-net-architecture-for-image-segmentation-74bef8caefee> [Ziyare Tarihi: 24 Mart 2022].

Lin, Z., & Guo, W., 2020, Sorghum panicle detection and counting using unmanned aerial system images and deep learning, *Frontiers in Plant Science*, 11, 1346.

Lu, W., Du, R., Niu, P., Xing, G., Luo, H., Deng, Y., & Shu, L., 2022, Soybean Yield Preharvest Prediction Based on Bean Pods and Leaves Image Recognition Using Deep Learning Neural Network Combined With GRNN, *Frontiers in Plant Science*, 12, 791256.

Minaee, S., Boykov, Y. Y., Porikli, F., Plaza, A. J., Kehtarnavaz, N., & Terzopoulos, D., 2021, Image segmentation using deep learning: A survey, *IEEE transactions on pattern analysis and machine intelligence*.

Mwiti, D., 2022, The Definitive Guide to Semantic Segmentation for Deep Learning in Python [Online], <https://cnvrg.io/semantic-segmentation/> [Ziyare Tarihi: 23 Mart 2022].
Nasiri, S., Khosravani, M. R., & Weinberg, K., 2017, Fracture mechanics and mechanical fault detection by artificial intelligence methods: A review, *Engineering Failure Analysis*, 81, 270-293.

Ngo, T., 2021, Deep Learning for Wireless Signal Parameter Predictions (Doctoral dissertation).

Nwankpa, C., Ijomah, W., Gachagan, A., & Marshall, S., 2018, Activation functions: Comparison of trends in practice and research for deep learning.

Ozdogan, B. , Gacar, A. & Aktas, H., 2017, Digital agriculture practices in the context of agriculture 4.0, *Journal of Economics Finance and Accounting*, 4 (2) , 186-193.

Pallaw, 2019, Understand Semantic segmentation with the Fully Convolutional Network U-Net step-by-step, [online], Medium, <https://pallawids.medium.com/understand-semantic-segmentation-with-the-fully-convolutional-network-u-net-step-by-step-9d287b12c852> [Ziyare Tarihi: 20 Haziran 2022].

Reyes-Yanes, A., Martinez, P., & Ahmad, R., 2020, Real-time growth rate and fresh weight estimation for little gem romaine lettuce in aquaponic grow beds, *Computers and Electronics in Agriculture*, 179, 105827.

Ronneberger, O., Fischer, P., & Brox, T., 2015, U-net: Convolutional networks for biomedical image segmentation, *In International Conference on Medical image computing and computer-assisted intervention*, 234-241.

Saleem, M. H., Potgieter, J., & Arif, K. M., 2019, Plant disease detection and classification by deep learning, *Plants*, 8(11), 468.

Sant'Ana, D. A., Pache, M. C. B., Martins, J., Soares, W. P., de Melo, S. L. N., Garcia, V., ... & Pistori, H., 2021, Weighing live sheep using computer vision techniques and regression machine learning, *Machine Learning with Applications*, 5, 100076.

Seera, M., Lim, C. P., Ishak, D., & Singh, H., 2011, Fault detection and diagnosis of induction motors using motor current signature analysis and a hybrid FMM–CART model, *IEEE transactions on neural networks and learning systems*, 23(1), 97-108.

Shah K., 2021, U-Net: Convolutional Networks for Biomedical Image Segmentation, [online], Medium, <https://medium.com/projectpro/u-net-convolutional-networks-for-biomedical-image-segmentation-435699255d26> [Ziyare Tarihi: 28 Haziran 2022].

Sharma, S., Sharma, S., & Athaiya, A., 2017, Activation functions in neural networks, *Towards Data Science*, 6(12), 310-316.

Singh, R., Digumarthy, S. R., Muse, V. V., Kambadakone, A. R., Blake, M. A., Tabari, A., ... & Kalra, M. K., 2020, Image quality and lesion detection on deep learning reconstruction and iterative reconstruction of submillisievert chest and abdominal CT, *American Journal of Roentgenology*, 214(3), 566-573.

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*.

Soother, D. K., Daudpoto, J., Harris, N. R., Hussain, M., Mehran, S., Kalwar, I. H., ... & Memon, T. D., 2021, The importance of feature processing in deep-learning-based condition monitoring of motors, *Mathematical Problems in Engineering*.

Strauch, Y. L., 2019, Evolving Convolutional Neural Network Topologies for Image Recognition.

Tumer, A. E. & Edebali, S., 2015, An Artificial Neural Network Model for Wastewater Treatment Plant of Konya, *International Journal of Intelligent Systems and Applications in Engineering*, 3 (4), 131-135.

Verma, S., Chug, A., Singh, A. P., Sharma, S., & Rajvanshi, P., 2019, Deep learning-based mobile application for plant disease diagnosis: A proof of concept with a case study on tomato plant, *In Applications of image processing and soft computing systems in agriculture*, 242-271.

Villacrés, J. F., & Auat Cheein, F., 2020, Detection and characterization of cherries: A deep learning usability case study in Chile, *Agronomy*, 10(6), 835.

Wang, G., Sun, Y., & Wang, J., 2017, Automatic image-based plant disease severity estimation using deep learning, *Computational intelligence and neuroscience*.

Wang, Z., Shadpour, S., Chan, E., Rotondo, V., Wood, K. M., & Tulpan, D., 2021, ASAS-NANP SYMPOSIUM: Applications of machine learning for livestock body weight prediction from digital images, *Journal of Animal Science*, 99(2).

Xiancheng, W., Wei, L., Bingyi, M., He, J., Jiang, Z., Xu, W., ... & Zhaomeng, S., 2018, Retina blood vessel segmentation using a U-net based Convolutional neural network, *In Procedia Computer Science: International Conference on Data Science*, 8-9.

Yamamoto, K., Guo, W., Yoshioka, Y., & Ninomiya, S., 2014, On plant detection of intact tomato fruits using image analysis and machine learning methods, *Sensors*, 14(7), 12191-12206.

Zhang, C., & Lu, Y., 2021, Study on artificial intelligence: The state of the art and future prospects, *Journal of Industrial Information Integration*, 23, 100224.

Zyro, 2022, Arka plan Silici [online], <https://zyro.com/tr/araclar/arka-plan-silici> [Ziyaret Tarihi: 20 Haziran 2022].

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı : Halil KAYRA
Uyruğu : T.C.

EĞİTİM

Üniversite : Dicle Üniversitesi, Diyarbakır
Yüksek Lisans:

Bitirme Yılı
16.07.2019

UZMANLIK ALANI

Makine Dinamiği ve Teorisi

YABANCI DİLLER

İngilizce

YAYINLAR

KAYRA, H. ve KOÇ, S. (2021). Duvar takip robot navigasyonunun çok katmanlı yapay sinir ağı ile performans tahmini 7. Uluslararası Mardin Artuklu Bilimsel Araştırmalar Kongresi Konferans Bildirisi ISBN: 978-625-8423-02-0 (Yüksek lisans çalışmasından yapılmıştır.)