SEMANTİK SEGMENTASYON VE STEREO GÖRME TEKNİKLERİ KULLANILARAK CANLI HAYVAN AĞIRLIĞININ TAHMİNİ ÜZERİNE BİR ÇALIŞMA

Derviş Kaan Akdik
191402108

YÜKSEK LİSANS TEZİ

Bilgisayar Mühendisliği Anabilim Dalı
Bilgisayar Mühendisliği Yüksek Lisans Programı
Danışman: Dr. Öğr. Üyesi Erdal Güvenoğlu

İstanbul
T.C. Maltepe Üniversitesi
Lisansüstü Eğitim Enstitüsü Enstitüsü
Eylül, 2021

SEMANTİK SEGMENTASYON VE STEREO GÖRME TEKNİKLERİ KULLANILARAK CANLI HAYVAN AĞIRLIĞININ TAHMİNİ ÜZERİNE BİR ÇALIŞMA

Derviş Kaan Akdik
191402108
Orcid: 0000-0003-2317-0304

YÜKSEK LİSANS TEZİ
Bilgisayar Mühendisliği Anabilim Dalı
Bilgisayar Mühendisliği Yüksek Lisans Programı
Danışman: Dr. Öğr. Üyesi Erdal Güvenoğlu

İstanbul
T.C. Maltepe Üniversitesi
Lisansüstü Eğitim Enstitüsü Enstitüsü
Eylül, 2021



JÜRİ VE ENSTİTÜ ONAYI

Bu belge, Yükseköğretim Kurulu tarafından 19.01.2021 tarihli "*Lisansüstü Tezlerin Elektronik Ortamda Toplanması, Düzenlenmesi ve Erişime Açılmasına İlişkin Yönerge*" ile bildirilen 6689 Sayılı Kişisel Verilerin Korunması Kanunu kapsamında gizlenmiştir.

ETİK İLKE VE KURALLARA UYUM BEYANI

Bu belge, Yükseköğretim Kurulu tarafından 19.01.2021 tarihli "*Lisansüstü Tezlerin Elektronik Ortamda Toplanması, Düzenlenmesi ve Erişime Açılmasına İlişkin Yönerge*" ile bildirilen 6689 Sayılı Kişisel Verilerin Korunması Kanunu kapsamında gizlenmiştir.

TEŞEKKÜR

Yüksek lisans eğitimim sürecinde bana emeği geçen tüm bölüm hocalarıma ve bu tezin hazırlanmasında bana yol gösteren değerli danışman hocam Dr. Öğr. Üyesi Erdal Güvenoğlu'na sonsuz teşekkürlerimi sunarım.

Derviş Kaan Akdik

Eylül, 2021

ÖZ

SEMANTİK SEGMENTASYON VE STEREO GÖRME TEKNİKLERİ KULLANILARAK CANLI HAYVAN AĞIRLIĞININ TAHMİNİ ÜZERİNE BİR ÇALIŞMA

Derviş Kaan Akdik
Yüksek Lisans Tezi
Bilgisayar Mühendisliği Anabilim Dalı
Bilgisayar Mühendisliği Yüksek Lisans Programı
Danışman: Dr. Öğr. Üyesi Erdal Güvenoğlu
Maltepe Üniversitesi Lisansüstü Eğitim Enstitüsü, 2021

Hayvancılık insanlık tarihinin başlangıcından beri insanoğlunun vazgeçilmez geçim kaynaklarından biridir. Günümüzde artan insan nüfusuna bağlı olarak hayvancılık endüstrisi de hızla gelişmektedir. Yetiştirilen hayvanların izlenmesi ve gelişim süreçlerinin takip edilmesi verimlilik açısından öneme sahiptir. Bu kapsamda büyükbaş besi çiftliklerinde yetiştirilen hayvanların ağırlıklarının belirlenmesi ve düzenli olarak takip edilmesi gerekir. Büyükbaş hayvancılıkta hayvanların düzenli olarak kantara çıkarılması ve ağırlıklarının kayıt altına alınması zor bir iştir. Bu nedenle bu işlem çoğu zaman düzenli olarak gerçekleştirilemez veya ihmal edilebilir.

Bu çalışmada stereo görme ve semantik segmentasyon yöntemleri beraber kullanılarak büyükbaş hayvanların ağırlığı tahmin edilmeye çalışılmıştır. Özdeş iki kameradan oluşan stereo düzenek ile farklı açılardan hayvan fotoğrafları çekilmiş ve stereo hesabı ile hayvanların kamera düzlemine olan mesafeleri hesaplanmıştır. Segmentasyon yöntemleri ile hayvanların görüntülerde kapladığı alanlar belirlenmiş ve bu veriler kullanılarak farklı derin öğrenme modelleri eğitilmiştir. Veri seti ve modellerin öneminden bahsedilmiştir. Yapılan çalışma gelecekte yapılabilecek ek çalışmalarla sonuçların daha da iyileştirilebileceğini göstermektedir.

Anahtar Sözcükler: Stereo Görme, Semantik Segmentasyon, Derin Öğrenme, Canlı Hayvan Ağırlığı Tahmini, Görüntü İşleme.

ABSTRACT

A RESEARCH ON ESTIMATION OF LIVE ANIMAL WEIGHT USING SEMANTIC SEGMENTATION AND STEREO VISION TECHNIQUES

Derviş Kaan Akdik
Master Thesis
Department of Computer Engineering
Computer Engineering Programme
Advisor: Asst. Prof. Erdal Güvenoğlu
Maltepe University Graduate School, 2021

Animal husbandry has been one of the indispensable sources of livelihood of human beings since the beginning of human history. Today, depending on the increasing human population, the livestock industry is developing rapidly. Monitoring the animals raised and following their development processes are important in terms of efficiency. In this context, it is necessary to determine the weight of the animals raised in cattle breeding farms and to follow them regularly. In cattle breeding, it is a difficult task to regularly weigh animals and record their weights. For this reason, this process often cannot be performed regularly or can be neglected.

In this study, it was tried to estimate the weight of cattle by using stereo vision and semantic segmentation methods together. Animal photos were taken from different angles with a stereo setup consisting of two identical cameras, and the distances of the animals to the camera plane were calculated with the stereo calculation. The areas covered by the animals in the images were determined by segmentation methods and different deep learning models were trained using these data. The importance of dataset and models is mentioned. The study shows that the results can be improved further with additional studies that can be done in the future.

Keywords: Stereo Vision, Semantic Segmentation, Deep Learning, Live Animal Weight Estimation, Image Processing

İÇİNDEKİLER

JÜRİ VE ENSTİTÜ ONAYI ii
ETİK İLKE VE KURALLARA UYUM BEYANI iii
TEŞEKKÜR iv
ÖZ…….. v
ABSTRACT vi
İÇİNDEKİLER vii
ŞEKİLLER LİSTESİ ix
TABLOLAR LİSTESİ xii
KISALTMALAR LİSTESİ xiii
ÖZGEÇMİŞ xiv
BÖLÜM 1. GİRİŞ 1
BÖLÜM 2. LİTERATÜR ARAŞTIRMASI 4
BÖLÜM 3. MATERYAL VE YÖNTEM 13
3.1. Video Kameralar ve Kalibrasyon 13
3.1.1. Kamera Kalibrasyonu 17
3.2. Stereo Görme 20
3.2.1. Stereo Kalibrasyon 22
3.2.2. Stereo Düzeltme 24
3.3. Yapay Sinir Ağları 25
3.3.1. Aktivasyon Fonksiyonları 28
3.3.2. İleri Beslemeli Ağlar 30
3.3.3. Geri Beslemeli Ağlar 31
3.4. Konvolüsyonel Ağlar 31
3.4.1. Konvolüsyonel Ağı Oluşturan Katmanlar 33
3.4.2. Konvolüsyonel Ağlarda Özellik Çıkarımı 38
3.4.3. Yaygın Konvolüsyonel Ağ Mimarileri 39
3.5. Görüntü Bölümleme ve Semantik Segmentasyon 46
3.5.1. Semantik Segmentasyon 47
3.5.2. Örnek Bölümleme 47
3.5.4. Öne Çıkan Bazı Segmentasyon Yöntemleri 48

BÖLÜM 4. GERÇEKLEŞTİRİLEN ÇALIŞMA 81
4.1. Veri Hazırlama 87
4.2. Veri Normalizasyonu 98
4.3. Yapay Sinir Ağlarının Tasarımı ve Eğitimi 99
BÖLÜM 5. SONUÇ 113
KAYNAKÇA 115

ŞEKİLLER LİSTESİ

Şekil 3.1 Kameraların temel çalışma prensibi [28] 14
Şekil 3.2 İğne Deliği Kamera Modeli [30] 14
Şekil 3.3 İğne Deliği İle Görüntü Düzleminin Yer Değişmiş Hali [30] 15
Şekil 3.4 Radyal Bozulma Etkisi [29] 17
Şekil 3.5 Radyal Bozulma [30] 18
Şekil 3.6 Teğetsel Bozulma [30] 19
Şekil 3.7 Kamera İçsel Matrisi Ve Bozulma Katsayıları 20
Şekil 3.8 Stereo Düzenek geometrisi[30] 20
Şekil 3.9 Mesafe Ve Disparity İlişkisi[30] 22
Şekil 3.10 İki Kamera Arasındaki Geometrik İlişki[30] 22
Şekil 3.11 Stereo Kamera Düzeneği İle Çekilmiş Senkronize Görüntü 23
Şekil 3.12 Eşitsizlik Haritası (Sol) Ve Derinlik Haritası (Sağ)[36] 23
Şekil 3.13 Orijinal Görüntü(a) Hizalanmış Görüntü(b)[34] 24
Şekil 3.14 Stereo Düzeltme işlemi [30] 25
Şekil 3.15 Biyolojik Bir Sinir Hücresinin Yapısı 26
Şekil 3.16 Bir Nöronun Matematiksel Modellemesi [40] 26
Şekil 3.17 Çok Katmanlı Yapay Sinir Ağı 28
Şekil 3.18 Aktivasyon Fonksiyonları 29
Şekil 3.19 Ortalama Kare Hatası İçin Dereceli Alçalma[38] 31
Şekil 3.20 Konvolüsyonel Ağların Temel Çalışma Şekli 32
Şekil 3.21 Giriş Katmanı İçin Nöron Hacmi ve Nöron Yapısı[42] 34
Şekil 3.22 Bir Konvolüsyonel Katmanda Filtreleme[42] 36
Şekil 3.23 Havuzlama Katmanının Çalışma şekli[42] 37
Şekil 3.24 Özellik Haritası Ve Çıkış Sınıflandırması[42] 39
Şekil 3.25 Lenet-5 Mimarisi[41] 40
Şekil 3.26 Alexnet Mimarisi[27] 40
Şekil 3.27 Zf-Net Mimarisi[43] 41
Şekil 3.28 Googlenet Mimarisi[44] 42
Şekil 3.29 Vgg-16 Katman Yapısı[46] 43
Şekil 3.30 Resnet Mimarisi[47] 45

Şekil 3.31 Görüntü Sınıflandırma, Nesne Algılama, Görüntü Bölümleme 46
Şekil 3.32 Semantik Segmentasyon 47
Şekil 3.33 Örnek Bölümleme 47
Şekil 3.34 Tamamen Evrişimli Ağ Mimarisi[5] 49
Şekil 3.35 FCN Atlama Mimarisi[5] 50
Şekil 3.36 Bir Sınıflandırma Ağının Konvolüsyonel Ağa Dönüşümü[5] 51
Şekil 3.37 Farklı Adım Sayılarına Sahip Fcn Mimarilerinin Çıktıları[5] 54
Şekil 3.38 Mask R-CNN Yapısı[63] 56
Şekil 3.39 Rolalign İşlemi[63] 57
Şekil 3.40 Resnet Ve Fpn Temelli Faster R-CNN Modellerinin Genişletilmesi[63] 58
Şekil 3.41 Üstte Fcıs+++ Modelinin Altta İse Mask R-Cnn Modelinin Sonuçları[63] .. 59
Şekil 3.42 U-Net Mimarisi[2] 60
Şekil 3.43 Cıresan Ve Ark. Tarafından Önerilen Kayan Pencere Yaklaşımı[74] 61
Şekil 3.44 Hücre Görüntülerinin U-Net Mimarisi İle Segmentasyonu[2] 62
Şekil 3.45 Hole Algoritması Veya Atrous Algoritmasının Çalışma Mantığı[75] 64
Şekil 3.46 Standart Evrişim ve Atrous Evrişim Sonuçları[76] 66
Şekil 3.47 Koşullu Rastgele Alanlar Yaklaşımı ile Segmentasyon Sonuçları[75] 67
Şekil 3.48 Deeplab Modeli Genel Yapısı[75] 67
Şekil 3.49 İlk Deeplab Modelinin Pascal Voc-2012 Üzerindeki Sonuçları 70
Şekil 3.50 ASPP Modül Yapısı[76] 72
Şekil 3.51 Deeplabv3 Modelinde Atrous Evrişim Blokları [77] 74
Şekil 3.52 Uzamsal Piramit Havuzlama Ve Kodlayıcı-Kod Çözücü Yapısı[78] 76
Şekil 3.53 Atrous Derinlemesine Evrişim Gösterimi[78] 76
Şekil 3.54 Deeplabv3+ Mimarisinin Genel Yapısı[78] 78
Şekil 3.55 Deeplabv3+ İçin Modifiye Edilmiş Xception Mimarisi[78] 79
Şekil 3.56 Deeplabv3+ Mimarisının Doğrulama Seti Üzerindeki Sonuçları[78] 80
Şekil 4.1 Temel İşlem Adımları 82
Şekil 4.2 Satranç Tahtası İle Kamera Kalibrasyonu[30] 83
Şekil 4.3 Sol Kamera Matrisi 83
Şekil 4.4 Sağ Kamera Matrisi 84
Şekil 4.5 Kalibrasyonsuz Ve Kalibrasyonlu Görüntüler [29] 84
Şekil 4.6 Görüntü Elde Etmek İçin Hazırlanan Stereo Düzenek 85

Şekil 4.7 Stereo Kalibrasyon Matrisleri 86
Şekil 4.8 Rektifikasyonsuz ve Rektifikasyonlu Stereo Fotoğraflar 88
Şekil 4.9 Yandan Çekilmiş Stereo Görüntülerde Segmentasyon Sonuçları 89
Şekil 4.10 Arkadan Çelikmiş Stereo Görüntülerde Segmentasyon Sonuçları 90
Şekil 4.11 Disparity Değeri ve Hayvanın Görüntüdeki Piksel Sayıları 91
Şekil 4.12 Yandan Çekim Verileri 93
Şekil 4.13 Arkadan Çekim Verileri 94
Şekil 4.14 Yan Değer Ve Ağırlık İlişkisi 97
Şekil 4.15 Arka Değer Ve Ağırlık İlişkisi 98
Şekil 4.16 Birinci Yapay Sinir Ağının Kayıp Grafiği 100
Şekil 4.17 Birinci Ağın Test Seti Üzerinde Yaptığı Tahminler 102
Şekil 4.18 Birinci Ağın Eğitim Seti Üzerinde Yaptığı Tahminler 102
Şekil 4.19 Birinci Ağın Test Seti Üzerinde Hata Dağılımı Grafiği 103
Şekil 4.20 Birinci Ağın Eğitim Seti Üzerinde Hata Dağılımı Grafiği 103
Şekil 4.21 İkinci Yapay Sinir Ağımızın Kayıp Grafiği 104
Şekil 4.22 İkinci Ağın Test Seti Üzerinde Yaptığı Tahminler 105
Şekil 4.23 İkinci Ağın Eğitim Seti Üzerinde Yaptığı Tahminler 106
Şekil 4.24 İkinci Ağın Test Seti Üzerinde Hata Dağılımı Grafiği 106
Şekil 4.25 İkinci Ağın Eğitim Seti Üzerinde Hata Dağılımı Grafiği 107
Şekil 4.26 Üçüncü Yapay Sinir Ağımızın Kayıp Grafiği 108
Şekil 4.27 Üçüncü Ağın Test Seti Üzerinde Yaptığı Tahminler 110
Şekil 4.28 Üçüncü Ağın Eğitim Seti Üzerinde Yaptığı Tahminler 110
Şekil 4.29 Üçüncü Ağın Test Seti Üzerinde Hata Dağılımı Grafiği 111
Şekil 4.30 Üçüncü Ağın Eğitim Seti Üzerinde Hata Dağılımı Grafiği 111

TABLOLAR LİSTESİ

Tablo 3.1 FCN Mimarisinde Kullanılan Temel Mimarilerin Başarım Karşılaştırması .. 53
Tablo 3.2 Farklı Adımlı FCN Mimarilerinin Performans Ve Doğruluk Karşılaştırması 55
Tablo 3.3 Deeplab Varyasyonlarının Pascal Voc 2012 Üzerinde Karşılaştırılması 69
Tablo 3.4 Deeplab Modelinin Farklı Parametre Değerleriyle Performans Sonuçları..... 70
Tablo 4.1 Stereo Görüntüleri Kullanarak Oluşturduğumuz Veriseti 92
Tablo 4.2 Güncellenmiş Veri Seti..... 96
Tablo 4.3 Örnek Normalizasyon Sonuçları 99
Tablo 4.4 Birinci Yapay Sinir Ağımızın Mimari Özeti 100
Tablo 4.5 Birinci Ağın Tahmin Değerleri 101
Tablo 4.6 İkinci Ağın Tahmin Değerleri 105
Tablo 4.7 Üçüncü Yapay Sinir Ağımızın Mimari Özeti..... 107
Tablo 4.8 Üçüncü Ağın Tahmin Değerleri 109

KISALTMALAR

GPU : Graphics processing Unit, Grafik işlemci birimi

MLHMP : Multi-level hierarchic Markov processes, Çok seviyeli hiyerarşik Markov süreçleri

CCTV : Closed-circuit television, kapalı devre televizyon

YSA : Yapay sinir ağları

MLR : Multi Linear Regression, çoklu doğrusal regresyon

LASSO : Least absolute shrinkage and selection operator, En Az Mutlak Büzülme ve Seçim Operatörü

PLS : Partial Least Squares, Kısmi En Küçük Kareler

RMSE : Root Mean Square Error, Kök Ortalama Kare Hata

ReLU : Rectified Linear Unit, Rektifiyeli lineer birim

CNN : Convolutional Neural Network, konvolüsyonel sinir ağı

FCN : Fully Convolutional Network, tamamen evrişimli ağ

ILSVRC : Large Scale Visual Recognition Challenge, büyük ölçekli görsel tanımlama mücadelesi

SVM : Support Vector Machine, Destek Vektör Makinesi

LSTM : Long short-term memory, Uzun-Kısa Süreli Bellek

MRF : Marcov Random Field, Markov Rastgele Alanı

CRF : Conditional Rastgele Fields, Koşullu rastgele alanlar

DAG : Directed acyclic graph, Yönlendirilmiş döngüsel olmayan graf

R-CNN : Regional Convolutional Neural Network, Bölgesel konvolüsyonel sinir ağı

LargeFOV : Large Field of View, Geniş görüş alanı

MSc : Multi-Scale, Çok ölçekli

ASPP : Atrous spatial pyramid pooling, Atrous uzamsal piramit havuzlama

MSRA : Microsoft Research of Asia, Mikrosoft Asya Araştırma Ekibi

RoI : Region of Interest, İlgi bölgesi

ÖZGEÇMİŞ

Derviş Kaan Akdik

Bilgisayar Mühendisliği Anabilim Dalı

Eğitim

Derece	*Yıl*	*Üniversite, Enstitü, Anabilim/Anasanat Dalı*
Ls.	2006	Marmara Üniversitesi, Eğtim Fakültesi Bilgisayar ve Öğretim Teknolojileri
Lise	2000	Tuzla Teknik Lisesi, Bilgisayar, Tuzla / İstanbul

İş/İstihdam

Yıl	*Görev*
2006	Bilişim Teknolojileri Öğretmeni, Milli Eğitim Bakanlığı

BÖLÜM 1. GİRİŞ

İnsan hayatında hayvanlar hem günlük hayatta hem de ekonomik yönden önemli bir konuma sahiptirler. Birçok insan evinde hayvanlara yer vermekte, onları evlerinin bir üyesi olarak görmekte, sağlıkları ve bakımları ile ilgilenmektedirler. Ticari açıdan bakıldığında ise insanlık, ilk var olduğundan beri hayvanlardan menfaat sağlama çabası içine girmiştir. Günümüzde tüm dünyada ticari hayvan çiftlikleri kurulmuş, hayvan yetiştiriciliği çok büyük bir sektör haline gelmiştir.

Hayvancılık, kırsal kesimde tarımla uğraşan insanların yan uğraşı olmasının yanı sıra önemli bir sanayi sektörü haline gelmiştir. Kooperatifleşme, üretici birlikleri, kayıtlı yetiştiricilik, suni tohumlama uygulamaları ve hayvancılık desteklemeleri gibi uygulamalar sayesinde ülke ekonomisinde hayvancılık sektörünün yeri daha da önem kazanmaya başlamıştır [1].

Bu bağlamda insanlar, hayvanlarla ilgilenmek, onların sağlık durumlarını göz önünde bulundurmak ve gerekli tedbirleri almak zorundadır. Bir hayvanın sağlığı ile ilgili birçok parametre bulunabilir. Bu parametrelerin ölçümleri bazen kolay, bazen ise oldukça zahmetlidir.

Özellikle yüksek hacim ve ağırlığa sahip hayvanlarla çalışmak bazen çok zahmetli olabilir. Yüksek sayıda hayvanın olduğu çiftliklerde bu işlem çok fazla zaman ve emek gerektirebilmektedir. Büyükbaş bir hayvanın cinsine göre vücudundaki bazı ölçüler bir araya getirildiğinde bunlar, o hayvanın ağırlığı ile ilgili bir gösterge olabilir. Daha önce bu konuda birçok araştırma yapılmış, bu ölçüleri elde etmek için birçok farklı yöntem denenmiştir. Hayvanların geçiş yollarına stüdyo kurulup, termal kamera, lazer, kızılötesi kamera ve benzeri cihazlarla elde edilen bu ölçüler bilgisayar ortamında değerlendirilerek hayvanın ağırlığı tahmin edilmeye çalışılmıştır. Bu yöntemler hem maliyetli hem de nispeten yavaş ve zahmetli olabilir. Bu tür işlemlerde zaman ve emek son derece kıymetlidir. Aynı zamanda hayvanın doğal ortamından ayrılması hayvanda strese sebep olmakta ve sağlığını olumsuz etkilemektedir.

Bu tez çalışmasındaki amaç, son teknoloji bilgisayar görüsü tekniklerini kullanarak, büyükbaş bir hayvanı doğal ortamından mümkün olduğu kadar ayırmadan, hızlı ve etkili bir şekilde vücut ölçülerini belirleyip bunlar üzerinden ağırlığını tahmin etmeye çalışmaktır. Bu amaçla sınanmak istenen denenceler şunlardır:

- Semantik segmentasyon tekniği kullanılarak, bir canlının doğal ortamında, farklı açılardan çekilmiş görüntülerinin, bulunduğu ortamdan (arka plandan) ayırt edilmesine çalışılması, yani hayvanın görüntüde kapladığı piksellerin bulunması
- Ayırt edilen bu görüntülerde stereo görme tekniği ile hayvanın kameraya olan uzaklığının tespit edilmesi. Bu uzaklık değeri kullanılarak daha önceden oluşturulmuş eşleştirme tablolarından görüntüdeki bir pikselin gerçek hayatta kaç milimetreye denk geldiğinin bulunması.
- Bu ölçülerin daha önceden eğitilmiş bir yapay sinir ağı veya makine öğrenme algoritmasına gönderilerek hayvanın ağırlığının tahmin edilmeye çalışılması

Semantik segmentasyon teknikleri birçok amaç için olduğu gibi bir görüntüdeki canlı hayvanın sınırlarını belirlemek içinde kullanılabilir. Semantik segmentasyon başlığı altında birçok farklı model kullanılmaktadır. Bunlardan birisi olan U-Net nispeten basit ve kullanışlı olmasıyla ön plana çıkar. Dahası, bu ağ modeli oldukça hızlıdır. 512x512'lik bir görüntünün bölümlenmesi, yeni bir GPU'da bir saniyeden daha kısa sürer [2]. Bununla beraber semantik segmentasyon teknikleri hayvanın vücut ölçüleri hakkında bilgi verme konusunda yetersiz kalabilir.

Stereo görme teknikleri ise bir görüntüdeki derinliği ve görüntüdeki nesnelerin boyutları ile ilgili ölçüleri ortaya çıkarabilir. Stereo görüsü veya çiftli görselleştirme basitçe birden çok kamera kullanılarak gerçekleştirilen bir derinlik bulma işlemi olarak tanımlanabilir [3]. Stereo görme bir cismin hacminin hesaplanması için de kullanılabilir. Hacim tahmini için önceki çözümler genellikle belirli nesnelerle sınırlıdır. Geçmiş çalışmalar ağırlıklı olarak yiyecek hacmi tahmini problemine yaklaşır [4].

Semantik segmentasyon ve stereo görme teknikleri beraber kullanıldıklarında bir problemin çözümünde ne kadar etkili olabileceklerini, problemin çözümüne nasıl ve ne yönden katkı sağlayabileceklerini görmek araştırmanın önemi olarak görülebilir.

Bu tez çalışmasında kullanılan hayvan görüntülerinde sınırları belirlenmek istenen hayvan görüntüsü ile arka plan görüntüsünün renk tonlarının aşırı derecede benzer olmayacağı varsayılmıştır. Böyle bir ortamda semantik segmentasyon uygulamasının büyük ölçüde başarılı olacağı varsayılır. Semantik segmentasyon modelleri evrişimli ağlar üzerine kuruludur. Evrişimli ağlar, görüntüdeki özelliklerin hiyerarşilerini sağlayan güçlü görsel modellerdir [5]. Bir diğer varsayım olarak kamera ile farklı açılardan çekilen görüntülerde hayvanın tüm vücut görüntüsünün bulunacağı varsayılmıştır.

Büyükbaş bir hayvanın vücut ölçülerinden ağırlık tahminin yapılması hayvanın cinsine göre değişiklik gösterebilir. Türkiye'de yetiştirilen kültür ırk ve melezi sığırlar arasında Siyah Alaca olarak da tanımlanan Holştayn ırkı %70 oranında önemli bir yer tutar [6].

BÖLÜM 2. LİTERATÜR ARAŞTIRMASI

Bilgisayar görüsü kullanılarak canlı hayvan ağırlığının tahmini özellikle ticari hayvan işletmelerine zamandan ve emekten tasarruf etmek için önemlidir. Hayvan ağırlığını manuel olarak ölçmek insanlar için bazı hayvan türlerinde tehlikeli bile olabilir. Bu nedenden dolayı bu alanda geçmişten günümüze kadar yapılmış birçok çalışma bulunmaktadır. Bu çalışmaların bazıları aşağıda yer almaktadır.

Brandl ve ark. [7] bir görüntü analiz sistemi kullanarak domuzların vücut ölçülerini belirleyip bu ölçülerden hayvanların canlı ağırlıklarının tahmini üzerine çalışmışlardır. Çalışmada, 416 melezlenmiş domuz kullanılmış ve her bir domuz 25 ila yaklaşık 100 kg büyüme periyodu boyunca beş kez (yukarıdan) videoya kaydedilmiştir. Yapılan çalışmada tek bir hayvanın vücut ölçüleri ile ağırlıkları arasındaki ilişkiyi ifade etmek için spline(bağ interpolasyonu) fonksiyonları kullanılmıştır. Daha sonra domuzların karışık bir şekilde ağırlıklarını tahmin etmek için karma bir model kullanılmıştır. Spline fonksiyonlarının sonucunda standart sapma cinsinden ifade edilen hata %8-9, karma modelde %5-6 olarak tahmin edilmiştir. Sonuç olarak kesinlik oldukça düşük olsa da domuzların ağırlık tespiti için görüntü analiz teknikleri ümit verici görülmüştür. Bu tekniklerin nispeten düşük fiyatlı standart ekipmanla kullanılabileceği görülmüştür. Bununla birlikte, muhtemelen üç boyutlu bir yaklaşım kullanan otomatik görüntü analizi teknikleri ve sürü / cinse özgü alan-ağırlık ilişkilerinin otomatik kalibrasyonu için yeni teknikler konusunda daha fazla araştırmaya ihtiyaç olduğu sonucuna varmışlardır.

Schofield ve ark. [8] domuzların ağırlıklarıyla ilişkili olan vücut alanlarını kaydetmek için tasarlanmış bir otomatik görüntü toplama ve analiz sistemini kullanarak domuzların büyümelerini takip etmişlerdir. Baz aldıkları beden alanları ile ağırlık tahmini yapmışlardır. Üç grup erkek domuz 47 gün boyunca ağırlıkları 47 kilodan 90 kiloya çıkana kadar izlenmiştir. Grubun ortalama ağırlık hatasının %5 dahilinde olduğu tahmin edilmiştir. 60-90 kg ağırlık aralığında, Landrace ve Large White domuzları için hatanın %2'nin altına ve Meishan domuzları için %3'ün altına düştüğü belirlenmiştir. Her zaman mükemmele yakın görüntülerin toplanmasını sağlamaya çalışmak için donanımın karmaşıklığını ve maliyetini arttırmaktansa, ortalama kalitede birçok görüntüyü

yakalamak ve bunların bir kısmını yazılım tekniklerini kullanarak işlemenin daha pratik olduğu sonucuna varmışlardır.

Wet ve ark. [9] yaptıkları çalışmada bilgisayar destekli görüntü analizi ile etlik piliçlerin günlük vücut ağırlığı değişikliklerini araştırmışlardır. Çalışmada, ticari koşullarda yetiştirilen 50 adet etlik piliç kullanmışlardır. 50 tavuktan 10 tanesi rastgele seçilmiş ve 42 günlük büyüme periyodu sırasında 18 kez video kaydına alınmıştır. Görüntüler yüzey ve çevre piksellerinin sayısı, vücut boyutu ile canlı ağırlık arasında bir ilişki türetmek için kullanışmıştır. Görüntüdeki yüzey verileri üzerinden standart sapma olarak ifade edilen ağırlık tahminindeki göreceli hata %10 iken, görüntünün çevresi üzerinden yapılan hesaplamalarda hata %15 olarak bulunmuştur.

Kristensen ve ark. [10] kesim için seçilecek domuzların belirlenmesinde kullanılmak üzere canlı ağırlıkların görüntü analizi yoluyla otomatik olarak değerlendirilmesi için bir yöntem geliştirmişlerdir. Bu çalışmada kullanmış oldukları optimizasyon aracı hiyerarşik bir Markov sürecidir. Durum alanı, çevrimiçi canlı ağırlık tahmininden gelen gözlemlerin karar destek sistemi için girdi olarak hizmet edebileceği şekilde tanımlanmıştır. Bu şekilde, gelecekteki büyümenin altında yatan tahminlerin kesinliği arttırılmıştır. Amaç, çiftçiyi kümes başına pazara hazır domuz sayısı hakkında bilgilendirebilecek bir karar desteği oluşturmaktır. Modeli uygulamak için MLHMP yazılım sistemi kullanılmıştır.

Wu ve ark. [11] Canlı domuzların 3 boyutlu şekillerini oluşturmak için altı yüksek çözünürlüklü kamera (3032 × 2028 piksel) ve üç flaş ünitesi içeren bir stereo görüntüleme sistemi geliştirmişlerdir. Kameralar, her bir domuzun yandan, üstten ve arkadan görüntülerini yakalayan üç stereo bölmede düzenlenmiştir. Görüntü çözünürlüğü, domuzun yüzeyindeki piksel başına 0,4 milimetre olarak bulunmuştur. Sistem, 14 hafta boyunca, domuzların ağırlıkları 30 ila 80 kg arasında gelişim gösterecek şekilde 32 domuzun görüntülerini yakalamak için kullanılmıştır. Stereo görüntüleme sistemi, doğal deri dokusu üzerinde başarılı bir şekilde çalışmış ve yansıtılması için herhangi bir ek, yapay doku gerektirmemiştir. Domuzların 3 boyutlu modelleri, ± 0.6 mm'lik bir karekök ortalama sapması ile niteliksel olarak iyi bir şekilde oluşturulmuştur.

Stajnko ve ark. [12] Simental ırkı boğaların termografi ve termal görüntü analizi ile vücut boyutlarının ölçülerek canlı ağırlıklarının tahmin edilmesi üzerine çalışmışlardır. Çalışmada kullanılan boğa Fgrubu, ortalama olarak 139 günlük ve 181 kg ağırlığa sahip iken kesime kadar (449 gün, 594 kg) her 60 günde bir yedi kez ölçülmüştür. Çalışmada her boğa manuel olarak tartılmış ve vücut ölçüleri alınmıştır. Devamında termal kamera ile görüntülenmişlerdir. Geliştirilen sistemin, görüntü analiz algoritmasıyla birlikte, yedi yaş sınıfının tamamında cidago yüksekliği ve kalça yüksekliğini tahmin etmek için başarılı bir araç olduğunu kanıtlanmıştır.

Wang ve ark. [13] yaptıkları çalışmada, sabit görüntüleme için domuzu belirli bir alana sınırlamak zorunda kalmadan, domuz canlı ağırlığı tahmini için görüntü tabanlı bir gözden geçirme sistemi geliştirmişlerdir. Görüntü işleme için yakalanan görüntüleri otomatik olarak taramak ve seçmek için bir protokol geliştirmişlerdir. Yapay sinir ağı tekniği, canlı ağırlık yaklaşımının doğruluğunu artırmak amacıyla, bu çalışmada, gezinme görüntülerinden çıkarılan çok sayıda fiziksel özelliği domuz canlı ağırlığıyla ilişkilendirmek için kullanılmıştır. Sonuçlar, gözden geçirmeli tartım sisteminin ortalama bağıl hatasının yaklaşık %3 olduğunu göstermiştir. Sonuç olarak bu sistemin, hayvancılıkla uğraşanların makine vizyonuna dayalı bir tartım sistemi kullanarak domuzların canlı ağırlığının elde etmesini daha da kolaylaştırdığı belirtilmiştir.

Negretti ve ark. [14] Akdeniz bufalolarında canlı ağırlık ve vücut kondisyon skorunun görsel görüntü analizi ile belirlenmesi üzerine bir çalışma yapmışlardır. Çalışma sonuçları, bufalo türleri için bilgisayarlı görüntü analizinin canlı ağırlığın dolaylı olarak belirlenmesi için etkili bir ölçüm sistemi olduğunu doğrulamıştır. Ayrıca, bu tekniğin, besi derecesinin objektif olarak belirlenmesi için teknolojik olarak yenilikçi olduğu kanıtlanmıştır.

Mollah ve ark. [15] Etlik piliçlerin canlı ağırlığını tahmin etmek için bilgisayar destekli dijital görüntü analizi üzerine bir çalışma yapmışlardır. Standart yetiştirme koşullarında yetiştirilen 100 civcivden 7–42 günlük büyüme dönemi boyunca, rastgele seçilen 20 piliçten 1200 dijital görüntü yakalamışlardır. Yakalanan görüntüler, vücut yüzey alanını belirlemek için tarama görüntü analiz yazılımı IDRISI 32(32-bit object-oriented system, 32-bit nesne yönelimli sistem) ile analiz edilmiş ve ağırlıklarını vücut yüzey alanı piksellerinden tahmin etmek için doğrusal bir denklem geliştirilmiştir. Yüzey

alanı piksellerinden kalan kalıntıların hata yüzdesi olarak ifade edilen görüntü analizi ile etlik piliç ağırlık tahminindeki göreceli hata 0,04 ile 16,47 arasında bulunmuştur. Öte yandan, tahmini vücut ağırlıklarının, 35 günlük yaşa kadar elle ölçülen vücut ağırlıklarından önemli ölçüde farklı olmadığı görülmüştür. Bu nedenle canlı piliç tartmak için pratik bir görüntüleme sistemi geliştirilmesinin mümkün olduğu sonucuna varmışlardır.

Taşdemir ve ark. [16] yaptıkları çalışmada Holstein ineklerinin vücut ölçülerini dijital görüntü analizi kullanılarak belirlenmiş ve bunlar üzerinden her bir ineğin canlı ağırlığını tahmin etmeye çalışmışlardır. Bu amaçla, bir süt sığırcılığı çiftliğinde görüntü yakalama düzenlemesi kurulmuştur. Cidago yüksekliği, kalça yüksekliği, vücut uzunluğu, kalça genişliği ve ineklerin canlı ağırlıkları dahil olmak üzere vücut ölçüleri ilk olarak doğrudan ölçümle manuel olarak belirlenmiştir. Daha sonra ineklerin dijital fotoğrafları eşzamanlı olarak farklı yönlerden çekilmiş ve bu fotoğraflar her bir ineğin cidago yüksekliği, kalça yüksekliği, vücut uzunluğu ve kalça genişliğini hesaplamak için görüntü analiz yazılımı tarafından analiz edilmiştir. Görüntü analiz yazılımı ile elde edilen vücut ölçüleri manuel ölçümlerle karşılaştırıldıktan sonra doğruluk cidago yüksekliği için %97,72, kalça yüksekliği için %98,00, vücut uzunluğu için %97,89 ve kalça genişliği için %95,25 olarak belirlenmiştir. Daha sonra vücut ölçüleri kullanılarak canlı ağırlık tahmini, regresyon denklemleri yardımıyla gerçekleştirilmiş ve tartımla elde edilen tahmini ve gerçek (manuel) canlı ağırlıklar değerleri arasındaki korelasyon katsayısı 0,9787 olarak hesaplanmıştır. Sonuç olarak görüntü analiz yazılımı yönteminin Holstein'ın canlı ağırlık tahmini için uygun olduğu sonucuna varmışlardır.

Menesatti ve ark. [17] bir sahnedeki benzer noktaların üç boyutlu konumunu elde etmek için düşük maliyetli, çift web kamerasından oluşan yüksek çözünürlüklü bir sistem geliştirmişlerdir. Devamında Matlab ortamında iki farklı yazılım geliştirmişlerdir. Birinci yazılım, iki web kamerası tarafından fotoğraf karelerini etmek için, ikincisi görüntüleri işlemek ve benzer noktaların konumunu ve böylece noktalar arasındaki mesafeleri elde etmek için kullanılmıştır. Noktalar arasındaki mesafeler, Kısmi En Küçük Kareler regresyonu kullanılarak düzeltilmiştir. Sistemin verimliliği bir kalibrasyon kurulum deneyinin ardından 27 canlı Alpagota koyunu üzerinde değerlendirilmiştir. Kalibrasyon aşamasında, 10 metreden daha düşük bir mesafede, ortalama boyut hataları her zaman

%2'den daha düşük bulunmuştur. Alpagota koyunları üzerindeki ölçümler, cidago yüksekliği ve göğüs derinliği için daha düşük (yaklaşık %3,5) ve vücut uzunluğu için daha yüksek (yaklaşık %5,0) hata göstermiştir. Kalibrasyon aşamasındakilere göre daha yüksek olan bu hatalar, kısmen hayvanlar üzerindeki manuel ölçümlerin hassasiyetinden kaynaklanmıştır. Geliştirilen bir fonksiyon ile koyun ağırlığı tahmini, manuel olarak ölçülen verilerle ve stereo görme ile elde edilen verilerle sırasıyla 3,6 kg ve 4,4 kg hata ile gerçekleşmiştir. Bu kadar ucuz, kullanıma hazır ve kolay taşınabilir olan bu sistemin, ormancılık, tarım ve çevre gibi diğer biyosistem alanlarında uygulama alanı bulabileceği sonucuna varmışlardır.

Mortensen ve ark. [18] yaptıkları çalışmada, ticari bir üretim ortamında etlik piliçler için tam otomatik 3 boyutlu kamera tabanlı bir tartım sistemi geliştirilmişlerdir. Çalışmada, doğrudan bir derinlik görüntüsü veren düşük maliyetli bir 3D kamera (Kinect) kullanılmıştır. Kameranın, kendi kızılötesi ışık kaynağını içerdiğinden dolayı, kümesin değişen ışık koşullarına karşı dayanıklı olduğu görülmüştür. Çalışmada yeni geliştirilmiş bir görüntü işleme algoritması önerilmiştir. Algoritma, ilk olarak görüntüyü mesafeye dayalı bir watershed algoritmasıyla bölümlere ayırmış, ardından on iki farklı ağırlık tanımlayıcısı çıkarmış ve son olarak bir Bayes Yapay Sinir Ağı kullanarak bireysel piliç ağırlıklarını tahmin etmiştir. Ağırlık tahmini için bundan başka dört model de değerlendirilmiştir. Sistem, üreme döneminin son 20 günü boyunca 48.000 piliç ile ticari bir piliç kümesinde test edilmiştir. Referans ağırlıklarını tahmin etmek için geleneksel bir platform kantarı kullanılmıştır. Tahmin edilen ağırlıklar ile referans ağırlıklar arasında ortalama olarak %7,8'lik bir bağıl hata, yaklaşık 13.000 manuel açıklamalı görüntüde 83 piliç ile ayrı bir test setinde elde edilmiştir. Görüntüdeki piliç yoğunluğu arttıkça yetiştirme döneminin sonunda hataların genellikle daha büyük olduğu görülmüştür. Mutlak hata değerlerinin periyodun ilk yarısında 20–100 g, son yarısında 50–250 g aralığında olduğu görülmüştür. Sistemin, faaliyet analizi ve sağlık uyarıları gibi ticari piliç ağılında çok çeşitli ilave kamera tabanlı ölçümler için geliştirilerek kullanılabileceği sonucuna varmışlardır.

Shi ve ark. [19] yaptıkları çalışmada bina içi çiftlik koşullarında, 14 ila 25 haftalık 10 domuz kullanarak vücut boyutunu ve canlı vücut ağırlığını analiz ve tahmin etmek için binoküler stereo görüş sistemini LabVIEW geliştirme platformuyla birleştirmişlerdir.

Deney tesisi, veri toplama sistemi, görüntü toplama sistemi ve görüntü analiz sisteminden oluşturulmuştur. Domuzun kimlik tanıma ve görüntü alımının otomatik olarak uygulanmasını sağlamak için dürbün kamera ve özel olarak tasarlanmış sınırlama çiti içeren veri toplama sistemi kullanılmıştır. Görüntüler vücut ağırlığı ile vücut büyüklüğü arasındaki ilişkiyi analiz etmek için diğer iki sistem tarafından elde edilmiş, kaydedilmiş ve işlenmiştir. Binoküler kameralarla elde edilen görüntüler derinlik eşiğine göre bölümlere ayrılmıştır. Sistem, vücut uzunluğu ve omuz yüksekliği için R^2'yi 0,91 ile 0,98 arasında tahmin etmiştir. Zemin ölçeğinden elde edilen sırt bölgesi ve vücut ağırlığı verileri kullanılarak lineer model üzerinde en küçük kareler yöntemi ile regresyon sonucuna göre fonksiyonun parametreleri hesaplanmıştır. Sonuçlar, canlı vücut ağırlığının, yeniden yapılandırılmış görüntüden hesaplanan piksel değerlerinin toplamı ile yakından ilişkili olduğunu ortaya koymuştur. R^2 değeri belirleme katsayısı 0.9931 olarak bulunmuştur. Bu durumun, bu prosedürün canlı ağırlığı tahmin etmede iyi bir performans gösterdiği anlamına geldiği görülmüştür. Çalışma sonunda tüm otomatik sistemlerin gerçek çiftlik ortamında uygulanma potansiyeline sonucuna varılmıştır.

Hansen ve ark. [20] tek bir 3D video yakalama cihazı kullanarak süt ineği vücut durumunun, hareketliliğinin ve ağırlığının otomatik olarak izlenmesi üzerine bir çalışma yapmışlardır. Bu çalışmada, aynı anda tek bir cihaz kullanarak ilgili hayvan durumu verilerinin (vücut durumu, topallık ve ağırlık) farklı biçimlerini elde edebilen 3D görüntüleme teknolojisi önerilmiştir. Sürekli işletimde çiftlik koşullarında elde edilen sonuçların, sürünün manuel olarak puanlanmasına benzer veya daha iyi olduğu gösterilmiştir. Ayrıca, puanlama kullanmanın doğasında olan sınırlamalar da göz önünde bulundurulmuştur ve ardışık gözlemlere göre göreceli değişime duyarlılığın, soyut ve keyfi puanlama sistemleri olarak kabul edilebilecek sistemlerin kullanımından daha fazla fayda sağladığı savunulmuştur.

Nir ve ark. [21] düve boyunu ve vücut kütlesini otomatik olarak tahmin etmek için 3D Bilgisayarlı görme sistemi üzerine bir çalışma gerçekleştirmişlerdir. Geliştirdikleri yöntem, tek bir düşük maliyetli Microsoft Kinect V2 3D sensörü, bilgisayarla görme, makine öğrenimi ve özellik çıkarma aşamasının bir parçası olarak kuantil regresyonlu elips uydurma kullanarak nesne tanımayı içerir. Kamera, mandıra çiftliğinde, beslenme bölgesi ile yatma alanı arasındaki serbest yürüme yolunun üzerindeki tavana kurulmuştur.

107 İsrail Holstein düvesinin video verileri kaydedilmiş ve insan tarafından gözlemlenen vücut kütlesi, kalça yüksekliği ve cidar yüksekliğine göre doğrulanmıştır. Test edilen sistem, son teknoloji modellere göre tahminlerin normalleştirilmiş Kök Ortalama Kare Hatasını, omuz yüksekliği, kalça yüksekliği ve vücut kütlesi için sırasıyla %70,4, %69,8 ve %42,6 oranında iyileştirilmiştir. Modeller ayrıca farklı bir mandıra çiftliğinde doğrulanmış ve benzer sonuçlar vermiştir. Metodolojinin, koyunlar, domuzlar, atlar ve bufalo gibi diğer eliptik şekilli hayvan gövdelerine uyarlanabilir ve uygulanabilir olduğu sonucuna varmışlardır.

Song ve ark. [22] 3 boyutlu görüş kullanarak süt ineklerinin otomatik vücut ağırlığı tahmini üzerine bir çalışma yapmışlardır. Bu vaka çalışmasında, üstten görünüşlü 3-D kamera ile donatılmış bir inek seçim kutusunda bir görüntü elde etme kurulumu oluşturulmuştur. Kalça yüksekliği, kalça genişliği ve sağrı uzunluğunun morfolojik özellikleri, süt ineklerinin sağrı bölgesinden alınan ham 3-D görüntülerinden otomatik olarak çıkarılmıştır. Bu özellikler süt, yaş ve paritedeki günlerle birleştirilmiş, vücut ağırlığını tahmin etmek için çoklu doğrusal regresyon modellerinde kullanılmıştır. En iyi tahmin modelini bulmak ve ara modeller oluşturmak için kapsamlı bir özellik seçim algoritması kullanılmıştır. Her model, kök ortalama kare hatası ve ortalama mutlak yüzde hatası ile, birini dışarıda bırakarak çapraz doğrulama ile doğrulanmıştır. Kalça genişliği (0,006 m ölçüm değişkenliği), sütte gün sayısı ve pariteden oluşan modelin, 41,2 kg kök ortalama kare hatası ve %5,2 ortalama mutlak yüzde hatayla en düşük hata ile en iyi model olduğu görülmüştür . Görüntü alma kurulumu, görüntü analizi ve en iyi tahmin modelini içeren bütünleşmiş sistem, vücut ağırlıklarını yarı otomatik veya manuel yöntemler kullanılarak elde edilene benzer bir performansla tahmin etmiştir. Dahası, basitleştirilmiş morfolojik özellik ölçümünün değişkenliği, vücut ağırlığı tahminindeki belirsizliğe ihmal edilebilir bir katkı göstermiştir. Süt ineği vücut ağırlığı tahmininin, daha öngörücü morfolojik özelliklerin dahil edilmesi ve tahmin modeli yapısının iyileştirilmesiyle daha iyi gerçekleştirilebileceği sonucuna varmışlardır.

Suwannakhun ve ark. [23] domuz üretiminde, canlı domuzların ağırlıklarını gerçek zamanlı olarak tespit edebilecek bir sistem üzerinde çalışmışlardır. Önerilen sistem, renk, doku, ağırlık merkezi, ana eksen uzunluğu, küçük eksen uzunluğu,

eksantriklik ve alan gibi önemli özellikler sunmuştur. Domuzun tahmini ağırlığının tanınması için, sinir ağına sahip orijinal veri tabanındaki istatistikler kullanılmıştır.

Ter-Sarkisov ve ark. [24] Tamamen Evrişimli Sinir Ağını(Fully Convolutional Network) kullanan Sığır Örneği Segmentasyonu üzerine bir çalışma yapmışlardır. Bu çalışmada, ilgi alanlarını tahmin etmeden nesneleri ayrı ayrı etiketlemeyi öğrenmek için tamamen evrişimli bir ağı genişleten yeni bir örnek segmentasyon algoritması sunulmuştur. Yeni algoritma, sığır etinin bir CCTV kamera ile alınan kayıtlarının yanı sıra MS COCO ve Pascal VOC veri kümeleri kullanılarak eğitilmiştir. Yapılan kapsamlı denemeler, yaklaşımın son teknoloji çözümlerden elde edilen verilerde %8'e kadar daha iyi performans gösterdiğini ortaya koymuştur.

Tasdemir ve ark. [25] vücut boyutları fotogrametri yöntemi ile belirlenen Holstein ineklerinin canlı ağırlıklarını yapay sinir ağı üzerinden tahmin etmek üzerine bir çalışma yapmışlardır. Fotogrametri ile ineklerin sarkma yüksekliği, kalça yüksekliği, vücut uzunluğu, kalça genişliği belirlenmiştir. Bu vücut ölçüleri kullanılarak yapay sinir ağı tahmin modeli geliştirilmiştir. Veri kümesi, ön işlemeden sonra eğitim ve test veri kümesi olarak ikiye ayrılmıştır. Farklı yapılandırılmış yapay sinir ağı modelleri oluşturularak en iyi performansa sahip yapay sinir ağı modeli belirlenmiştir. Daha sonra bu yapay sinir ağı modeli ile görüntülerden elde edilen ölçümler kullanılarak hayvanların canlı ağırlıkları tahmin edilmiştir. Ölçekten elde edilen tahmini canlı ağırlıkların karşılaştırılmasından sonra korelasyon katsayısı bulunmuştur (R = 0.995). İstatistiksel analiz, her iki grubun da anlamlı olduğunu ve yapay sinir ağlarının canlı ağırlık tahmininde güvenle kullanılabileceğini göstermiştir.

Cominotte ve ark. [26] Besi sığırlarında vücut ağırlığını ve ortalama günlük ilerlemeyi tahmin etmek için otomatik bilgisayarlı görme sistemi üzerine bir çalışma yapmışlardır. Bu çalışmada sığırlarda canlı ağırlığı ve ortalama günlük ilerlemeyi tahmin etmek için kullanılan otomatikleştirilmiş bir bilgisayarlı görme sisteminin tahmin kalitesini değerlendirmek ve Çoklu Doğrusal Regresyon (MLR), En Az Mutlak Büzülme ve Seçim Operatörü (LASSO), Kısmi En Küçük Kareler (PLS) ve Yapay Sinir Ağlar (YSA) dahil olmak üzere farklı öngörücü yaklaşımları karşılaştırmak üzerinde durulmuştur. Sütten kesme, stoklama ve besi yeri aşamalarında toplam 234 Nellore sığır görüntüsü toplanmıştır. İlk olarak, her bir hayvanın vücut hacmi, alanı, uzunluğu ve

diğerleri gibi biyometrik vücut ölçümleri Kinect® sensörü ile yakalanan üç boyutlu görüntüler kullanılarak gerçekleştirilmiş ve ilgili canlı ağırlıkları bir elektronik ölçek kullanılarak elde edilmiştir. Daha sonra, biyometrik ölçümler dört öngörücü yaklaşımda (MLR, LASSO, PLS ve YSA) açıklayıcı değişkenler olarak kullanılmıştır. Tahmin kalitesini değerlendirmek için, birini dışarıda bırakan çapraz doğrulama yöntemi kullanılmıştır. YSA'nın, Kök Ortalama Kare Hatası (RMSE) ve kare tahmin korelasyonu (r^2) açısından en iyi tahmin yaklaşımı olduğu görülmüştür. Genel olarak sonuçlar, önerilen otomatik bilgisayarlı görme sisteminin sığırlarda gerçek zamanlı canlı ağırlık ve günlük ortalama ilerlemeyi tahmin etmek için başarıyla kullanılabileceğini göstermiştir.

Yukarıda özetlenen çalışmalarda görüldüğü gibi bir hayvanın canlı ağırlığını ölçmek için genel olarak kullanılan yöntem, stereo kamera, derinlik sensörü içeren kameralar veya bunlara benzer donanımlar kullanarak hayvanın birtakım vücut ölçülerini elde etmek, devamında bu ölçüleri çeşitli makine öğrenmesi veya yapay sinir ağı modellerinde kullanarak tahminler yapmaktır. Tez çalışmamızda stereo görme ve semantik segmentasyon teknikleri birlikte kullanılarak canlı ağırlık tahmini yapmaya çalışılacaktır.

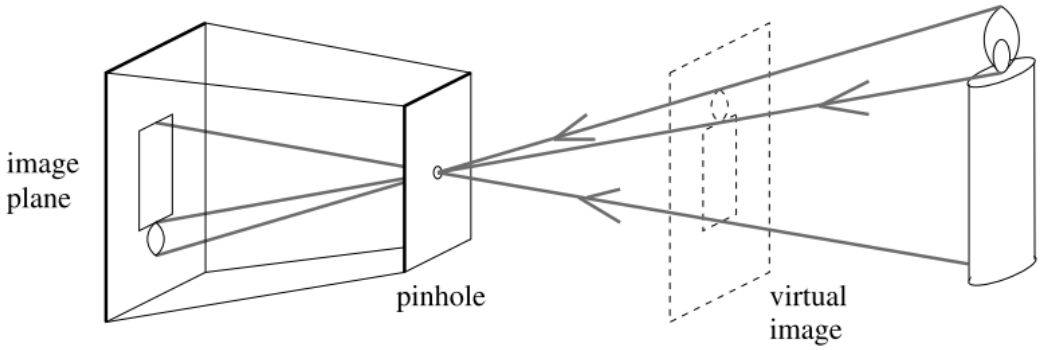
BÖLÜM 3. MATERYAL VE YÖNTEM

Bu alandaki son çalışmalar bilgisayar görüsü ve yapay zekanın birlikte etkin olarak kullanılması ile yapılmaya çalışılmaktadır. Yapay zekanın popüler hale gelme sürecinden önce bilgisayar görüsü ile birlikte daha çok makine öğrenmesi algoritmaları kullanılmıştır. 90'lı yıllarla beraber donanım kapasitelerinin ve bilgisayarların işlem gücünün artması sayesinde yapay zekâ ve yapay sinir ağları tekrar popüler hale gelmiştir. Özellikle 2012 yılında Alex Krizhevsky, Ilya Sutskever ve Geoff Hinton tarafından geliştirilen bir konvolüsyonel ağ mimarisi olan AlexNet [27] ile beraber yapay sinir ağı tabanlı olan konvolüsyonel ağlar, bilgisayar görüsü ile ilgili birçok alandaki problemler için kullanılmaya başlanmıştır. Ayrıca python, java, matlab gibi programlama dilleri ile kullanılabilen yapay zekâ ve bilgisayar görüsü kütüphaneleri sayesinde geçmişte çözümü uzun ve zahmetli olan birçok problem kısa zamanda ve kolaylıkla çözülebilir hale gelmiştir.

3.1. Video Kameralar ve Kalibrasyon

Bir kamera basitçe, etraftan yansıyan ışınları bir mercek sistemi aracılığıyla bir noktada toplayıp bu noktadan da ışığa duyarlı özel bir düzleme aktararak bu düzlemde görüntüyü oluşturan bir cihaz olarak tanımlanabilir.

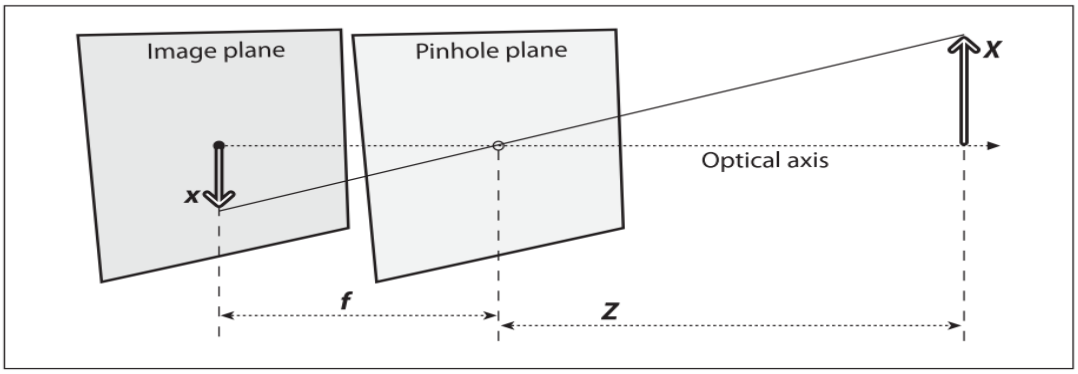
Bir kamera tarafından oluşturulan bir görüntü gerçekte üç boyutlu bir dünyanın iki boyutlu bir düzlemde temsilidir. Üç boyutlu dünyadan iki boyutlu bir görüntüye düşüş, bir boyutu kaybettiğimiz bir yansıma sürecidir [29].



Şekil 3.1. Kameraların temel çalışma prensibi [28]

Şekil 3.1'de görüldüğü gibi görüntü düzlemine yansıyan nesnenin görüntüsü baş aşağı olarak görünür. Dijital kameralar bu görüntüyü ters çevirerek nesnenin düzgün görüntüsünü elde etmektedir.

Bir kamerada, ışınların toplandığı nokta olan odak noktası ile görüntü düzlemi arasındaki mesafe odak uzaklığı olarak isimlendirilir.



Şekil 3.2. İğne Deliği Kamera Modeli [30]

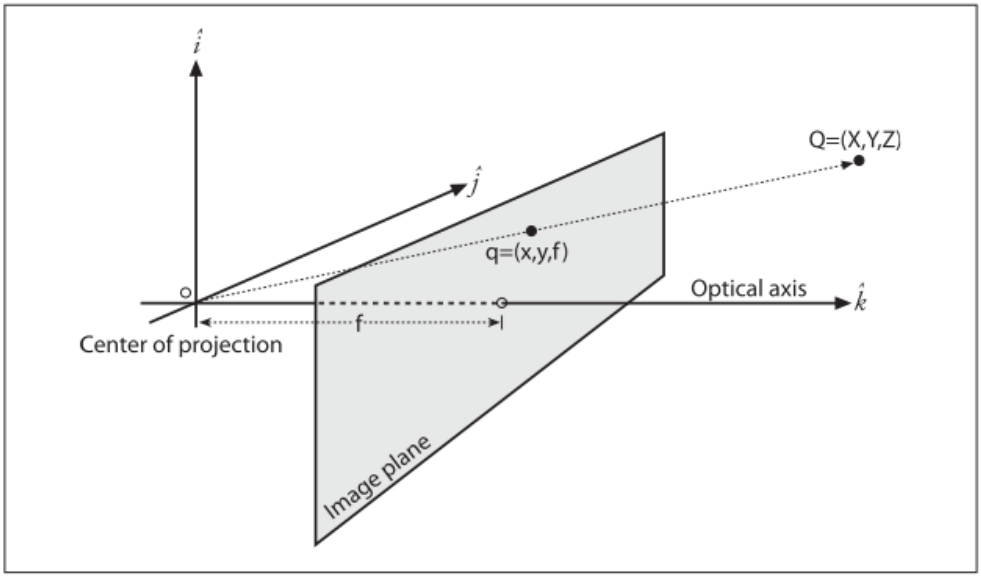
Şekil 3.2'de f odak uzaklığı, Z iğne deliğinden sahnedeki nesneye olan uzaklık, X nesnenin gerçek boyutu, x ise nesnenin görüntü düzlemine yansıyan halinin boyutudur. Şekil 3.2 incelenirse, iğne deliği ile görüntü düzlemindeki görüntünün boyutu ve yine iğne deliği ile gerçek nesne boyutu arasında benzer üçgenler olduğunu fark edilir. Bu üçgenler ve odak uzaklığı mesafesi beraber kullanılarak nesnenin görüntü düzlemindeki boyutu hakkında bilgi edinilebilir. Benzer üçgenler teoremi kullanılarak aşağıdaki gibi bir eşitsizlik yazılabilir:

$$-x = f\frac{X}{Z} \tag{3.1}$$

Örnek bir görüntü boyutu hesaplaması denklem 3.1'de verilmiştir. Burada x nesnenin görüntü düzlemine yansıyan boyutu, f odak uzaklığı, X nesnenin gerçek boyutu, Z ise nesne ile iğne deliği arasındaki mesafedir.

Odak uzaklığı nesnenin görüntü düzlemine yansıyan boyutu ile doğru orantılıdır. Odak uzaklığı arttıkça nesnenin görüntü düzlemindeki boyutu artar. Fakat odak uzaklığını arttırmak kameranın görüş açısını düşürecektir. Odak uzaklığı azaltılırsa kameranın görüş açışı artacaktır. Günümüzde fotoğrafçılar görüntüleyecekleri nesnelerin kameraya olan uzaklığı üzerinden odak uzaklığı ayarını yaparlar. Uzaktaki bir nesneyi görüntülemek için odak uzaklığı arttırılır, yakındaki bir nesneyi görüntülemek için ise odak uzaklığı azaltılır.

Şekil 3.2'deki iğne deliği ile görüntü düzlemini yer değiştirilirse, iğne deliği projeksiyonun merkezi olarak kabul edilebilir. Bu formda görüntü düzlemi ile optik eksenin kesişme noktasındaki noktaya ana nokta (principal point) denir. Şekil 3.2'deki görüntü düzleminin eşdeğeri olan bu yeni önden görüntü düzleminde uzaktaki nesnenin görüntüsü Şekil 3.1'deki görüntü düzlemindeki görüntü ile tam olarak aynı boyuttadır.



Şekil 3.3. İğne Deliği İle Görüntü Düzleminin Yer Değişmiş Hali [30]

Şekil 3.3'te görüldüğü gibi görüntü düzlemi artık iğne deliği ile obje arasına alınmıştır. İğne deliği ile görüntü düzlemindeki mesafe olan f odak uzaklığı sabit kalmış ve değişmemiştir. Görüntü Q gerçek noktasından yansıyan ışınların görüntü düzlemi ile kesişmesiyle oluşur. Bu durumda denklem 3.1'de ifade edilen x değeri negatif olmaktan kurtulur çünkü görüntü artık baş aşağı oluşmaz.

Optik eksenin görüntü düzleminin tam merkezinden geçtiği düşünülebilir. Temel kamera modelimiz için iğne deliğinin içinden geçen optik eksenin görüntü düzleminin tam merkezinden geçmesi için görüntü düzleminin mikron düzeyinde mükemmel bir hizalama ile yerleştirilmiş olması gerekir. Fakat pratikte böyle olmaz. Büyük bir olasılıkla optik eksen görüntü düzleminin tam merkezinden değil, yatay ve dikey olarak merkezin bir miktar uzağından geçer. Bu nedenle bu durumun modellenebilmesi ve hesaplamaların doğru sonuç vermesi sağlanmak için c_x ve c_y değişkenlerini ortaya konulmuştur. Bu değişkenler optik eksenin görüntü düzleminin merkezinin ne kadar uzağından geçtiğini yatay ve dikey olarak ifade eden değişkenlerdir. Bu değişkenleri kullanarak denklem 3.1 şu şekilde güncelleştirilebilir.

$$x_{screen} = f_x \left(\frac{X}{Z}\right) + c_x \tag{3.2}$$

Aynı denklem dikey olarak ifade edilebilir.

$$y_{screen} = f_y \left(\frac{X}{Z}\right) + c_y \tag{3.3}$$

Yukarıdaki denklemlerde f_x ve f_y olmak üzere iki farklı odak uzaklığı söz konusudur. Metrik olarak odak uzaklığı tek bir değerdir. Yukarıdaki denklemlerde iki farklı odak uzaklığı belirtilmesinin sebebi görüntü düzlemindeki piksellerin tam olarak bir kare olmamasından, bir miktar dikdörtgen olmasından kaynaklıdır. Bu durumda gerçekte tek bir değer olan odak uzaklığı x cinsinden f_x, y cinsinden ise f_y olarak ifade edilmelidir.

3.1.1. Kamera Kalibrasyonu

Günümüzde üretim masraflarını azaltmak bakımından oldukça ucuz kameralar üretilmektedir. Bu nedenle bu kameralarla çekilen görüntülerde birtakım bozulmalar meydana gelmektedir. Bu bozulmaların önüne geçmek için kamera kalibrasyonu denilen işlemin yapılması gerekir. Lens kaynaklı veya görüntü düzleminin hatalı hizalanmasıyla ilgili bozulmalar söz konusu olabilir.

Teorik olarak, hiçbir bozulmaya yol açmayacak bir lens tanımlamak mümkündür. Ancak pratikte hiçbir lens mükemmel değildir. Bu, esas olarak imalat nedenlerinden dolayıdır; "küresel" bir mercek yapmak, matematiksel açıdan daha ideal "parabolik" bir mercek yapmaktan çok daha kolaydır. Lensi ve görüntü düzlemini mekanik açıdan tam olarak hizalamak da zordur [30].

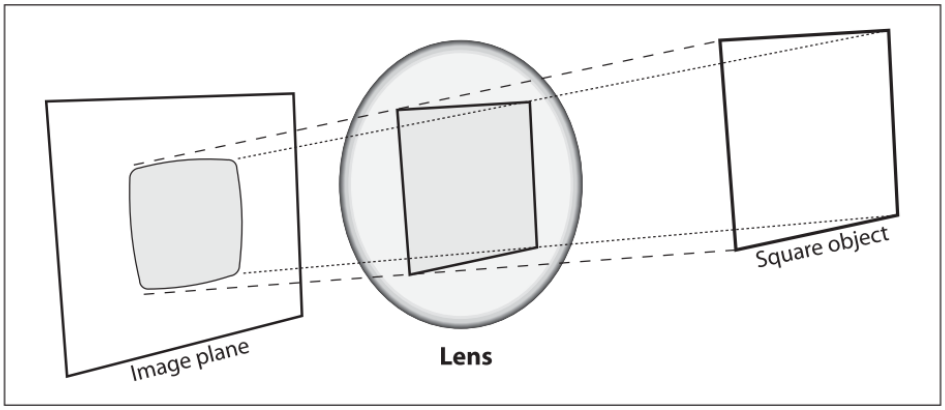
Lens kaynaklı iki ana bozulma türü söz konusudur. Bunlardan birincisi lensin şeklinden kaynaklanan radyal bozulmalar, ikincisi ise kameranın bir bütün olarak montajından kaynaklanan teğetsel bozulmalardır.

Radyal distorsiyon, kameradaki lenslerden kaynaklı olarak, özellikle görüntü düzleminin kenarlarına yakın bölgelerdeki piksellerin konumlarının bozulması sebebiyle meydana gelir. Bu durum barrel veya balık-gözü etkisi olarak isimlendirilir.



Şekil 3.4. Radyal Bozulma Etkisi [29]

Şekil 3.4'e bakılırsa görüntünün kenar bölgelerinde düz olması gereken hatların bir miktar eğik olduğunu fark edilir.



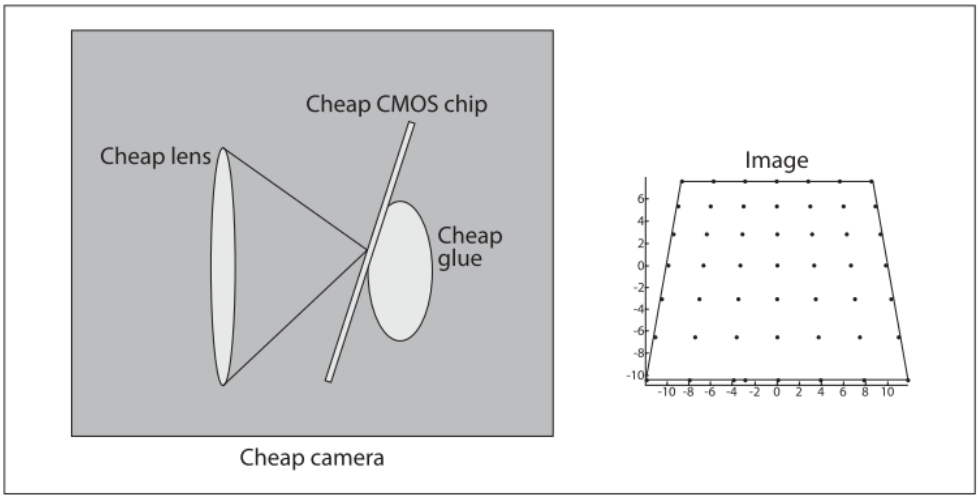
Şekil 3.5. Radyal Bozulma [30]

Radyal distorsiyon kaynaklı hatalar Taylor Serisinin ilk birkaç karakteri ile ifade edilebilir. Taylor serisi karmaşık matematiksel fonksiyonları polinom serisi olarak ifade etmek için kullanılan matematiksel bir tekniktir. Taylor serisi matematikte, bir fonksiyonun, o fonksiyonun terimlerinin tek bir noktadaki türev değerlerinden hesaplanan sonsuz toplamı şeklinde yazılması şeklindeki gösterimi/açılımıdır. Adını İngiliz matematikçi Brook Taylor'dan almıştır. Bu teknik kullanılarak radyal bozulmalar aşağıdaki eşitsizliklerle ifade edilebilir:

$$x_{corrected} = x(1 + k_1 r^2 + k_2 r^4 + k_3 r^6)$$
$$y_{corrected} = y(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \quad (3.4)$$

Burada k, Taylor serisinden gelen, r ise merceğin merkezinden ne kadar uzaklaştığımızı ifade eden terimlerdir. Eşitliğin sağ tarafında yer alan x ve y değerleri bozulmaya uğramış görüntünün görüntü düzlemindeki x ve y değerleridir. Eşitliğin sol tarafındaki $x_{corrected}$ ve $y_{corrected}$ ise düzeltmenin sonucu olan doğru konumlardır.

Diğer bir bozulma türü olan teğetsel bozulma, kameranın üretim aşamasında, lensin görüntü düzlemine tam olarak paralel hizalanamamasından kaynaklanan bozulma türüdür. Şekil 3.6'da görüldüğü gibi görüntü düzlemi ile mercek birbirlerine tam olarak hizalanmazlarsa kameranın ürettiği görüntüde bozulma meydana gelir.



Şekil 3.6. Teğetsel Bozulma [30]

Teğetsel bozulma kaynaklı hatalar, p1 ve p2 parametrelerinin kullanıldığı denklem 3.5, ile ifade edilebilir. Bu denklemler 1966 yılında D. C. Brown tarafından yapılan "Decentering Distortion of Lenses" [31] isimli çalışmada ortaya konulmuştur.

$$x_{corrected} = x + [2p_1 y + p_2(r^2 + 2x^2)]$$
$$y_{corrected} = y + [p_1(r^2 + 2y^2) + 2p_2 x] \quad (3.5)$$

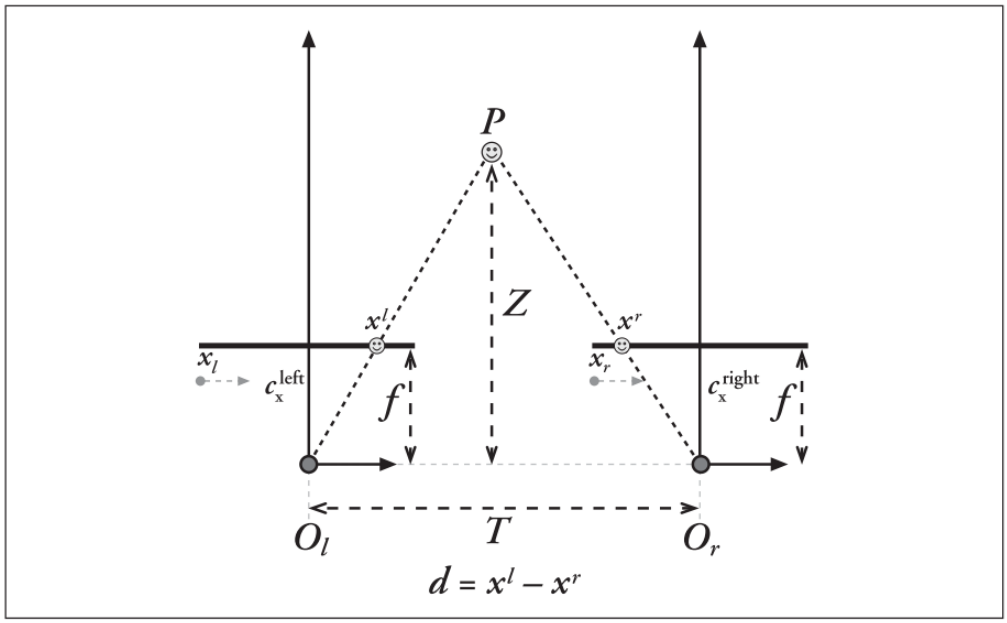
Denklem 3.4 ve denklem 3.5'te görüldüğü gibi, radyal ve teğetsel bozulmalar olarak ifade edilen bozulma türlerini ifade eden denklemlerde toplamda beş adet parametre bulunur: k1, k2, k3, p1 ve p2. Bir kamerayı kalibre edebilmek için bu değerlerin belirlenmesi gerekir. Kalibrasyon fonksiyonları sayesinde bu değerler kolayca belirlenebilir. Tüm bu değerler kameranın içsel parametreleri olarak ifade edilir. f_x, f_y, c_x, c_y değerleri 3x3 boyutunda bir matrisle, . k1, k2, k3, p1 ve p2 ise 1x5 boyutunda tek bir matrisle ifade edilir.

$$\begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \qquad \begin{bmatrix} k_1 & k_2 & p_1 & p_2 & k_3 \end{bmatrix}$$

Şekil 3.7. Kamera İçsel Matrisi Ve Bozulma Katsayıları

3.2. Stereo Görme

Stereo görme, birbirlerine göre konumları ve projeksiyonları bilinen iki kamera tarafından görüntülenen bir noktanın kamera düzlemine olan mesafesini ve konumunu hesaplamak için kullanılan bir tekniktir.



Şekil 3.8. Stereo Düzenek geometrisi[30]

Şekil 3.8'de O_l ve O_r her iki kameranın merkez noktaları, P uzaydaki herhangi bir nokta, x_l ve x_r P noktasının her iki görüntü düzlemindeki yatay koordinatı, f kameraların odak uzaklığı, Z, P noktasının kamera düzlemine olan uzaklığı, T ise kameraların merkez noktaları arasındaki uzaklık mesafesidir. d ise P noktasının her iki kameranın görüntü düzlemindeki yatay koordinatların mutlak farkıdır. c_x ve c_y ise her iki görüntü

düzlemindeki merkez noktalardır (principal point). Bu noktaların teoride görüntü düzlemlerinin tam merkezinde yer alması gerekir. Fakat bu noktalar kameraların üretim koşulları nedeniyle genellikle görüntü düzlemi merkezinin bir miktar uzağında bulunur. Bu değerlerin tam olarak görüntü düzleminin merkezinde yer aldığı varsayılırsa, tüm bu değerler ile benzer üçgenler teoremi kullanılarak P noktasının kamera düzlemine olan mesafesi, yani Z değeri denklem 3.6 ile hesaplanabilir.

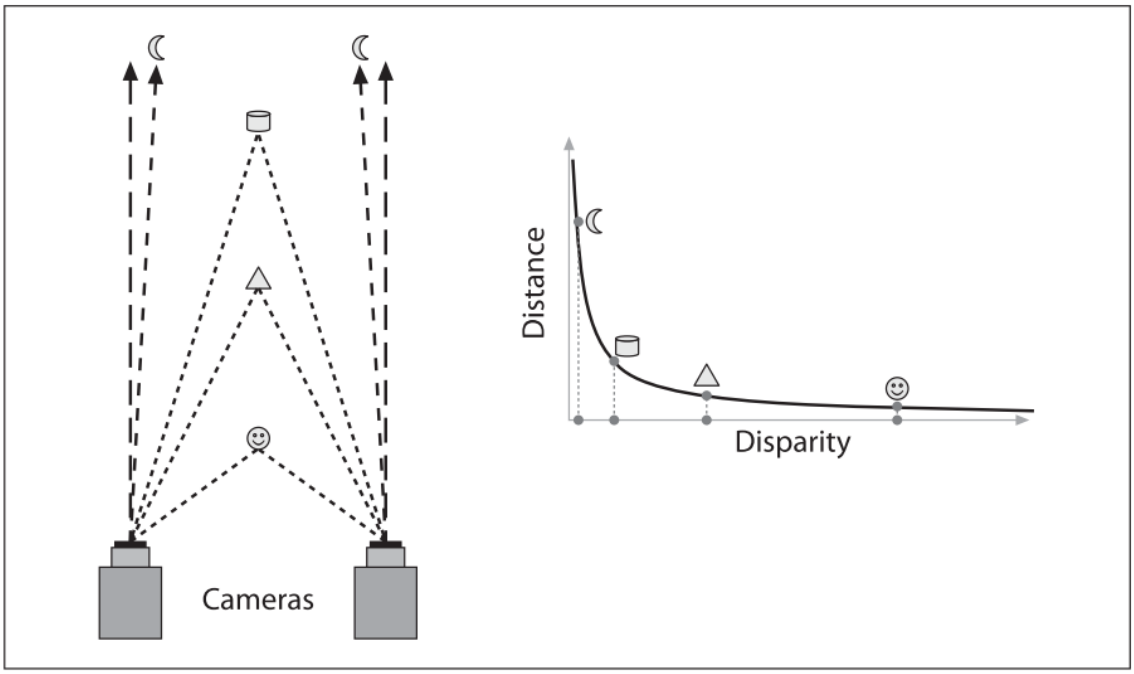
$$\frac{T-(x^l-x^r)}{Z-f}=\frac{T}{Z}\rightarrow Z=\frac{fT}{x^l-x^r} \tag{3.6}$$

Şekil 3.8'de $x^l - x^r$ değeri d değişkeni ile ifade edilmiştir. Dolayısı ile formül basitçe denklem 3.7'deki gibi yazılabilir:

$$Z=\frac{fT}{d} \tag{3.7}$$

Stereo görmede d değeri aynı zamanda disparity(eşitsizlik) olarak ifade edilir.

Stereo görmede kameralara uzak olan nesnelerin disparity değeri, kameraya yakın olan nesnelerin disparity değerinden daha düşüktür. Yani bir nesne stereo kamera düzeneğinden uzaklaştıkça nesnenin her iki kameranın görüntü düzlemindeki konumu görüntü düzleminin merkezine doğru yaklaşır. Buda bu nesnenin her iki kamera düzlemindeki konumlarının farkını, yani xl-xr değerini azaltır. Bu noktada Z derinliği ile d eşitsizlik değerlerinin birbirleri ile doğrusal olmayan bir biçimde ters orantılı olduğu görülür. Bu orantının doğrusal olmamasının nedeni nesnenin belli bir uzaklık mesafesinden sonra disparity değerinde fazla bir değişikliğe sebep olmamasıdır. Yakın mesafelerde disparity değeri çok değişiklik gösterirken uzak mesafelerdeki değişim disparity üzerinde fazla bir değişikliğe sebep olmayacaktır. Şekil 3.9'da bu durum grafiksel olarak ifade edilmiştir.

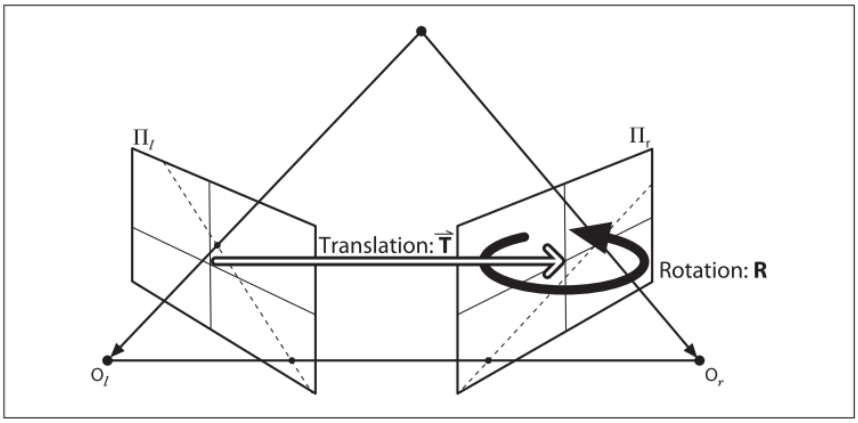


Şekil 3.9. Mesafe Ve Disparity İlişkisi[30]

Bu durumun bir sonucu olarak stereo görmenin kamera düzlemine nispeten yakın nesneler için daha doğru çalışacağı sonucu çıkarılabilir.

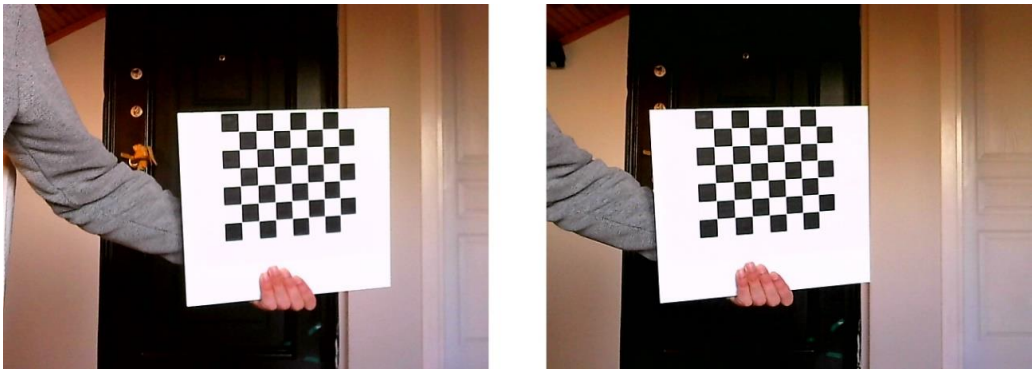
3.2.1. Stereo Kalibrasyon

Stereo kalibrasyon uzaydaki iki kamera arasındaki geometrik ilişkiyi hesaplama işlemidir [30]. Stereo kalibrasyondaki temel amaç şekil 3.10'de görüldüğü gibi iki kamera arasındaki çeviri matrisi(R) ve öteleme vektörünün bulunmasıdır.



Şekil 3.10. İki Kamera Arasındaki Geometrik İlişki[30]

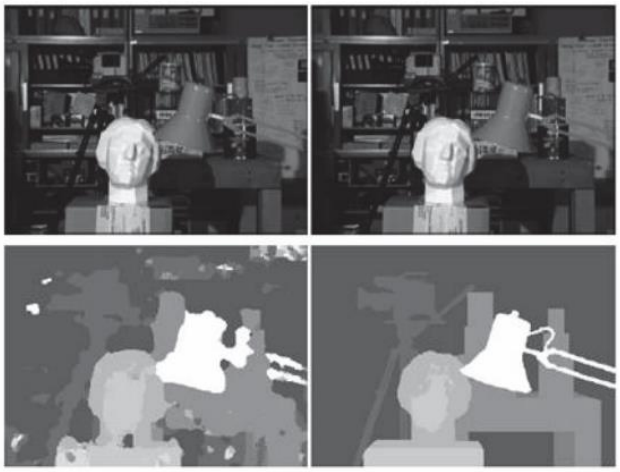
Stereo kalibrasyon yapmak için tek kamera kalibrasyonundaki gibi bir satranç tahtası kullanılabilir. Buradaki tek fark görüntülerin her iki kamera ile senkronize bir şekilde alınmasıdır.



Şekil 3.11. Stereo Kamera Düzeneği İle Çekilmiş Senkronize Görüntü

Stereo kalibrasyon işlemini yapmadan önce her iki kameranın daha önce anlatıldığı gibi tek tek kalibre edilmesi ve kalibrasyon değerlerinin elde edilmesi gerekir. Stereo kalibrasyon işlemi, kameraların bu değerlerini parametre olarak isteyecektir.

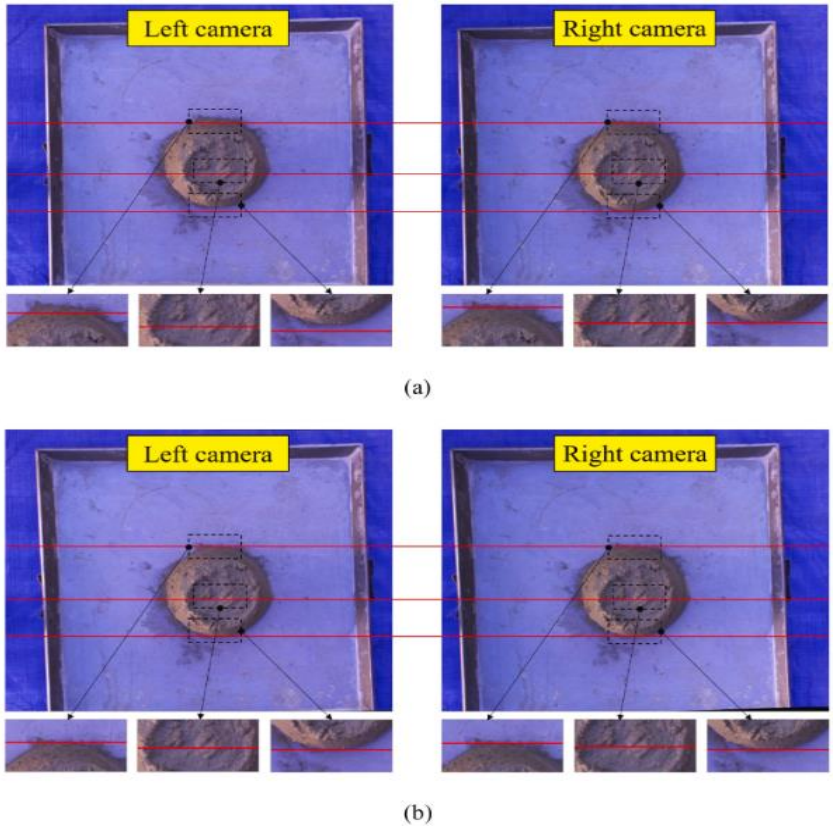
Kalibrasyon işlemi için farklı mesafe ve açılardan senkronize olarak çekilecek 15-20 adet görüntü yeterli olacaktır. Stereo kalibrasyon fonksiyonu bu görüntüleri kullanarak stereo kalibrasyon matrisini oluşturur.



Şekil 3.12. Eşitsizlik Haritası (Sol) Ve Derinlik Haritası (Sağ)[36]

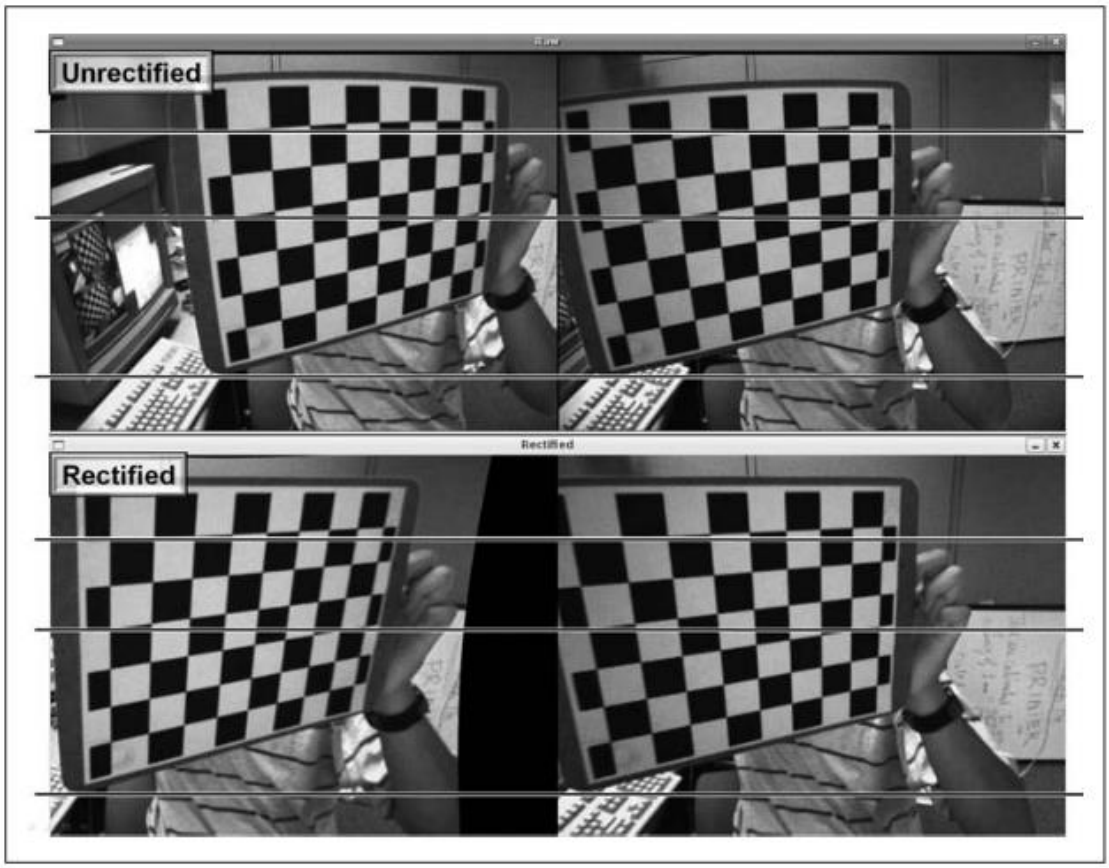
3.2.2. Stereo Düzeltme

Stereo düzeltmenin amacı her iki kamera görüntüsünü hem bozulmalardan kurtarmak hem de birbirleri ile tam olarak aynı düzleme getirmektir. Bunun yanı sıra stereo düzeltme görüntüleri dikey olarak aynı hizaya getirir. Böylece derinlik hesaplaması iki boyuttan bir boyuta iner ve hesaplama maliyetimiz önemli ölçüde azalır. Şekil 3.13-(b)'de dikey olarak hizalanmış stereo görüntü çifti görülmektedir.



Şekil 3.13. Orijinal Görüntü(a) Hizalanmış Görüntü(b)[34]

Derinlik hesaplamasının iki boyuttan bir boyuta inmesi, aynı noktayı iki farklı kameradan gelen görüntülerde ararken, diğer bir deyişle iki farklı noktayı eşleştirmeye çalışırken sadece bir satırda arama yapmanın yeterli olacağı anlamına gelir. Şekil 3.13'de stereo düzenekten gelen görüntüler bulunmaktadır. Üst kısımda bu görüntülerin stereo düzeltme yapılmamış hali görülmektedir. Alt bölümde ise bu görüntülere stereo düzeltme işleminin yapılmış hali görülmektedir. Düzeltilmiş görüntülerde hem radyal ve teğetsel bozulmalar düzeltilmiş hem de görüntüler dikey olarak hizalanmıştır.



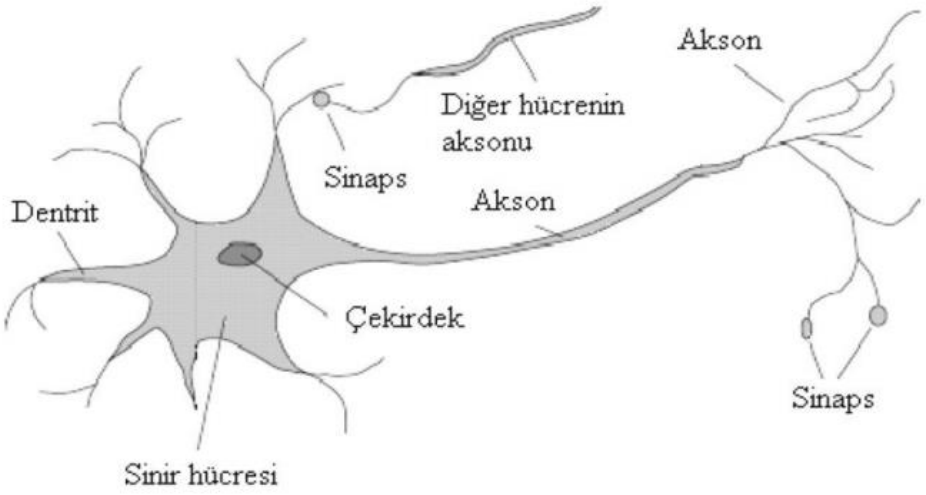
Şekil 3.14. Stereo Düzeltme işlemi [30]

3.3. Yapay Sinir Ağları

Yapay sinir ağları, hassas öğrenme yetenekleri ve her türdeki fonksiyonla ilgili hesaplamalar yapabilmeleri sebebiyle çok tercih edilen bir veri işleme tekniğidir. Klasik bir bilgisayar programı program akışı boyunca kendisine bildirildiği gibi verileri işler ve sonuç üretir. Yapay sinir ağları ise sinir ağı modeli oluşturulduktan sonra önceki girdiler ve çıktılar ile eğitilir. Eğitildikten sonra kendisine verilen yeni girdi verilerine göre en uygun çıktıyı hesaplayarak çalışırlar.

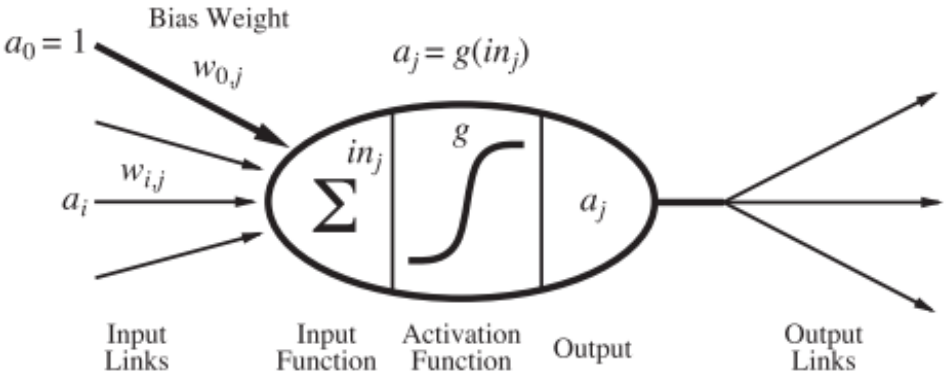
Yapay sinir ağlarının temelinde insan beynindeki sinir sistemini oluşturan nöronların çalışma mantığı yatar. İnsan beynindeki tek bir nöron, kendisine gelen sinyal belli bir eşik değerini aştığında tepki verir ve kendisine gelen sinyali bir sonraki nörona iletir. Bir nöron temel olarak hücre gövdesi, akson ve dendrit olmak üzere 3 bölüme ayrılır. Hücre gövdesi hücreyi denetler ve hücre etkinliklerinden sorumludur. Dendritler

hücre gövdesinden dışarı doğru uzanan uzantılardır. Diğer nöronlardan gelen bilgileri hücre gövdesine aktarırlar. Aksonlar ise nörondaki bilgiyi diğer nöronlara iletmekle sorumludur. Biyolojik bir sinir hücresinin temel yapısı şekil 3.15'de görülmektedir.



Şekil 3.15. Biyolojik Bir Sinir Hücresinin Yapısı

Yapay sinir ağları ile ilgili ilk çalışmalar 1943 yılında Mc Colloch-Pitts modeli ile başlamıştır. Teknolojinin gelişmesi ve yüksek hesap kapasiteli bilgisayarların üretilmesi ile 1980'li yıllarda tekrar hız kazanmıştır [39].



Şekil 3.16. Bir Nöronun Matematiksel Modellemesi [40]

Şekil 3.16'da bir nöronun matematiksel modellemesi görülmektedir. Buna aynı zamanda perceptron (algılayıcı)'da denmektedir. İnsan beynindeki gerçek nöronlarda

dendrit, diğer nöronların aksonlarından elektrik sinyalleri alırken, algılayıcıda bu elektrik sinyalleri sayısal değerler olarak temsil edilir.

Basit bir algılayıcıda hesaplama denklem 3.8'deki gibi gerçekleşir.

$$a_j = g\left(\sum_{i=0}^{n} w_{i,j}, a_i\right) \tag{3.8}$$

Denklem 3.8'de a_j algılayıcının çıkış değeri, g algılayıcıda kullanılan fonksiyon, a_i i indis numaralı girişten gelen değer, $w_{i,j}$ ise i indis numaralı bağlantı üzerindeki ağırlık değeridir. Bir sinir ağında ağırlıklar, belli koşullar göz önünde bulundurularak rastgele başlatılır.

Sinir ağlarında aktivasyon fonksiyonu olarak farklı fonksiyonlar kullanılabilir. Bu fonksiyonların birbirlerine göre avantajları ve dezavantajları bulunur. Hangi fonksiyonun seçileceği çözülecek problemin niteliğine göre belirlenir.

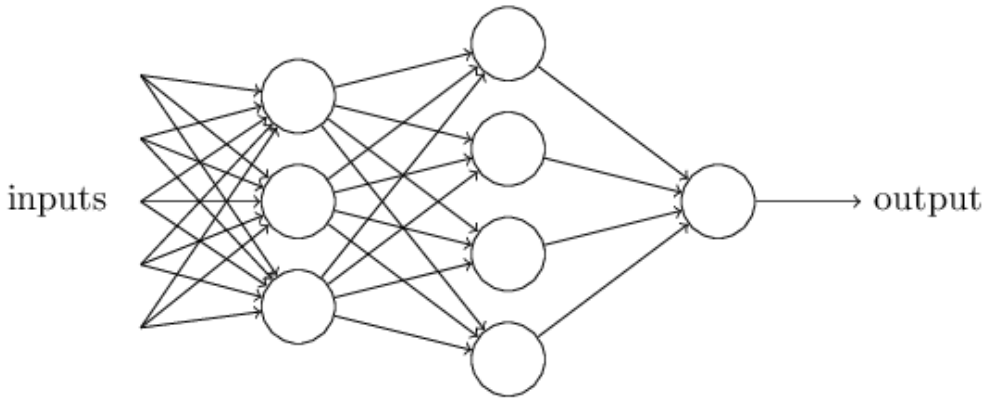
Şekil 3.16'da ayrıca Bias weight (önyargı ağırlığı) olarak adlandırılan bir parametre daha bulunur. 1950'lerin sonlarında, Frank Rosenblatt, Mc Colloch-Pitts modelinin geliştirilmiş versiyonu olan bir ağ tanıtmışlardır. Bu çalışmada Rosenblatt, Mc Colloch-Pitts modeline ek olarak sapmayı temsil eden, önyargı ağırlığı (Bias weight) isminde ekstra bir girdi eklemiştir. Dolayısı ile denklem aşağıdaki hale gelmiştir.

$$a_j = g\left(\sum_{i=0}^{n} w_{i,j}, a_i\right) + b \tag{3.9}$$

Şekil 3.16'da ayrıca algılayıcının merkezinde bir aktivasyon fonksiyonu olduğu görülmektedir. Algılayıcılarda farklı tipte aktivasyon fonksiyonları kullanılabilir.

Bazı durumlarda ağın çok daha hassas çalışması istenebilir. Örneğin tek bir hesaplama katmanından oluşan bir yapay sinir ağı kullanılarak lojik AND ve OR problemleri çözülebilir. Bunun sebebi AND ve OR kapılarının tek bir karar yüzeyi ile çözülebilmesidir. Fakat XOR problemini çözmek için bir tek karar yüzeyi yeterli

olmayacaktır. Bu durumda yapay sinir ağımızın yapısını geliştirerek hesaplama katmanı ile çıkış katmanı arasına yeni bir katman eklememiz gerekecektir. Eklenen bu katman ile artık sinir ağımız çok katmanlı yapay sinir ağı olarak adlandırılır.

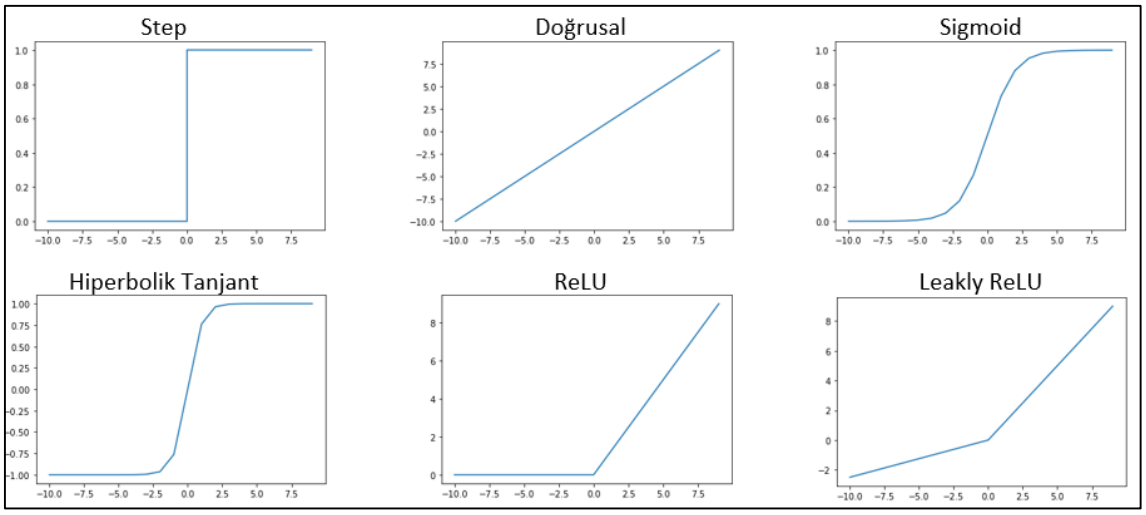


Şekil 3.17. Çok Katmanlı Yapay Sinir Ağı

Eklenen ara katmanlar aynı zamanda gizli katman olarak adlandırılır.

3.3.1. Aktivasyon Fonksiyonları

Sinir ağlarında bulunan nöronlar için birçok farklı aktivasyon fonksiyonu bulunmaktadır. Çözülecek problemin niteliğine göre uygun fonksiyonu seçmek ağın etkinliği açısından son derece önemlidir. Bu fonksiyonların birbirlerine göre avantaj ve dezavantajları bulunur. Uzun süre kullanılan sigmoid fonksiyonuna yapay sinir ağının öğrenmesi yönünden oldukça başarılı bir fonksiyon olmasına rağmen hem hesaplama maliyetinin yüksek olması hem de yapısı gereği ağın doygunluk problemine düşmesine sebep olduğu için temkinli yaklaşılır. Zaman içerisinde buna benzer dezavantajları ortadan kaldırmak için birçok fonksiyon ortaya konulmuştur.



Şekil 3.18. Aktivasyon Fonksiyonları

Step Fonksiyonu: İkili değer üreten bir fonksiyondur. Giriş değeri eşik değerini aştığı zaman çıkış üreten, aşmadığı zaman çıkış üretmeyen bir fonksiyondur. Giriş değerine göre sadece iki çıkış üretilmesi gereken problemlerde yani ikili sınıflandırma problemlerinde kullanılabilir. Fonksiyonun türevi öğrenme değeri temsil etmediği için öğrenme algoritmalarında kullanılmaz.

Doğrusal(Linear) Fonksiyon: Basamak fonksiyonundan farklı olarak bir dizi aktivasyon değeri üretir. Bir yapay sinir ağında birkaç nöronu birbirine bağlamak için kullanılabilir. Türevi basamak fonksiyonunda olduğu gibi sabittir ve öğrenme değeri temsil etmeyeceği için öğrenme algoritmalarında kullanılamaz.

Sigmoid Fonksiyon: Doğrusal olmayan bir fonksiyondur. Bu nedenle doğrusal olmayan birçok problemin çözümünde kullanılabilir. En önemlisi herhangi bir noktasında türevinin alınabilmesidir. Bu nedenle öğrenme algoritmalarında kullanılabilir. 0 ile 1 arasında bir çıkış değeri üretir. Sigmoid fonksiyonunun en büyük sorunu giriş değeri belli bir aralığın üstünde veya altında olduğunda fonksiyonun çok az tepki vermesidir. Böyle durumlarda fonksiyon doyuma ulaşır (saturation) ve bu aralığın dışında türevi çok düşük olur. Bu nedenle sinir ağlarında öğrenmeye etkisi çok az olur. Bu duruma sinir ağlarında gradyanların ölmesi/kaybolması (vanishing gradient) problemi denir.

Hiperbolik Tanjant Fonksiyonu: Sigmoid fonksiyonuna oldukça benzer bir yapıya sahiptir. Sigmoid fonksiyondan farkı ürettiği çıkış aralığının -1 ile +1 arasında olmasıdır.

Sigmoid fonksiyonuna göre avantajı ise türevinin daha dik olması yani daha çok değer alabilmesidir. Bu da öğrenme hızını arttırabilme anlamına gelmektedir. Sigmoid fonksiyonda olduğu gibi giriş belli bir aralığın dışında olduğunda vanishing gradient problemi söz konusu olur. Hesaplama maliyeti sigmoid fonksiyonda olduğu gibi yüksektir.

ReLU (Rectified Linear Unit) Fonksiyonu: Pozitif eksende doğrusal fonksiyon ile aynı şekilde çalışır. Negatif eksende ise sıfır değerini üretir. Hesaplama maliyeti düşük olduğu için çok fazla nöronlu ağlarda kullanışlıdır. Pozitif eksende doygunluk göstermez dolayısı ile vanishing gradient problemini önler. Bu fonksiyonun kombinasyonları ile herhangi başka bir fonksiyona yakınsamak mümkündür. Negatif eksende 0 değerlerini alması ağın daha hızlı çalışacağı anlamına da gelmektedir. Fakat bu, aynı zamanda negatif eksende öğrenme işleminin gerçekleşmeyeceği anlamına gelir. Bazı durumlarda öğrenme sırasında önyargı değerleri negatif değerler alabilir. Bu negatif değer yüksek olursa bağlı olduğu nöron, ağırlıklar pozitif olsa bile toplamda negatif değer üretir. Bu da nöronun çıktısının sıfır olmasına sebep olur. Bu durumda o nöron bir anlamda ölü nöron olarak kabul edilir. Çünkü çıktısı sıfır olan bir fonksiyonun türevi de sıfır olur. Bu durumda o nörona bağlı olan ağırlıklar ve önyargı güncellenmeyecektir.

Sızdıran (Leakly) ReLU fonksiyonu: ReLU fonksiyonunun en büyük problemi negatif eksende öğrenmenin gerçekleşmiyor olmasıydı. Sızdıran ReLU fonksiyonu bu sorunun üstesinden gelmek için geliştirildi. Negatif eksende fonksiyona sızdırma değeri adında bir değer vererek o bölgede fonksiyonunun türevinin sıfır olmasını engelleriz. Bu sayede negatif eksende de öğrenme işlemi gerçekleşir.

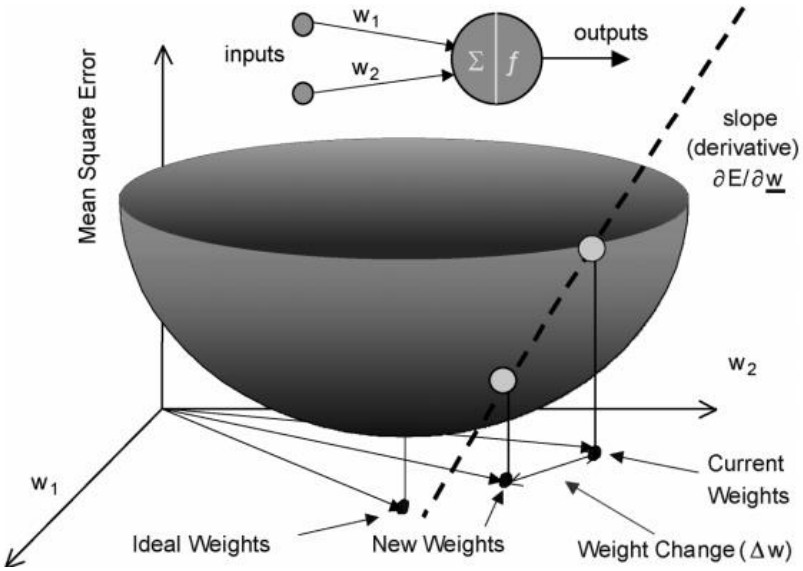
3.3.2. İleri Beslemeli Ağlar

Temel olarak ileri beslemeli sinir ağlarında bir katmandaki çıkışlar bir sonraki katmanın girişi olarak değerlendirilir. Bu, ağda herhangi bir döngü olmadığı anlamına gelir. Hesaplamalar girişten çıkışa doğru tek yönlüdür. Bilgi her zaman ileri doğru beslenir. Herhangi bir geri besleme söz konusu değildir.

3.3.3. Geri Beslemeli Ağlar

Geri yayılımlı öğrenme algoritması, bir sinir ağının öğrenme sürecinin en önemli noktasıdır. Eğitim sürecinde beklenen çıktı ile gerçekleşen çıktı arasındaki farktan yola çıkılarak geriye doğru zincirleme yöntemi ile ağın ağırlıklarının güncellenmesi sürecidir. Geri yayılımlı öğrenmede aktivasyon fonksiyonu olarak genellikle 0 ile 1 arasında gerçel bir değer üreten sigmoid fonksiyonu kullanılır [39]. Geri yayılma süreci, doğrudan genel hata gradyanının türetilmesinden ortaya çıkar [40].

Bir yapay sinir ağında geri yayılım algoritması yeteri kadar tekrar edilirse ağ bir süre sonra doğru çıkışları üretmeye başlar. Tüm bu süreç dereceli alçalma olarak ifade edilir. Şekil 3.19 ortalama kare hatası için dereceli alçalmayı göstermektedir.



Şekil 3.19. Ortalama Kare Hatası İçin Dereceli Alçalma[38]

3.4. Konvolüsyonel Ağlar

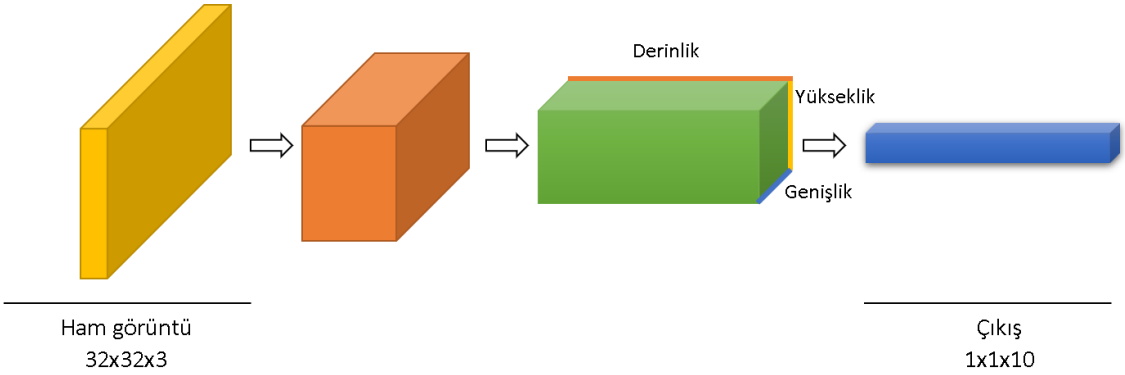
Konvolüsyonel ağlar yapay sinir ağlarına benzer yapıya sahiptir. Öğrenebilen ağırlıklar ve önyargı değerleri ile tıpkı yapay sinir ağları gibi çalışırlar. Konvolüsyonel ağdaki bir nöron, yapay sinir ağında olduğu gibi kendisine gelen girişleri toplar, bir aktivasyon fonksiyonuna sokar ve bir sonuç üretir. Bir konvolüsyonel ağ girdi olarak ham görüntü bilgilerini alır, bunları işler ve çıkışta sınıflandırma puanları oluşturur. Çıkış

katmanında tıpkı yapay sinir ağlarında olduğu gibi bir kayıp fonksiyonu söz konusudur. Öğrenme, yapay sinir ağlarındakine benzer şekilde gerçekleşir.

Konvolüsyonel ağlar girdilerin görüntü olacağını varsayar. Yapay sinir ağları ise girdi olarak tek bir vektör alır. Bir görüntü dosyası tek bir vektöre göre genellikle çok daha yoğun veri içerir. Çok sayıda parametre ile işlem yapmak yüksek miktarda işlem gücü gerektirir. Bu durum konvolüsyonel ağların mimarisini ve çalışma şeklini belirleyen önemli bir etkendir. Konvolüsyonel ağların yaptığı en önemli işlerden biri yüksek miktardaki parametre sayısını, girdideki anlamsal bilgiyi mümkün olduğunca koruyarak düşürmektir.

Evrişimli sinir ağının ilk başarılı uygulaması, LeCun ve ark. [41] tarafından gerçekleştirilmiştir. Bu çalışmada posta kodlarını ve rakamları okumak, görüntüdeki birden çok konumdaki özellikleri çıkarmak için LeNet5 adlı bir mimari tanıtılmıştır.

Bir görüntü dosyası genellikle genişlik, yükseklik ve derinlik olmak üzere üç boyutludur. Örneğin, CIFAR-10 veri setindeki görüntüler, 32x32x3 olmak üzere üç boyuta sahiptir. Konvolüsyonel ağlar, girdinin görüntülerden oluştuğu varsayımından yola çıkar. Bu nedenle katmanlarındaki nöronların düzeni üç boyutludur. Konvolüsyonel ağlarda, bir katmandaki nöronlar kendisinden önceki katmanın küçük bir bölgesine bağlanır. Bu bölgeye nöronun alıcı alanı denir. Bir konvolüsyonel ağ sürecinin sonunda CIFAR-10 veri seti için son çıktı katmanı 1x1x10 boyutlarına sahip olacaktır. Üç boyutlu bir girdi, derinlik boyutu boyunca düzenlenmiş tek bir sınıf puanları vektörüne dönüşür.



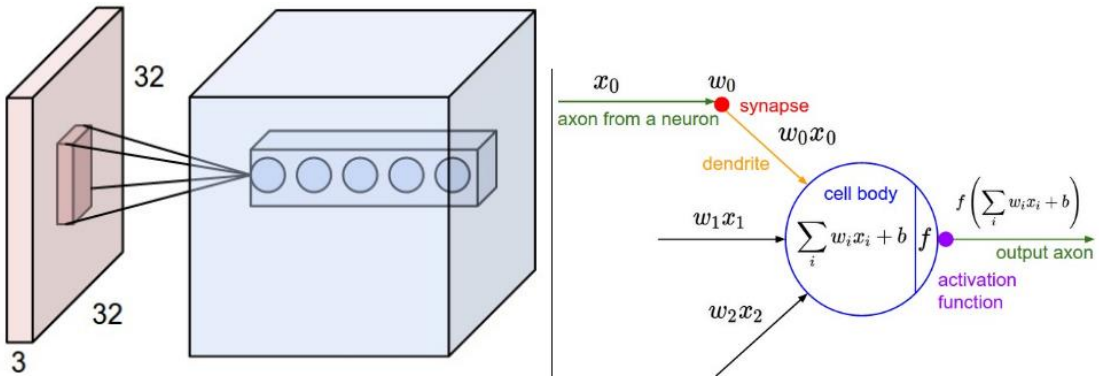
Şekil 3.20. Konvolüsyonel Ağların Temel Çalışma Şekli

3.4.1. Konvolüsyonel Ağı Oluşturan Katmanlar

Konvolüsyonel ağlarda konvolüsyonel (evrişim) katmanı, havuzlama katmanı ve tam bağlı katman olmak üzere üç ana katman söz konusudur. Bunların dışında genellikle evrişim katmanlarını takip eden bir aktivasyon katmanı da bulunur. Bu katmanda genellikle ReLU aktivasyon fonksiyonu kullanılır.

Konvolüsyonel Katman: Giriş görüntüsündeki yerel bölgelere bağlı nöronların çıktısını hesaplar, her biri ağırlıkları ile giriş hacminde bağlı oldukları küçük bir bölge arasında bir iç çarpım hesaplar. Bir konvolüsyonel katman filtrelerden oluşur. Örneğin 32x32x3 boyutunda bir giriş görüntüsüne sekiz filtreden oluşan bir konvolüsyonel katman uygulanorsa 32x32x8 hacminde bir sonuç elde edilir. Konvolüsyonel katmandaki filtreler öğrenebilme yeteneğine sahiptir. Bu filtreler yapay sinir ağlarında nöronlar arasında bulunan öğrenebilen ağırlıklar gibi çalışır. Boyut olarak küçük olmalarına rağmen giriş görüntüsünün derinliği boyunca uzanırlar. İleriye doğru hesaplama sırasında, her bir filtreyi giriş hacminin genişliği ve yüksekliği boyunca kaydırılır. Filtre girişleri ile giriş görüntüsü üzerinde herhangi bir konumdaki giriş arasında çarpım ve toplam yaparak yeni bir değer elde edilir. Filtre giriş hacminin genişliği ve yüksekliği boyunca kaydırıldığında, her uzamsal konumda o filtrenin yanıtlarını veren 2 boyutlu bir aktivasyon haritası oluşur. Bu aktivasyon haritalarının sayısı filtre sayısı kadardır ve konvolüsyonel katman için çıkış hacmini oluşturur.

Bir konvolüsyonel katman için derinlik, adım ve dolgu olarak isimlendirilen hiper parametreler söz konusudur. Bu hiper parametreler katman çıktısının boyutunu etkiler. Derinlik, konvolüsyonel katman çıktısının derinliğini ifade eder, aynı zamanda kullanılan filtre sayısıdır. Örneğin derinlik değeri yani filtre sayısı beş olursa katman çıktısının derinliği beş olacaktır. Bu aynı zamanda giriş görüntüsünün ilgili bölgesi için beş adet aktivasyon haritası olacağı anlamına gelir. Bu noktada girişin ilgili bölgesi için kullanılan filtrelerden her biri girdide farklı bir özelliği arar. Örneğin, birinci evrişimli katman ham görüntüyü girdi olarak alırsa, derinlik boyutu boyunca farklı nöronlar, çeşitli yönlendirilmiş kenarların veya renk lekelerinin varlığında aktif hale gelebilir. Girişin ilgili bölgesi için filtre sayısı kadar nöron hacmi olur ve buna derinlik sütunu denir.



Şekil 3.21. Giriş Katmanı İçin Nöron Hacmi ve Nöron Yapısı[42]

Adım hiper parametresi ise filtrenin giriş görüntüsü üzerinde hareket ettirilirken adım sayısını belirtir. Örneğin adım hiper parametresi bir olarak belirlenirse filtre, giriş görüntüsünde her seferinde bir piksel hareket edecektir. Adım iki olduğunda filtre her seferinde iki piksel arayla hareket edecektir. Bu durumun giriş görüntüsü üzerindeki parametre sayısını azaltacağı açıktır.

Dolgu parametresi ise bir konvolüsyonel katmanın üreteceği çıkış hacminin kontrol edilmesini sağlar. Bazı durumlarda çıktı hacminin giriş hacmi ile aynı uzamsal boyutlarda yani genişlik ve yükseklik değerlerinde olması istenir. Bu durumda giriş görüntüsünün etrafına manuel olarak piksel eklenip çıkış hacminin büyüklüğünü kontrol edebilebilir. Bazı durumlarda ise kullanmak istenilen filtrenin boyutu giriş görüntüsüne uygun olmayabilir. Buna benzer durumlarda dolgu parametresi kullanılarak oluşacak sorunların önüne geçilebilir.

Konvolüsyonel katmanlar için söz konusu olan bu üç hiperparametre değerleri kullanılarak çıkış hacminin uzamsal boyutu yani genişlik ve yükseklik değerini hesaplayan bir fonksiyon ortaya konulabilir. Nöronların alıcı alanı yani filtre boyutu F, adım değeri A, dolgu değeri D, giriş hacminin uzamsal boyutu G, çıkış hacminin uzamsal boyutu Ç olarak ifade edilirse bu durumda Ç'yi hesaplamak için aşağıdaki denklem yazılabilir:

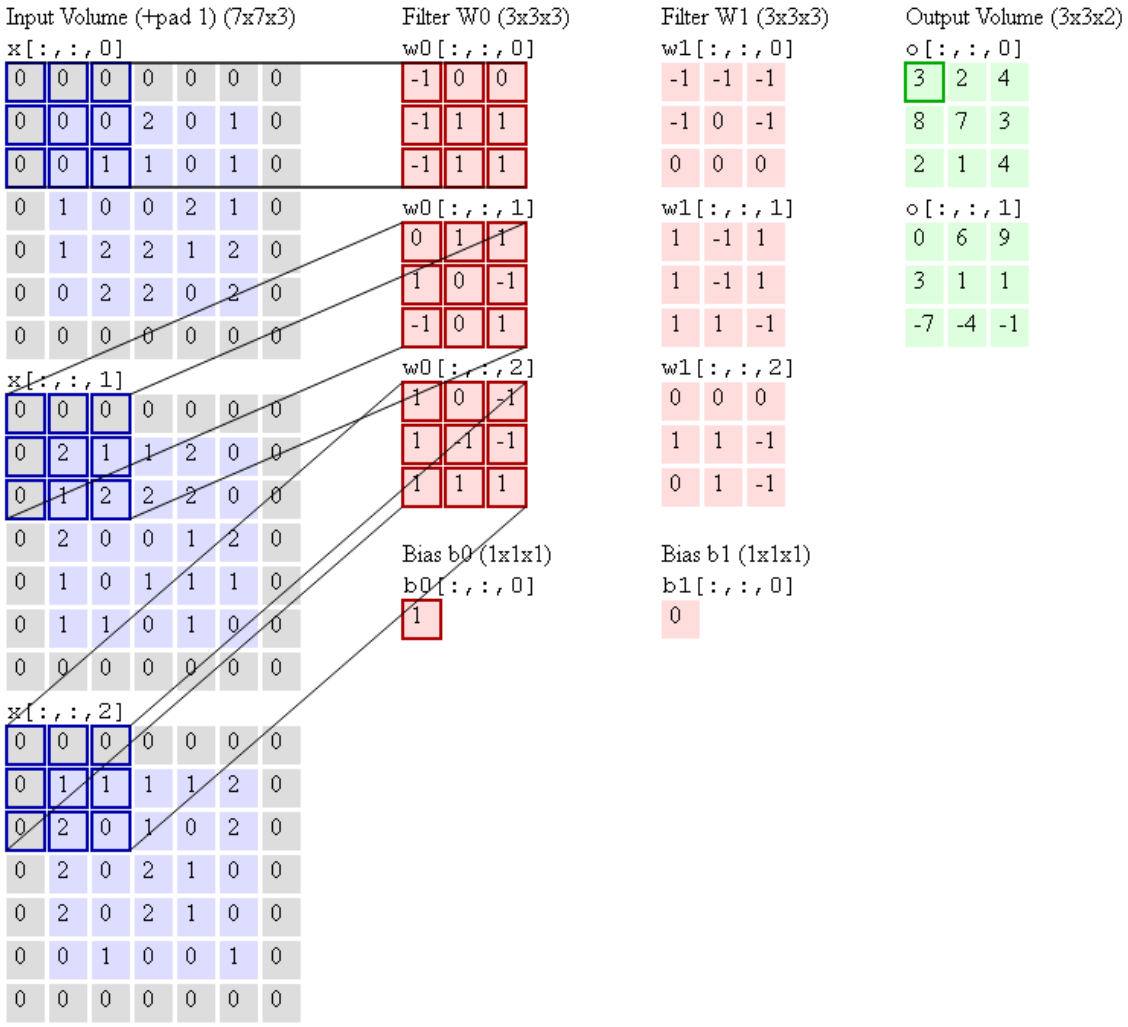
$$Ç = (G - F + 2D) / A + 1 \quad (3.10)$$

Örneğin 10x10 boyutunda giriş hacmi ve 3x3 boyutunda filtre ile birlikte adım değeri 3, dolgu değeri de 1 olmak koşulu ile çıkış hacmi denklem 3.11'deki gibi hesaplanabilir.

$$Ç = (10 – 3 + 2 * 1) / 3 + 1 = 4 \quad (3.11)$$

Bulunan bu 4 değeri yalnızca bir boyut için geçerlidir. Bu durumda çıkış hacminin genişliği ve yüksekliği 4, derinliği ise filtre sayısı kadar olacaktır.

Şekil 3.22'de konvolüsyonel katmanının çalışmasını gösteren şema bulunmaktadır. Bu şemada mavi renkli matrisler giriş hacmini, kırmızı renkli matrisler filtreleri, yeşil renkli matrisler ise çıkış hacimlerini gösterir. Giriş hacminde genişlik ve yükseklik değerleri beş piksel, derinlik değeri ise üçtür. Konvolüsyonel katmanda filtre sayısı iki, filtre genişliği üç, adım değeri iki ve dolgu değeri ise birdir. Her bir filtre için bir adet önyargı değeri bulunmaktadır.



Şekil 3.22. Bir Konvolüsyonel Katmanda Filtreleme[42]

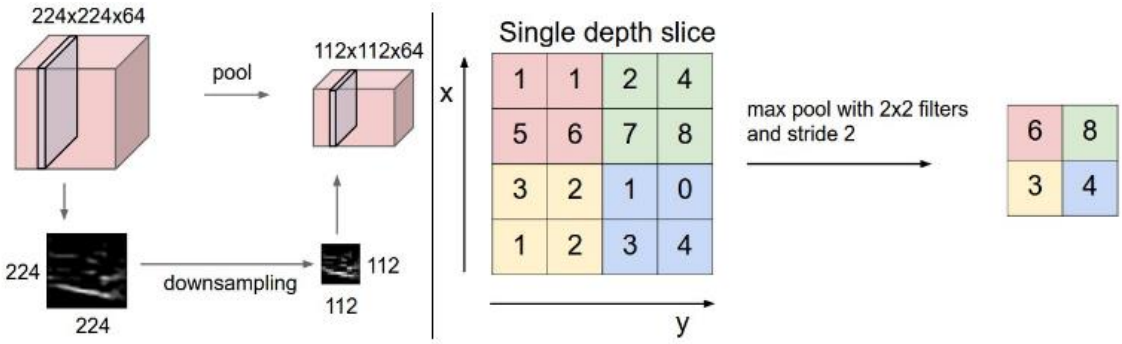
Havuzlama Katmanı: Konvolüsyonel ağlar girdisi görüntüler olan, dolayısıyla yüksek sayıda parametre ile çalışmak durumunda olan ağlardır. Yüksek sayıda parametre ile çalışmak yüksek miktarda işlem gücü gerektirir. Havuzlama katmanı bu noktada parametre sayısını makul seviyelere çekmek amacı ile kullanılır. Havuzlama katmanları genel olarak konvolüsyonel katmanlar arasına periyodik olarak eklenir. Havuzlama Katmanı, girişin her derinlik diliminde bağımsız olarak çalışır. Genellikle maksimum ve ortalama olmak üzere iki çeşit yöntem söz konusudur. En yaygın biçim, girişteki her derinlik diliminde hem genişlik hem de yükseklik boyunca ikilik bir adımla uygulanan 2x2 boyutundaki filtrelere sahip bir havuzlama katmanıdır. Böyle bir havuzlama işlemi sonucunda katmanın boyutu %75 oranında atılacaktır. Havuzlama işlemi sonunda derinlik boyutu değişmeden kalır.

Havuzlama katmanı için filtre boyutu ve adım olmak üzere iki adet hiperparametre söz konusudur. Filtre boyutu genellikle 2x2 olarak seçilir. Adım hiperparametresi ise tıpkı konvolüsyonel katmandaki adım parametresi gibi çalışır. Filtrenin giriş hacmi boyunca kaçar adım aralıklarla kaydırılacağını belirler. Bu parametrelere göre çıkış hacminin bir boyutunu hesaplayan fonksiyon şu şekilde gösterilebilir:

$$Ç = (G-F) / A+1 \tag{3.12}$$

Denklem 3.12'de Ç çıkış hacminin bir boyutu, G giriş hacminin bir boyutu, F filtre boyutu, A ise adım değeridir.

Genel olarak filtre boyutu 3x3, adım sayısı iki olarak seçilen bir varyasyon ve filtre boyutu 2x2, adım sayısı iki olarak seçilen ve daha yaygın olarak kullanılan diğer bir varyasyon olmak üzere iki tür havuzlama tekniği söz konusudur. Havuzlama katmanı için 3x3'ten daha büyük boyuta sahip filtreler pek kullanılmaz. Bunun sebebi o büyüklükteki filtrelerin çok fazla anlamsal verinin kaybolmasına sebep olmasıdır.



Şekil 3.23. Havuzlama Katmanının Çalışma şekli[42]

Şekil 3.23'de görüldüğü gibi havuzlama katmanındaki filtreler giriş hacmi boyunca kaydırılır ve her bir adımda bir çıktı üretir. Yukarıdaki şekilde filtre boyutu iki, adım sayısı iki ve havuzlama işlemi olarak maksimum seçilmiştir. Havuzlama işlemi giriş hacminin derinliğini değiştirmez.

Havuzlama katmanı her ne kadar ağdaki parametre sayısını azaltsa da görüntüdeki anlamsal bilginin kaybolmasına da sebep olmaktadır. Tek bir basit havuzlama işleminde bile giriş hacminin %75'i atılır. Dolayısı ile bu alanda yapılan çalışmalarda havuzlama katmanlarına çok sıcak bakılmaz. Havuzlama katmanı kullanmak yerine evrişimli katmanların daha büyük adımlarla kullanılarak giriş hacminin boyutunun yani parametre sayısının azaltılabileceği düşünülmektedir.

Aktivasyon Katmanı: Konvolüsyonel ağlarda aktivasyon katmanlarında ReLU fonksiyonu kullanılır. Aktivasyon katmanı aynı zamanda ReLU katmanı olarak adlandırılır. Aktivasyon veya ReLU katmanı, sıfır eşik değeri olmak üzere maksimum (0, $+\infty$) aralığında element bazlı bir aktivasyon işlevi uygulayacaktır. Aktivasyon katmanları genellikle konvolüsyonel katmanı takip eder. Bu katmanda ReLU fonksiyonunun tercih edilme sebebi sinir ağlarına benzer şekilde bu fonksiyonun, sigmoid ve hiperbolik tanjant fonksiyonlarına kıyasla daha az maliyetli olmasıdır.

Tam Bağlı(Fully Connected) Katman: Sıradan sinir ağlarında olduğu gibi ve adından da anlaşılacağı gibi, bu katmandaki her nöron bir önceki katmandaki tüm nöronlara bağlanır. Konvolüsyonel ağların en sonunda yer alır ve sınıflama yapmak için kullanılırlar. Yine sinir ağlarında olduğu gibi bu katmanlar içinde önyargı değerleri söz konusudur. Bu katmanlarda genellikle softmax fonksiyonları kullanılır.

Bir konvolüsyonel ağ piksellerden oluşan giriş görüntüsünü yukarıda açıklanan katmanlar ile işleyerek çıkışta sınıf puanları elde eder. Bu noktada konvolüsyonel katman ve tam bağlı katman tıpkı yapay sinir ağlarında olduğu gibi ağırlık ve önyargı değerleri içerir. Bu değerler yapay sinir ağlarında olduğu gibi eğitilir. Aktivasyon ve havuzlama katmanları ise sabit işlemler gerçekleştirir ve bu katmanlarda herhangi bir eğitim gerçekleşmez.

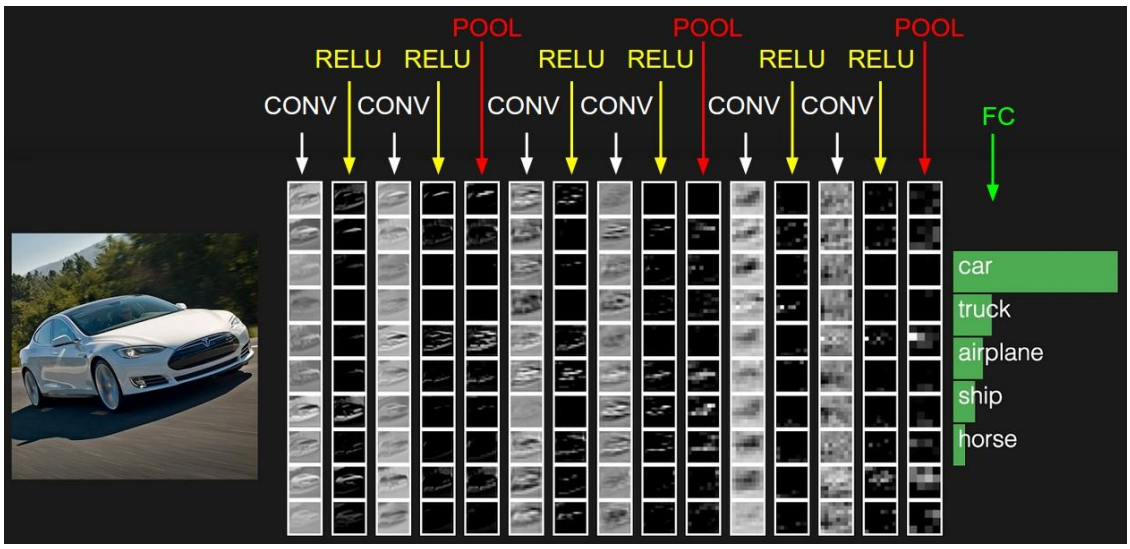
3.4.2. Konvolüsyonel Ağlarda Özellik Çıkarımı

Bir özellik haritası veya aktivasyon haritası, belirli bir filtrenin çıkış aktivasyonları olarak değerlendirilebilir.

Her bir filtre görüntü üzerinde gezdirilirken her bir adımda bir değer üretir ve bu değere bir önyargı değeri eklenerek oluşan yeni değer bir aktivasyon fonksiyonuna

gönderilir. Bir filtre katmanı için tüm aktivasyon fonksiyonlarının çıktısı olan matrise özellik haritası (Feature Map) denir. Böylece o filtre katmanı ile ana görüntü üzerinden özellikler tespit edilmiş olur. Görüntü üzerinde birden çok özellik bulunduğu için bu özellikleri tespit edebilmek adına birden fazla filtre kullanılır. Her bir filtre görüntüdeki farklı özellikleri ortaya çıkarır.

Özellik haritası ve etkinleştirme haritası tam olarak aynı şeyi ifade eder. Aktivasyon haritası olarak adlandırılır. Bunun sebebi görüntünün farklı kısımlarının aktivasyonuna karşılık gelen bir haritalama ve aynı zamanda bir özellik haritası olmasıdır. Aynı zamanda görüntüde belirli bir özelliğin bulunduğu yerin bir haritasıdır. Şekil 3.24'de bir özellik haritası ve çıkış sınıflandırması görülmektedir.



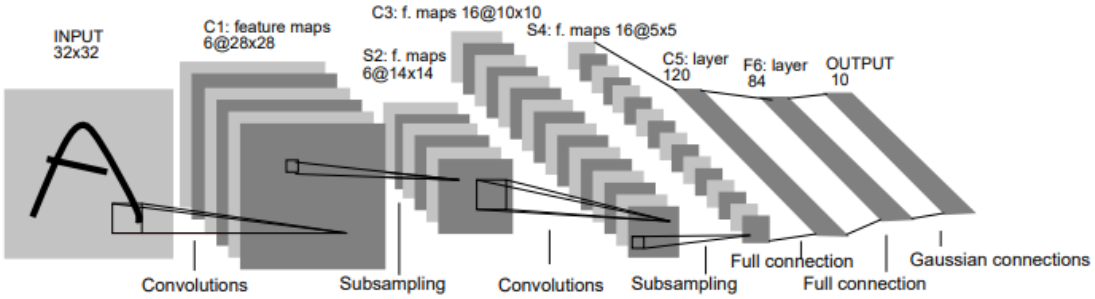
Şekil 3.24. Özellik Haritası Ve Çıkış Sınıflandırması[42]

3.4.3. Yaygın Konvolüsyonel Ağ Mimarileri

Konvolüsyonel ağlar alanında daha önce birçok mimari ortaya konulmuş ve bunların bazıları özellikle görüntü sınıflandırma problemlerinde oldukça öne çıkmıştır. Öne çıkan mimarilerden bazılarına aşağıda değinilmiştir.

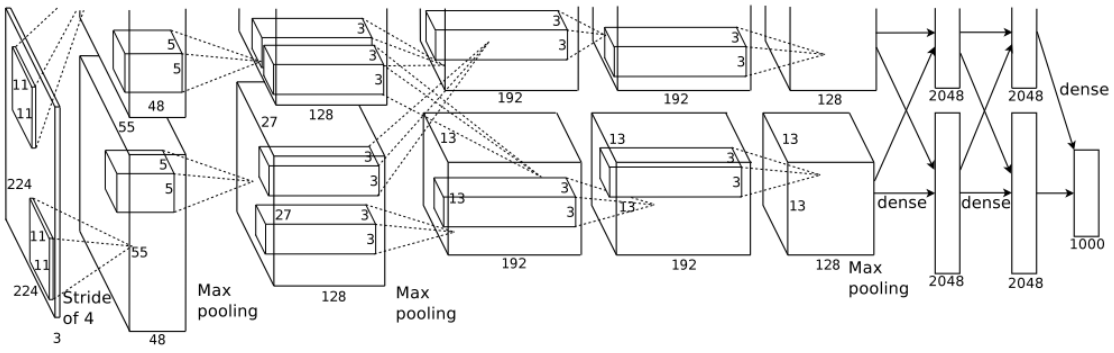
LeNet-5: Konvolüsyonel ağların ilk başarılı uygulamaları 1990'larda Yann LeCun tarafından geliştirilmiştir. Bunlardan en bilineni, posta kodlarını, rakamları ve benzeri görüntüleri okumak için kullanılan LeNet mimarisidir. Yann LeCun, "Gradient-Based

Learning Applied to Document Recognition" başlıklı çalışmasında el yazısı karakter tanımaya uygulanan çeşitli yöntemleri gözden geçirmekte ve bunları standart bir el yazısı rakam tanıma görevinde karşılaştırmaktadır [41].



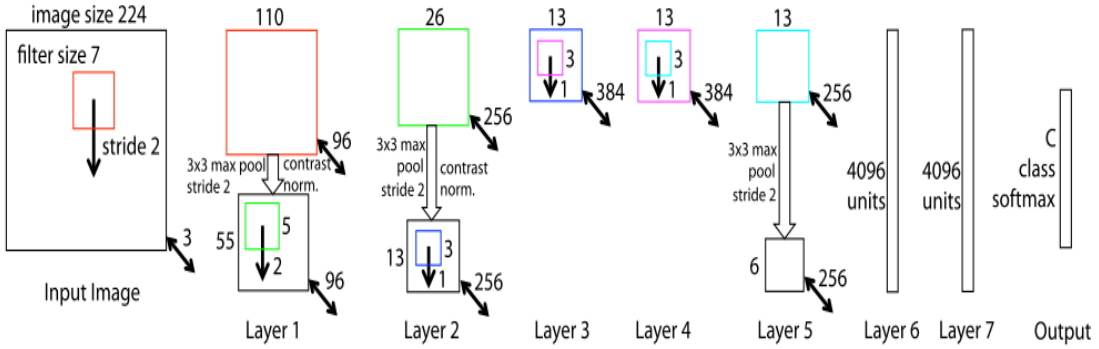
Şekil 3.25. Lenet-5 Mimarisi[41]

AlexNet: Evrişimli Ağları popüler hale getiren ilk çalışma, Alex Krizhevsky, Ilya Sutskever ve Geoff Hinton tarafından geliştirilen AlexNet oldu. AlexNet, 2012'de ImageNet ILSVRC yarışmasında sunulmuş ve önemli ölçüde iyi performans göstermiştir. Bu ağ, LeNet mimarisine çok benzer bir mimariye sahiptir. Ancak daha derin, daha büyük ve üst üste yığılmış evrişimli katmanlar içermiştir. Geliştirilen bu ağ 60 milyon parametreye ve 650.000 nörona sahiptir. Beş evrişimli katmandan oluşmuş, bunlardan bazılarını maksimum havuzlama katmanları izlemiş ve ağın son bölümü 1000 nöronlu softmax ile tam bağlantılı üç katmandan oluşturulmuştur. Eğitimi hızlandırmak için, aktivasyon katmanlarında ReLU fonksiyonları ve evrişim işlemini çok verimli bir şekilde gerçekleştiren GPU uygulaması kullanımıştır.



Şekil 3.26. Alexnet Mimarisi[27]

ZF-Net: ImageNet ILSVRC 2013 yarışmasını kazanan mimaridir. Matthew Zeiler ve Rob Fergus tarafından geliştirilen bir evrişimli ağ olan ZFNet (Zeiler & Fergus Net'in kısaltması) olarak tanınmıştır. Özellikle ağın ortasında bulunan konvolüsyonel katmanların boyutunu genişleterek ve ilk katmandaki adım ve filtre boyutunu küçülterek mimari hiper parametrelerinde ince ayar yapmışlardır. AlexNet'in geliştirilmiş bir versiyonu olarak değerlendirilebilir. Geliştirdikleri modelin, en sonunda bulunan softmax katmanlarının yeniden eğitilmesiyle diğer datasetler içinde genelleştirilebileceğini ortaya koymuşlardır. Yeniden eğittikleri softmax katmanları ile modellerinin, Caltech-101 ve Caltech-256 datasetleri üzerinde başarılı sonuçlar verdiğini göstermişlerdir.

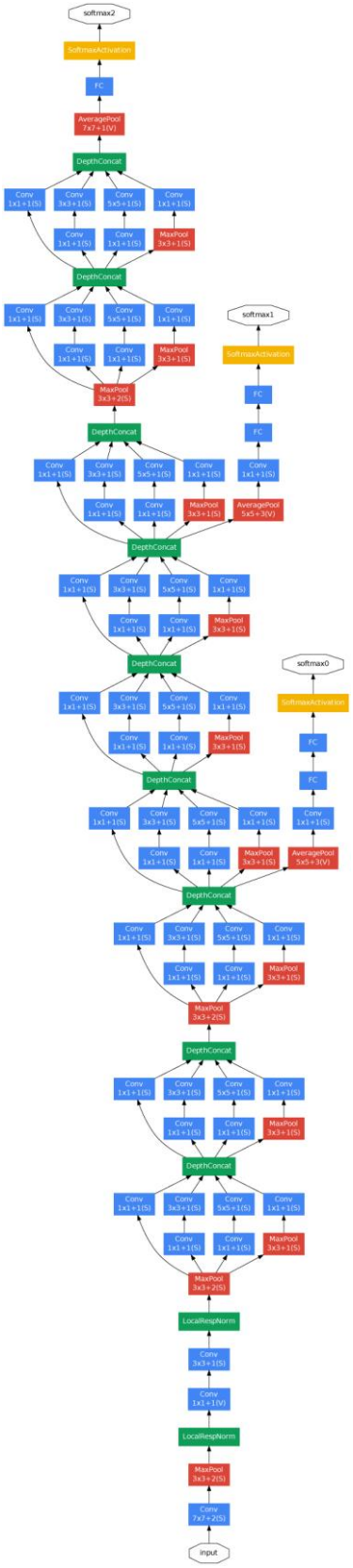


Şekil 3.27. Zf-Net Mimarisi[43]

GoogleLeNet: Google firmasında görev yapan Szegedy ve arkadaşları tarafından geliştirilen bir modeldir. "Going deeper with convolutions" [44] isimli çalışmada ortaya konulmuştur. ImageNet ILSVRC 2014 yarışmasını kazanmıştır.

Szegedy ve ekibi dikkatlice hazırlanmış bir tasarımla, hesaplama bütçesini sabit tutarken ağın derinliğini ve genişliğini arttırmışlardır. Ana katkısı, ağdaki parametre sayısını önemli ölçüde azaltan bir başlangıç modülünün geliştirilmesidir. AlexNet'te kullanılan 60 milyon parametreye göre sadece 4 milyon parametre ile parametre sayısını önemli ölçüde azaltmıştır.

Bu çalışmada konvolüsyonel ağın en sonundaki tam bağlantılı katmanlar yerine ortalama havuzlama kullanılmış ve çok önemli görünmeyen çok sayıda parametre ortadan kaldırılmıştır Ayrıca, GoogLeNet mimarisinin en son Inception-v4 olmak üzere çeşitli takip sürümleri de bulunmaktadır.

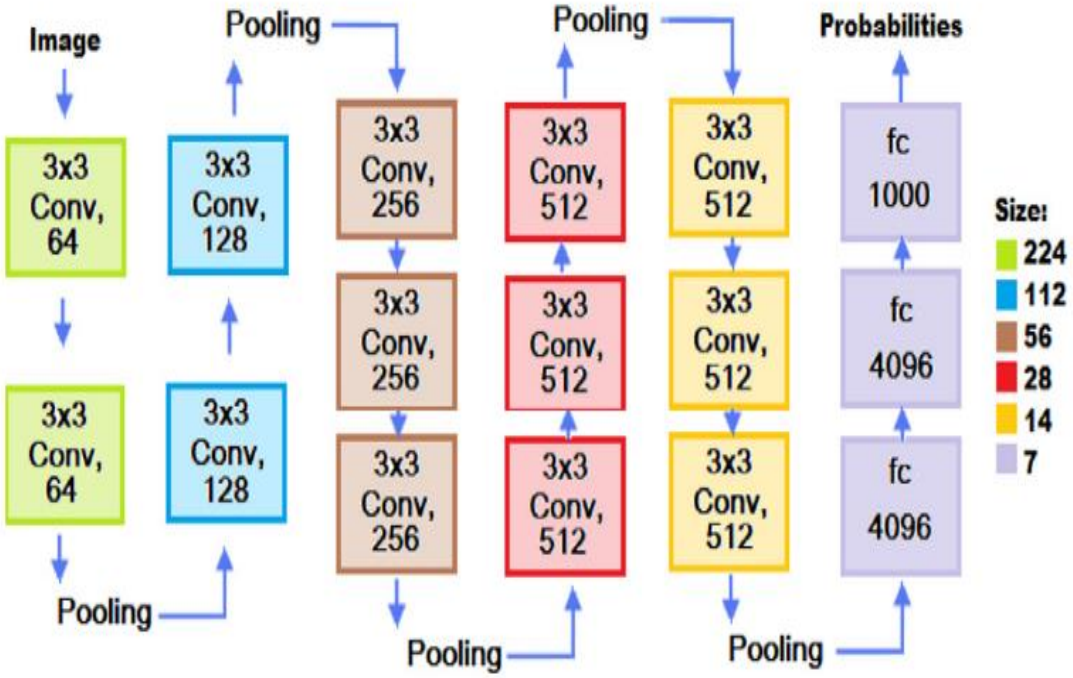


Şekil 3.28. Googlenet Mimarisi[44]

VGGNet: ILSVRC 2014 yarışmasında ikinci olan bu model Karen Simonyan ve Andrew Zisserman tarafından geliştirilmiştir.

Ana katkıları, çok küçük (3x3) evrişim filtrelerine sahip bir mimari kullanarak derinliği artan ağların kapsamlı bir değerlendirmesidir. Derinliğin 16 ila 19 katmana çıkarılmasıyla önceki modellere göre önemli bir iyileştirmenin elde edilebileceğini göstermişlerdir [45].

Geliştirilen model 16 konvolüsyonel ve tam bağlantılı katmanı içerir ve baştan sona yalnızca 3x3 evrişim ve 2x2 havuzlama gerçekleştiren son derece homojen bir mimariye sahiptir. VGGNet'in bir dezavantajı, değerlendirmenin daha maliyetli olması ve çok daha fazla bellek ve parametre (140M) kullanmasıdır. Bu parametrelerin çoğu, ilk tam bağlantılı katmandadır. Daha sonraları, bu tam bağlantılı katmanların performans düşüşü olmadan kaldırılabileceği ve gerekli parametrelerin sayısını önemli ölçüde azalttığı bulunmuştur. Şekil 3.29 onaltı evrişim katmanlı bir VGGNet mimarisini göstermektedir.

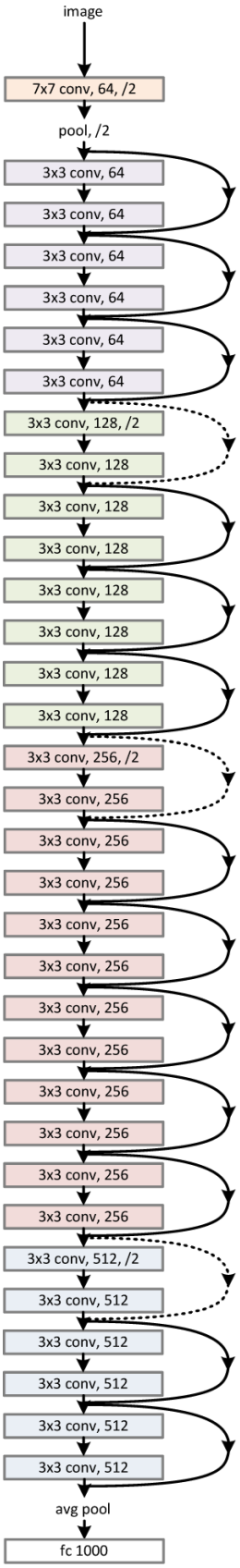


Şekil 3.29. Vgg-16 Katman Yapısı[46]

ResNet: Kaiming He ve arkadaşları tarafından geliştirilen ResNet. ILSVRC 2015 yarışmasının galibi olmuştur. Özel atlama bağlantılarına ve yoğun bir toplu normalleştirme kullanımına sahiptir. Mimaride ayrıca ağın sonunda tamamen bağlantılı katmanlar eksiktir.

ResNet'in optimize edilmesinin daha kolay olduğunu ve önemli ölçüde artan derinlikten doğruluk elde edebileceğini gösteren kapsamlı deneysel kanıtlar sağlamışlardır. ImageNet veri setinde artık ağları 152 katmana kadar derinlikte değerlendirmişlerdir. Ağın, VGG ağlarından 8 kat daha derin ancak yine de karmaşıklığının daha düşük olduğunu göstermişlerdir. Bu artık ağlardan oluşan bir model, ImageNet test setinde % 3,57 hata değeri elde etmiştir [47].

ResNet varyasyonları şu anda en gelişmiş konvolüsyonel sinir ağı modelleridir ve uygulamada kullanılmak için varsayılan seçimdir. Özellikle, Kaiming He ve diğerleri tarafından orijinal ResNet mimarisi üzerine geliştiren başka mimarilerde söz konusudur. Bu çalışmalardan biri yine aynı kişiler tarafından yapılan "Identity Mappings in Deep Residual Networks" [48] başlıkla çalışmadır. Bu çalışmada Kaiming He ve arkadaşları, atlamalı bağlantılar ve ekleme sonrası aktivasyon olarak kimlik eşleştirmeleri kullanıldığında ileri ve geri sinyallerin doğrudan bir bloktan başka bir bloğa yayılabileceğini öne süren kalıntı yapı bloklarının arkasındaki yayılma formülasyonlarını analiz etmişlerdir.



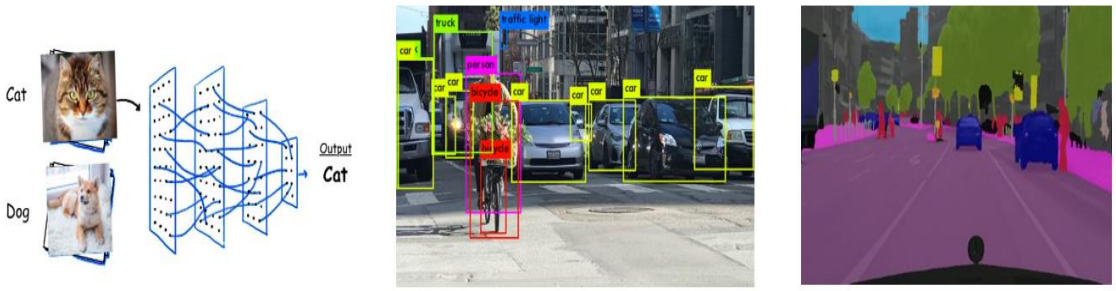
Şekil 3.30. Resnet Mimarisi[47]

3.5. Görüntü Bölümleme ve Semantik Segmentasyon

Görüntü bölümleme, bilgisayar bilimlerinde uzun süredir üzerinde çalışılan bir alandır. Derin öğrenme modelleri ortaya çıkmadan önce bu görevi çözmek için makine öğrenmesinde watershed algoritması, görüntü eşikleme, K-ortalamalı kümeleme, grafik bölümleme yöntemleri, SVM ve bunların benzeri pek çok algoritma tasarlanmıştır.

Derin öğrenme modelleri ortaya çıktıktan sonra görüntü sınıflandırma ve nesne algılama gibi problemlerde makine öğrenmesi algoritmalarına göre daha başarılı olmuşlardır. Özellikle derin öğrenme tabanlı konvolüsyonel ağlar bilgisayarlı görü problemlerinde büyük mesafeler alınmasına sebep olmuşlardır.

Bilgisayarlı görü ile ilgili problemler, artan zorluk sıralamasına göre görüntü sınıflandırma, nesne algılama ve görüntü bölümlemedir.



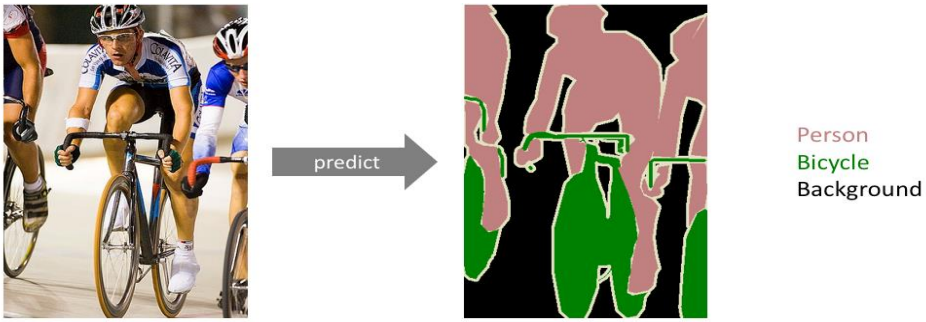
Şekil 3.31. Görüntü Sınıflandırma, Nesne Algılama, Görüntü Bölümleme

Görüntü sınıflandırma problemlerinde şekil 3.31'de solda görüldüğü gibi görüntünün tümünü etiketlendirme işlemi yapılır. Nesne tanıma problemlerinde ise görüntüde bulunan nesneleri ve bulundukları konumları etiketlendirme işlemleri yapılır. Bölümleme problemlerinde ise problem bir üst seviyeye taşınır ve görüntüde bulunan nesnelerin tam sınırlarının belirlenmesine çalışılır.

Görüntü bölümleme problemi bir görüntüdeki her pikseli bir sınıfa ait olacak şekilde sınıflandırma işlemidir. İki tür bölümleme tekniği söz konusudur.

3.5.1. Semantik Segmentasyon

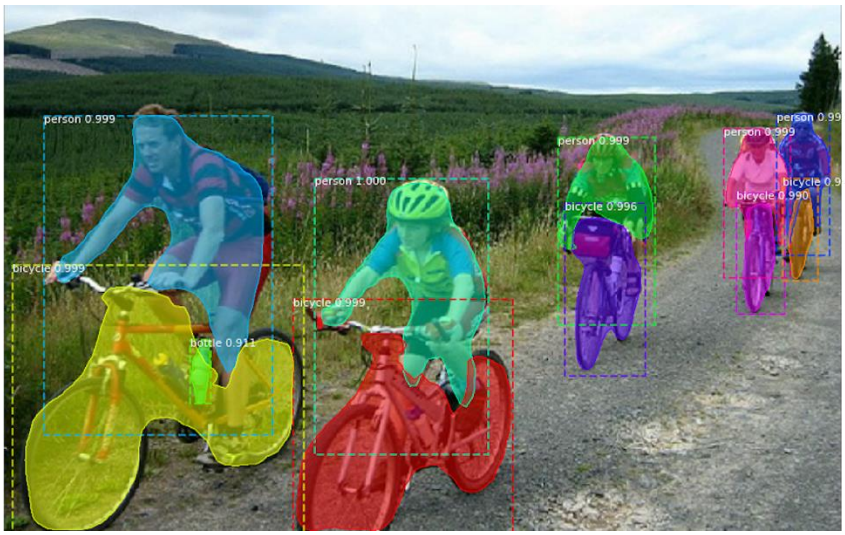
Semantik segmentasyonda görüntüdeki nesneler sınırlarının belirlenmesinin yanı sıra her birisi ait oldukları sınıflara göre sınıflandırılırlar. Örneğin bir görüntüde üç adet kedi varsa, bu üç kedi beraber tek bir sınıf oluşturur. Semantik segmentasyon görüntüdeki her pikseli bir sınıfa ait olacak şekilde sınıflandırır. Robot görüşü, otonom sürüş görevleri, tıbbi amaçlı görevler gibi çalışmalarda sıklıkla kullanılır.



Şekil 3.32. Semantik Segmentasyon

3.5.2. Örnek Bölümleme

Örnek bölümlemede görüntüdeki her nesnenin sınırlarının belirlenmesinin yanında hepsine farklı bir sınıf etiketi verilmesi esastır. Örneğin görüntüde üç adet kedi varsa bunların sınırları çizilir ve her birisine farklı bir etiket verilir.



Şekil 3.33. Örnek Bölümleme

3.5.4. Öne Çıkan Bazı Segmentasyon Yöntemleri

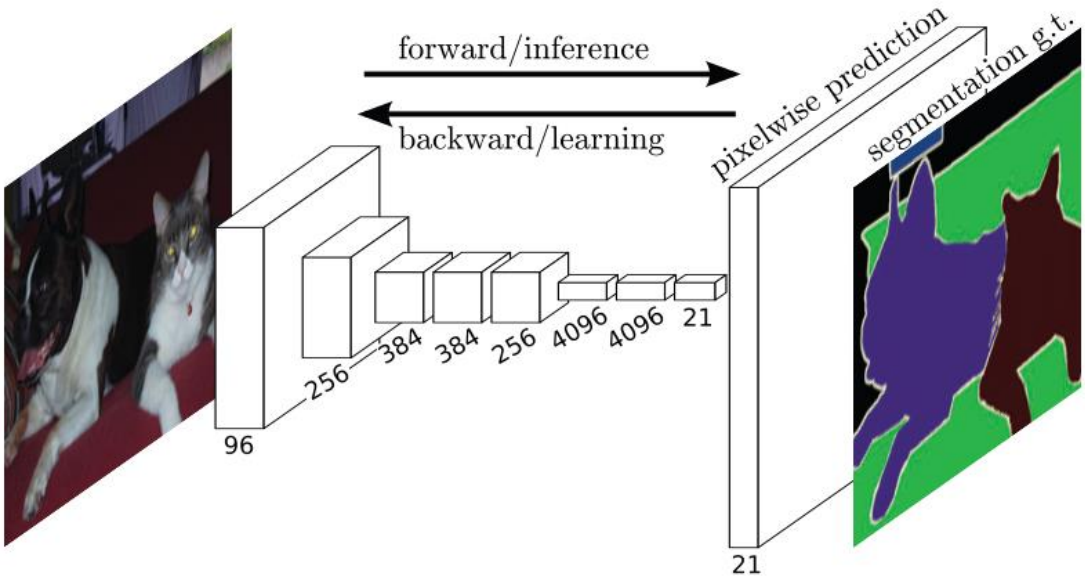
Bu bölümde popüler olan birkaç segmentasyon yönteminden derinlemesine bahsedilmiştir.

Fully Convolutional Network(Tamamen Evrişimli Ağ): Konvolüsyonel ağlar görüntü sınıflandırma, nesne tanıma ve segmentasyon görevlerinde etkin şekilde kullanılırlar. Tamamen evrişimli ağ modeli de segmentasyon problemlerini çözmek için konvolüsyonel ağların bu güçlü avantajlarından faydalanmaktır. Aynı sebeple segmentasyon problemlerini çözmeye çalışan diğer bir çok modelde tamamen evrişimli ağ modelini temel alır. Bu modeldeki temel anlayış, gelişigüzel büyüklükte girdi alan, verimli çıkarım ve öğrenmeyle uygun büyüklükte çıktı üreten "tamamen evrişimli" ağlar oluşturmaktır.

Tamamen evrişimli ağ modeli uçtan uca, piksel düzeyinde tahmin yapmak için yapılan ilk çalışmalardandır. Bu çalışma yapılana kadar konvolüsyonel ağlar genel olarak rastgele boyutlu girdi görüntülerini alıp çıkışta tamamen bağlı katmanlara dönüştüren yapılar olmuşlardır. Bu çalışmada sıradan konvolüsyonel ağlara ek olarak yukarı örnekleme katmanları, alt örneklemeli havuzlama ve piksel tabanlı tahmin özellikleri ortaya konmuş ve konvolüsyonel ağlar segmentasyon problemlerini çözmek için kullanılmaya başlanmıştır.

Bu model hem asimptotik hemde mutlak olarak etkili olmasının yanında nispeten basit mimarisi nedeniyle diğer birçok modeldeki karmaşık geliştirmelere olan ihtiyacı ortadan kaldırır. Modelde parça tabanlı eğitim(patch-wise training) yaygın olarak kullanılır. Parça tabanlı eğitiminde CNN, görüntünün tamamı yerine görüntünün küçük bir parçası üzerinde eğitilir. Örneğin, girdi görüntüsü 4000 x 4000 piksel boyutundaysa, üzerinde bir CNN eğitmek zor olacaktır. Görüntünün ölçeği küçültülebilir, ancak bu durumda önemli bilgiler kaybedilir. Bunun yerine, parça tabanlı semantik segmentasyon gerçekeştirilirse, model görüntüden, örneğin 250x250 piksel boyutundaki parçaları alır ve bu parçalar üzerinde sanki ayrı görüntülermiş gibi eğitim gerçekleştirir. Bu, görüntünün tümünü bir bütün olarak eğitmeye göre nispeten yavaş olsa da daha yüksek bir doğruluk değeri elde edilmesini sağlar.

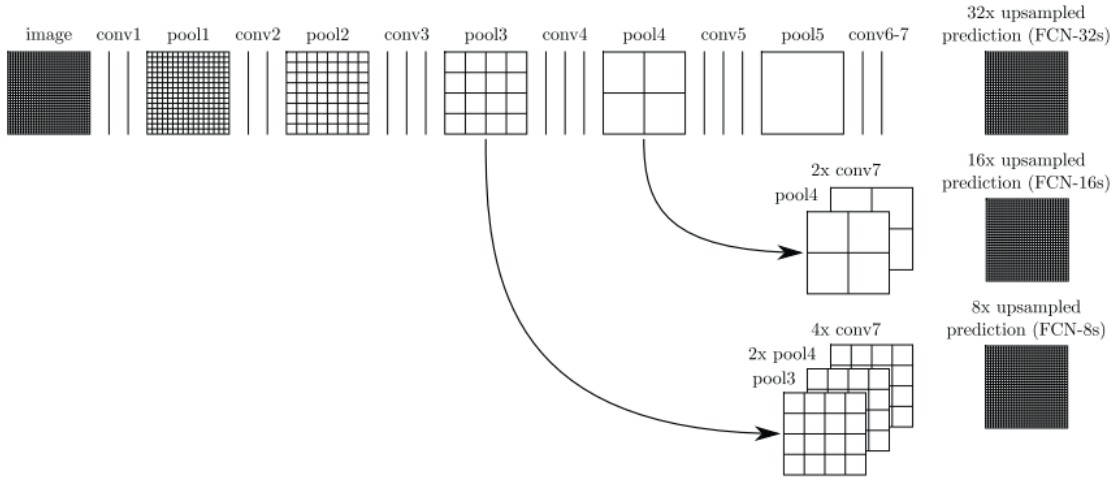
FCN temel olarak bir konvolüsyonel ağa çok benzer. Temel fark, sıradan bir konvolüsyonel ağın son bölümündeki tamamen bağlı katmanlar yerine bir takım üst örnekleme ve ters evrişim işlemleri uygulayarak görüntüde anlamsal bölümleme gerçekleştirmesidir. Bu sayede tamamen evrişimli ağlar, semantik bölümleme gibi piksel başına görevler için yoğun tahminler yapmayı verimli bir şekilde gerçekleştirebilir.



Şekil 3.34. Tamamen Evrişimli Ağ Mimarisi[5]

Semantik segmentasyonda çözülmesi gereken iki temel problem görüntüdeki nesnelerin ne olduğu ve nerede oldukları sorularıdır. Eğitilmiş bir semantik segmentasyon modeli içerisindeki global enformasyon, görüntüdeki nesnelerin ve görüntünün ne olduğu ile ilgilenirken yerel enformasyon ise nesnelerin görüntüde nerede olduğu ve tam sınırlarının belirlenmesi ile ilgilenir. Konvolüsyonel ağların en temel özelliklerinden olan özellik haritaları doğrusal olmayan bir şekilde yerelden globale doğru, öğenin görüntüdeki konumunu ve ne olduğunu ortaya çıkarmaya çalışır. Bu noktada bir girdi görüntüsü için görüntüdeki nesnelerin ne olduğunun bilgisi, yani anlamsal bilginin nispeten kabasal bir bilgi, bu nesnelerin görüntüde nerede bulunduklarının bilgisinin ise daha hassas bir bilgi olduğu söylenebilir. Böyle bir durumda bu iki bilgi grubu arasında açık bir ilişki kurmak ve bu bilgileri doğru bir şekilde birleştirmek gerekir. FCN'lerde bu işlem için şekil 3.35'de görülen atlama mimarisi kullanılır. Bu atlama mimarisinde,

çıktının anlambilimini ve uzamsal hassasiyetini iyileştirmek için uçtan uca öğrenme sözkonusudur.



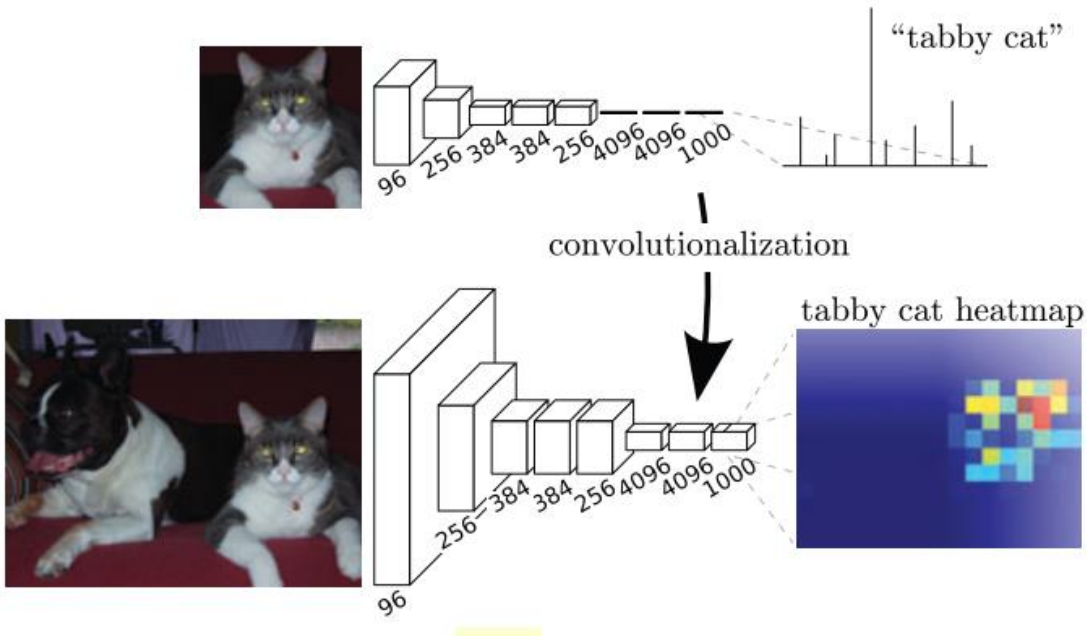
Şekil 3.35. FCN Atlama Mimarisi[5]

Tamamen evrişimli ağ modeli, o zamana kadar gerçekleştirilmiş başarılı derin öğrenme modellerinden faydalanır. Model, semantik segmentasyon görevleri için bu başarılı modelleri yeniden tasarlar ve ince ayarlarını gerçekleştirir. Bu modellere ek olarak görüntü sınıflandırmasını denetimli ön eğitim olarak kullanarak derin sınıflandırma mimarilerini uyarlar, genişletir ve tüm görüntü girdilerinden ve sonuçlarından basit ve verimli bir şekilde öğrenmek için tamamen evrişimli ince ayar(fine-tuning) yapar.

Bir konvolüsyonel ağdaki her veri katmanı, h × w × d boyutunda üç boyutlu bir dizidir; burada h ve w uzamsal boyutlardır(genişlik ve yükseklik) ve d, özellik veya kanal boyutudur(derinlik). Konvolüsyonel ağdaki ilk katman, piksel cinsinden genişliği h × w, derinliği d olan renk kanallarına sahip giriş görüntüsüdür. Daha yüksek katmanlardaki konumlar, önceki katmanda bağlı oldukları, alıcı alan olarak isimlendirilen konuma karşılık gelir [5].

Bölüm 3.4.3'de bahsettiğimiz LeNet, AlexNet VGGNet gibi tipik görüntü tanıma mimarileri, girişte sabit boyutlu girdileri alır ve uzamsal olmayan ve tamamen bağlı katmanlardan oluşan vektörler üretir. Bu ağların en sonunda yer alan tamamen birbirine

bağlı katmanlar, sabit boyutlara sahiptir ve giriş görüntülerindeki genişlik ve yükseklik gibi uzamsal koordinatları atar. Yine de bu tamamen bağlantılı katmanlar, tüm giriş bölgelerini kaplayan çekirdeklere sahip evrişimler olarak da görülebilir. Bunu yapmak, onları herhangi bir boyutta girdi ve sınıflandırma haritalarının çıktısını alan tamamen evrişimli ağlara dönüştürür.



Şekil 3.36. Bir Sınıflandırma Ağının Konvolüsyonel Ağa Dönüşümü[5]

Sınıflandırma ağlarının tamamen evrişimli olarak yeniden yorumlanması, herhangi bir boyuttaki girdiler için çıktı haritaları verirken, çıktı boyutları tipik olarak alt örnekleme ile azaltılır. Sınıflandırma ağları, filtreleri küçük ve hesaplama gereksinimlerini makul tutmak için alt örnekleme gerçekleştirir. Bu, bu ağların tamamen evrişimli bir versiyonunun çıktısının nispeten kaba bir çıktı olmasına neden olur.

Piksel bazlı tahmin için, bu kaba çıktıların tekrar piksellere bağlanması gerekir. Bunun için FCN'lerde birtakım yöntemler tanıtılmıştır. Bunlardan birincisi filtre yoğunluğunu azaltmak için Shift-and-stitch olarak adlandırılan bir işlemdir. Bu yöntemde girdinin kaydırılmış versiyonlarından elde edilen çıktıları birbirleri ile birleştirerek kaba çıktılardan yoğun tahminler elde edilebilir. Bu yöntem modelde denenmiş fakat etkin ve verimli bulunmadığından nihai modelde kullanılmamıştır. Bunun yerine nihai modelde,

aşağıda açıklayacağımız üst örnekleme yoluyla öğrenme yöntemi kullanılmıştır. Üst örnekleme yoluyla öğrenmenin, özellikle şekil 3.35'de görülen atlama bağlantılarıyla beraber kullanıldığında oldukça etkin ve verimli olduğu görülmüştür.

Üst örnekleme temel olarak geriye doğru adım adım evrişim olarak tanımlanabilir. Geriye doğru evrişim veya ters evrişim olarak da isimlendirilebilir. Üst örneklemede kaba çıktıları yoğun piksellere bağlamanın başka bir yolu da enterpolasyondur. Örneğin, basit çift doğrusal enterpolasyon, her y_{ij} çıktısını en yakın dört girişten yalnızca giriş ve çıkış hücrelerinin göreceli konumlarına bağlı olan doğrusal bir harita ile hesaplar. Üst örnekleme nihayetinde bir evrişim işlemi olduğu için sinir ağlarında öğrenmeyi gerçekleştiren geriyayılma algoritması ile model içinde uçtan uca piksel düzeyinde öğrenme için kullanılır.

Üst örnekleme gerçekleştirmek için kullanılan filtrelerde, tıpkı altörneklemede evrişim işlemi için kullanılan filtreler gibi öğrenebilir. Hatta ters evrişim katmanlarında kullanılan bu filtreler derin ağlardaki gibi doğrusal olmayan bir biçimde bile öğrenebilir.

Yapılan denemelerde, üst örneklemenin yoğun tahmini öğrenmek için hızlı ve etkili olduğu bulunmuştur. Nihai FCN mimarisinde iyileştirilmiş tahmin için bu üstörnekleme katmanları kullanılmıştır.

FCN mimarisi geliştirilirken ImageNet yarışmalarında öne çıkan modeller temel mimari olarak ele alınmış ve FCN mimarisi bu modeller üzerine geliştirilmiştir. ImageNet 2012'nin kazanan sınıflandırma mimarisi AlexNet, ImageNet 2014'de üstün başarı gösteren VGGNet ve GoogLeNet mimarileri bu kapsamda ele alınmıştır. VGGNet mimarisinde 19 katmana eşdeğer olan VGG16 mimarisi seçilmiştir. GoogLeNet mimarisinde ise performansı iyileştirmek için son ortalama havuzlama katmanı atılmış, sadece son kayıp katmanı kullanılmıştır. Ağın sonunda yer alan sınıflandırma katmanı da atılmış ve tamamen bağlı katmanlar evrişim katmanlarına dönüştürülmüştür. PASCAL veri setindeki arkaplan dahil 21 adet sınıfın tahmini için kanal boyutu 21 olan 1x1 evrişim eklenmiştir. Devamında kabasal çıktıları yoğun piksel çıktılarına dönüştürmek için üstörnekleme gerçekleştiren bir ters evrişim katmanı eklenmiştir. Tablo 3.1'de tüm bu temel mimarilerin sabit bir öğrenme oranıyla ve yeterli sayıda eğitim adımıyla eğitilmiş hallerinin ön doğrulama sonuçları karşılaştırılmıştır. Bu karşılaştırma PASCAL VOC

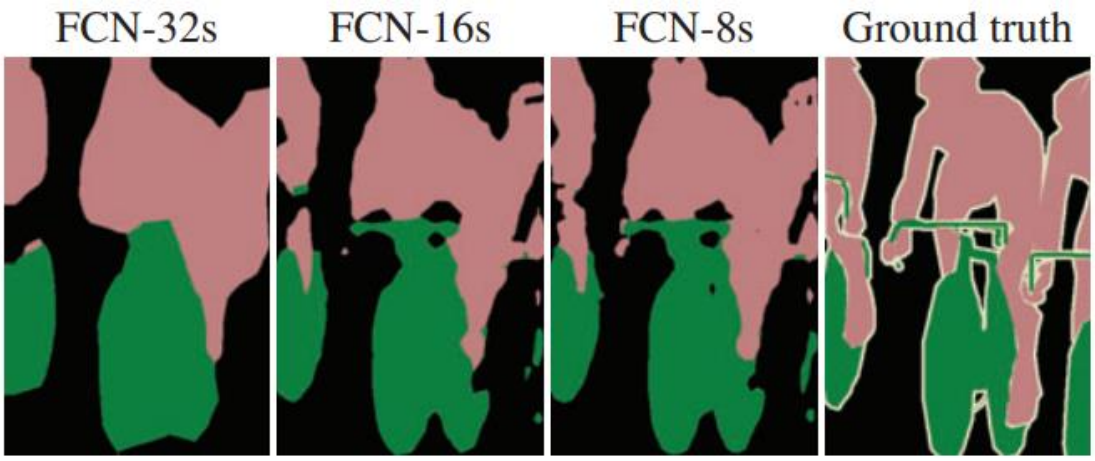
2011 validasyon seti üzerinde birleşme bölgelerinde ortalama kesişim(mean IU) ve çıkarım sürelerine(forward time) göre gerçekleştirilmiştir. Aynı zamanda tabloda ağın mimarisi ile ilgili olarak, evrişim katmanı sayısı(Conv layers), parametre sayısı(parameters), alıcı alan büyüklüğü(rf size) ve ağdaki en büyük adım sayısı(Max stride) gibi temel bilgilerde bulunmaktadır.

Tablo 3.1. FCN Mimarisinde Kullanılan Temel Mimarilerin Başarım Karşılaştırması

	FCN-AlexNet	FCN-VGG16	FCN-GoogLeNet
Mean IU	39.8	56.0	42.5
Forward time	50 ms	210 ms	59 ms
Conv. layers	8	16	22
Parameters	57 M	145 M	6 M
Rf size	355	404	907
Max stride	32	32	32

Tablo 3.1'e bakıldığında her ne kadar çıkarım süresi ve gerekli parametre sayısı fazla olsada 56.0 mean IU skoruyla FCN-VGG16 mimarisinin diğer modellere göre daha başarılı olduğu görülür.

Tamamen evrişimli sınıflandırma mimarileri temel alınarak geliştirilen FCN mimarisi segmentasyon problemlerinde kullanılmak için henüz tam hazır değildir. Çıktıları, şekil 3.37'de görüldüğü gibi nispeten kaba ve segmentasyon görevlerinde kullanılmak için yeteri kadar tatmin edici hale gelmemiştir. Bunun sebebi son katmanda gerçekleştirilen 32 piksel adımlı işlemdir. 32 piksel adım işlemi çıktıdaki ayrıntı ölçeğini sınırlar.



Şekil 3.37. Farklı Adım Sayılarına Sahip Fcn Mimarilerinin Çıktıları[5]

FNC mimarisinde bu problemi gidermek için son tahmin katmanını, adım sayısı düşük olan ilk katmanlarla birleştirme yoluna gidilir. Bu yöntem FCN mimarisini, alt katmanlardan daha yüksek katmanlara atlayan kenarlara sahip bir çizgi topolojisini, bir DAG(Directed acyclic graph) ağına dönüştürür. DAG ağı, yönlendirilmiş, döngüsel olmayan, grafik olarak düzenlenmiş katmanlara sahip, derin öğrenme için bir sinir ağıdır. Bir DAG ağı, katmanların birden çok katmandan girdilere ve birden çok katmana çıktılara sahip olduğu daha karmaşık bir mimariye sahip olabilir. FCN mimarisi alt katmanlardan üst katmanlara gerçekleştirilen bu bağlantılar vasıtasıyla bir DAG ağı haline gelir. İlk katmanları ve son katmanları birleştirmek, modelin girdinin bütününe sadık olarak yerel tahminler yapmasına olanak tanır. FCN mimarisinde doğrusal olmayan bu özellik hiyerarşisi deep jet olarak adlandırılır.

Konvolüsyonel ağ mimarilerinde fazla havuzlama katmanı kullanılması girdideki anlamsal bilginin kaybolmasına neden olur. Segmentasyon görevlerinde çok daha hassas bir çıkarım beklenir. Hassas bir çıkarım girdideki anlamsal bilginin mümkün olduğu kadar korunması ile mümkün olur. FCN mimarisinde bu anlamsal bilgiyi mümkün olduğu kadar muhafaza etmek için şekil 3.35'de görüldüğü gibi katmanlar arasında atlama bağlantıları kullanılır. Bu atlama bağlantıları ilk katmanlardan son katmanlara doğru gerçekleşir. Bu bağlantılar ağın başlangıç tarafına ne kadar yakın bölümünden başlarsa doğal olarak o kadar fazla anlamsal bilgi korunur. Şekil 3.35'de en altta görülen FCN-8s mimarisinde atlama bağlantısı 3 numaraları havuzlama katmanından ağın son bölümüne

gerçekleştirilmiştir. Bir konvolüsyonel ağda 2x2 boyutunda bir havuzlama işlemi görüntüdeki piksel sayısını yatay ve dikeyde yarı yarıya düşürür. Bu işlem üç defa gerçekleştirilirse görüntüdeki piksel sayısını toplamda yatay ve dikey olarak sekiz kat azaltır. Çıktıdaki segmentasyon haritasının giriş görüntüsü ile aynı çözünürlükte olması gerektiğinden sekiz kat azalmış çözünürlüğün üstörnekleme yoluyla 8 kat büyütülmesi gerekir. Segmentasyon çıktısının daha başarılı olmasını sağlamak için bu üstörnekleme işleminin de sınırlı tutulması gerekir. Tablo 3.2'de VGG16 sınıflandırma mimarisini temel olarak kullanan FCN-32s, FCN-16s ve FCN-8s mimarilerinin performans ve doğruluk karşılaştırılmaları görülmektedir. Bu tabloda en üstte yer alan FCN-32s-fixed mimarisi dışındaki mimariler uçtan uca eğitim ile eğitilmişler, FCN-32s-fixed modeli ise uçtan uca eğitildikten sonra son katmanı için ayrıca bir ince ayar eğitimi ile eğitilmiştir.

Tablo 3.2. Farklı Adımlı FCN Mimarilerinin Performans Ve Doğruluk Karşılaştırması

	Pixel acc	Mean acc.	Mean IU	F.W. IU
FCN-32s-fixed	83.0	59.7	45.4	72.0
FCN-32s	89.1	73.3	59.4	81.4
FCN-16s	90.0	75.7	62.4	83.0
FCN-8s	90.3	75.9	62.7	83.2

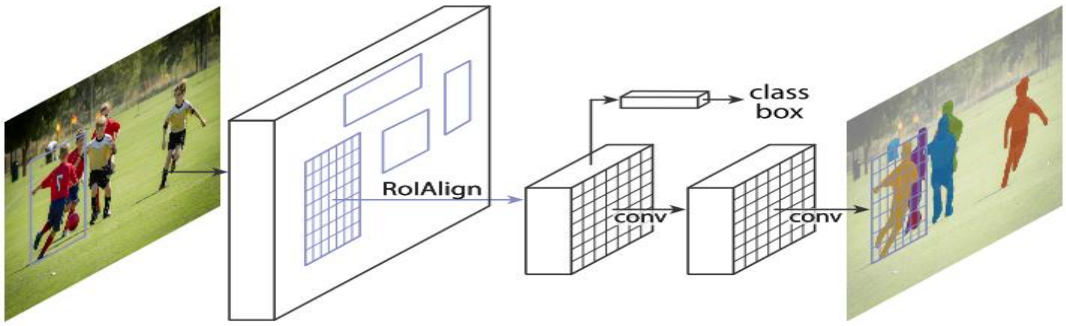
Modelin eğitilme aşamasında ise momentum ile beraber skotastik dereceli alçalma yöntemi kullanılmıştır. 20 resimden oluşan mini-parti büyüklüğü ve 10^{-3}, 10^{-4}, 5^{-5} öğrenme değerleri ile beraber sırasıyla FCN-AlexNet, FCN-VGG16, ve FCN-GoogLeNet mimarileri eğitilmiştir.

Sonuç olarak segmentasyon görevleri için geliştirilen tamamen evrişimli ağlar, klasik sınıflandırma ağlarının geliştirilmiş özel versiyonları olarak kabul edilebilir. Sınıflandırma ağları segmentasyon görevleri için genişletilirken çok çözünürlüklü katman kombinasyonlarıyla mimarilerinin geliştirilmesi ve ağların öğrenme ve çıkarım hızlarının yükseltilmesi üzerinde durulmuştur.

Mask R-CNN: Bu modelin çalışma prensibi temel olarak ikiye ayrılabilir. İlk bölümde görüntüdeki nesneler tıpkı derin öğrenmedeki nesne algılama problemlerinde olduğu gibi algılanır. İkinci bölümde ise algılanan bu nesneler için yüksek kaliteli segmentasyon

maskeleri oluşturulur. Bu yöntem bir görüntüdeki nesnelerin dikdörtgen sınırlarını belirleyen Faster R-CNN [62] isimli modele ek olarak nesnenin tam sınırlarını belirleyen maskeyi oluşturur ve Faster R-CNN modelini genişletir. Mask R-CNN modelini eğitmek kolaydır ve 5 fps'de çalışan Faster R-CNN modeline yalnızca küçük bir ek yük ekler. Dahası, Mask R-CNN modelinin diğer görevlere genellenmesi kolaydır. Örneğin, aynı çerçevede insan pozlarını tahmin etmemize izin verir [63].

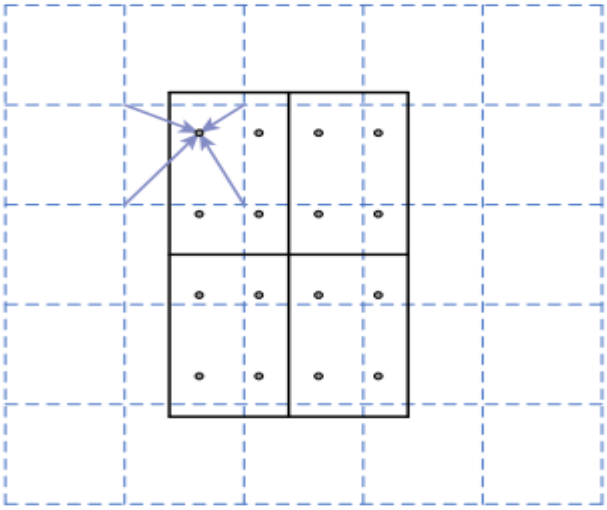
Bu yöntemde görüntüde algılanan nesnenin dikdörtgen sınırları Region of Interest (RoI) – ilgi bölgesi olarak adlandırılır ve yöntem belirnenen bu bölgede segmentasyon maskesi oluşturmaya çalışır. RoI olarak adlandırılan bu bölgede segmentasyon maskesini oluşturmak için küçük bir Fully Convolutional Network(Tamamen Evrişimli Ağ) [5] modeli kullanılır. Bu sayede yalnızca küçük bir hesaplama yükü karşılığında hızlı bir segmentasyon işlemi gerçekleştirilir.



Şekil 3.38. Mask R-CNN Yapısı[63]

Mask R-CNN modelinin üzerine geliştirildiği Faster R-CNN modeli ağ girişleri ve çıkışları arasında pikselden piksele hizalama için tasarlanmamıştır. Segmentasyon işlemlerinde giriş görüntüsünde tespit edilen her bir nesne için oluşturulacak segmentasyon maskesinin çıkış görüntüsünde de aynı konumda bulunması gerekir. Mask R-CNN modeli bu işlemi gerçekleştirmek için Şekil 3.38'de görüldüğü gibi, tam uzamsal konumları aslına sadık kalarak koruyan, RoIAlign adlı basit, niceleme içermeyen bir katman önerir. Bu katman görünüşte küçük bir işlem olmasına rağmen, maske konumunun doğruluğunu göreceli olarak %10 ila %50 oranında artırarak segmentasyon doğruluğuna büyük bir katkı sağlar. Bunun yanı sıra segmentasyon işlemlerinde bir nesnenin tam sınırlarının belirlenmesinin yanında o nesnenin ne olduğu, yani

sınıflandırılması da gerekir. Mask R-CNN modelinde her bir RoI bölgesi için nesnenin tam sınırlarını belirten bir binary maske oluşturulur. Nesnenin sınıflandırılması için ise RoI Classification(RoI sınıflandırma) denilen bir alt yöntem kullanılır. Çoğu segmentasyon modellerinde ise bu sınıflandırma piksel başına ve segmentasyon işlemi ile eşzamanlı olarak yapılır. Bu noktada Mask R-CNN modelinin sınıflandırma özelliği çoğu segmentasyon modelinden ayrılır.



Şekil 3.39. Rolalign İşlemi[63]

Şekil 3.39'da mavi kesikli çizgilerle gösterilen bölge bir özellik haritasını, siyah düz çizgilerle gösterilen bölgeler ise bir RoI alanını(bu örnekte 4 farklı RoI), noktalar ise her RoI bölgesindeki 4 örnekleme noktasını temsil eder. RoIAlign, özellik haritasındaki her bir örnekleme noktasının değerini, özellik haritasında(mavi kesikli çizgiler) içinde bulunduğu bölgenin köşe noktalarına göre çift doğrusal enterpolasyon(bilinear interpolation) kullanarak hesaplar.

Mask R-CNN modeli, eğitim sırasında giriş görüntüsünde belirlenen her bir RoI alanı için denklem 3.13'deki gibi bir kayıp fonksiyonu tanımlar.

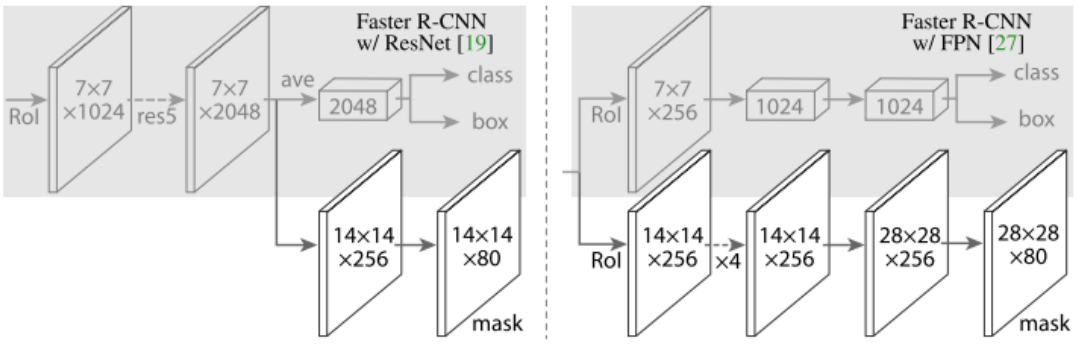
$$L = L_{cls} + L_{box} + L_{mask} \tag{3.13}$$

Burada sınıflandırma kaybı L_{cls} ve sınırlayıcı kutu kaybı L_{box}, Fast R-CNN [61] modelindeki ile aynıdır. Maskeleme işleminde ise her bir RoI alanı için m^2 boyutlu bir

çıktısı vardır ve bu, her K sınıfı için bir tane olmak üzere, m × m çözünürlüklü K ikili maskeyi kodlar. Burada piksel başına sigmoid uygulanır ve L_{mask} kayıp değeri, ortalama ikili çapraz entropi kaybı(average binary cross-entropy loss) olarak tanımlanır. Kesinleşmiş bir k sınıfı ile ilgili bir RoI alanı için L_{mask} değeri yalnızca k indis numaralı maskede tanımlanır ve diğer maske çıkışları kayba etki etmez.

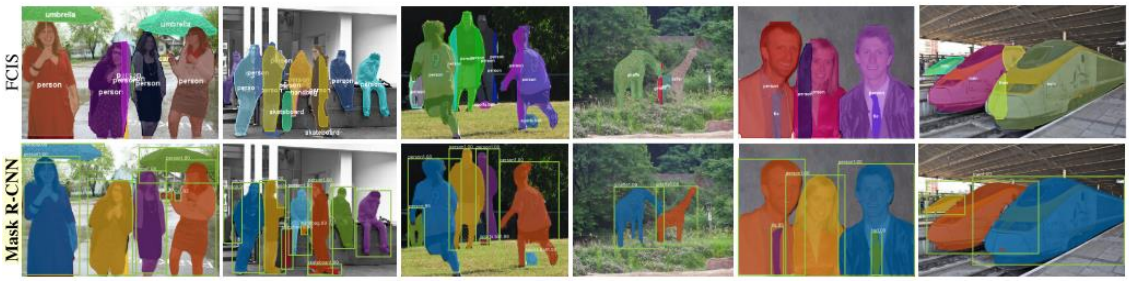
Segmentasyon maskesi görüntüdeki bir nesnesinin uzamsal düzenini kodlar. Bu nedenle, konvolüsyonel ağların en sonunda yer alan tam bağlı katmanların yaptığı gibi bir sınıf belirtmek veya nesnenin konumunu veren bir dikdörtgen çizmenin aksine maskelerin uzamsal yapısının çıkarılması, evrişimler tarafından sağlanan piksel piksel geçerlilik ile ele alınabilir.

Bu çalışma Faster R-CNN modelinin geliştirilmesiyle ortaya konulmuştur. Bu geliştirme gerçekleştirilirken başarım sonuçlarını karşılaştırmak için farklı temel mimariler kullanılmıştır. Şekil 3.40'da temel mimari olarak ResNet [72] ve FPN [64] kullanan iki farklı Faster R-CNN modelinin üzerine gerçekleştirilen genişletmeler görülmektedir.



Şekil 3.40. Resnet Ve Fpn Temelli Faster R-CNN Modellerinin Genişletilmesi[63]

Şekil 3.41'de Mask R-CNN modeli ile o zamana kadarki en başarılı modellerden biri olan FCIS+++ [73] modelinin sonuçları karşılaştırılmaktadır. FCIS+++, üst üste binen örneklerde sistematik yapaylıklar sergiler ve örnek segmentasyonunun temel zorluğuyla karşı karşıya kaldığını gösterir. Maske R-CNN modelinin böyle bir bozulma göstermediği görülmüştür.

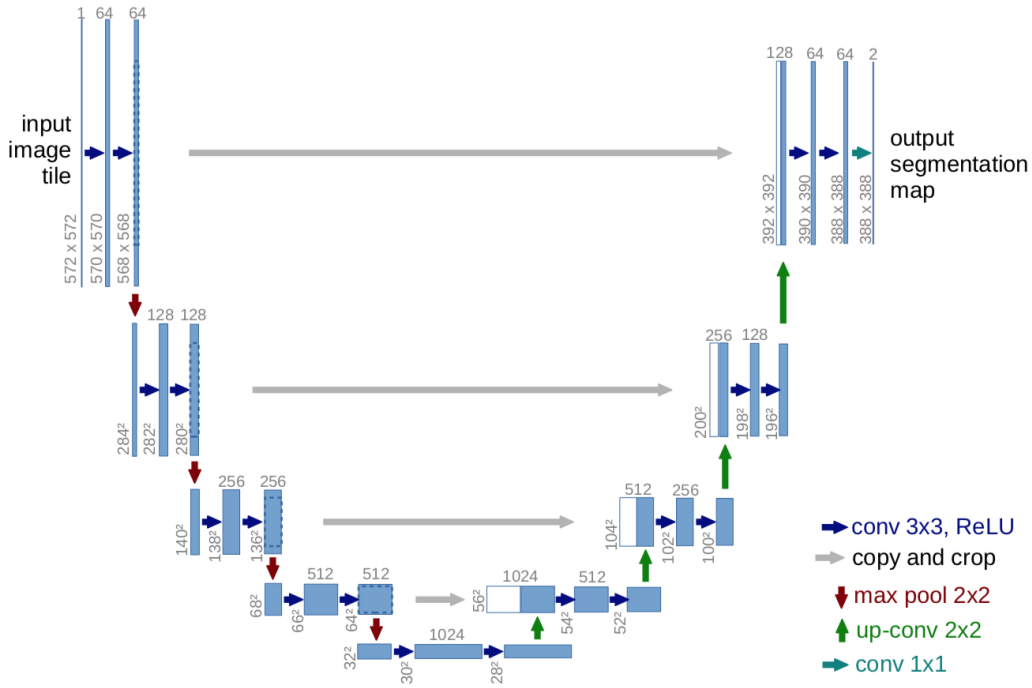


Şekil 3.41. Üstte Fcıs+++ Modelinin Altta İse Mask R-Cnn Modelinin Sonuçları[63]

U-Net: 2015 yılında biyomedikal alanındaki segmentasyon problemlerini çözmek için geliştirilen model nispeten eski bir model olmasına rağmen mimarisinin basitliği ve kullanım yaygınlığı nedeni bu tez çalışmasında incelenmiştir. Adını mimarisinin şeklinden alan U-net, evrişimsel ağ tabanlı bir modeldir. Tamamen evrişimli ağ(Fully Convolutional Network) üzerine geliştirilmiştir. Bu temel mimari, çok az eğitim görüntüsü ile çalışacak ve daha hassas segmentasyonlar sağlayacak şekilde değiştirilmiş ve genişletilmiştir.

Evrişimsel ağlar, çalışma mantıkları nedeniyle derin katmanlara inildikçe giriş görüntüsünden oldukça fazla miktarda veri kaybederler. Bu noktada U-net'in temel çözüm noktası, ilk katmanlardan son katmanlara gerçekleşen kısa yol bağlantılarıdır. Böylece son katmanlardaki veri kaybının önüne geçilmesi amaçlanmıştır.

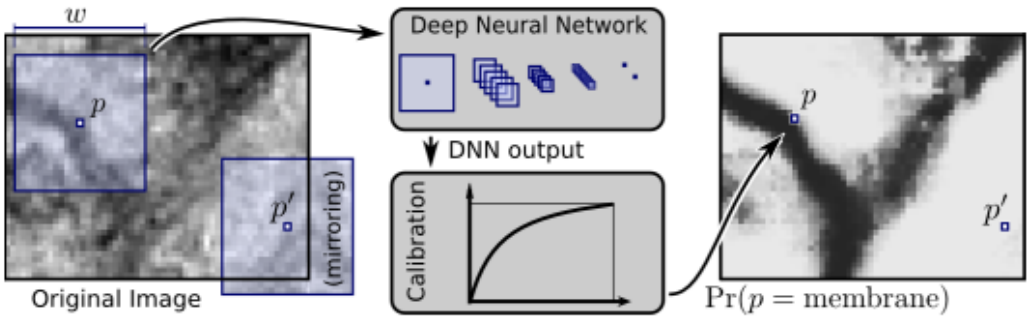
Derin ağların başarısı, eğitim için kullanılacak verisetlerinin hem niteliği hemde niceliğiyle doğrudan bağlantılıdır. U-net mimarisi geliştirilirken modelin başarımını arttırmak adına eğitim için kullanılacak verisetinin üzeirinde etkin bir veri arttırma işlemi gerçekleştirilmiştir. Mimarinin basitliği sayesinde nicelik olarak nispeten düşük seviyede bir veriseti ile de başarılı sonuçlar ortaya koyabileceği gösterilmiştir. U-net mimarisinin en baskın özelliği simetrik yapısı ve kullandığı kısayol bağlantılarıdır. Bu sayede ağın çok az görüntüden uçtan uca eğitilebileceği ve çözmeye çalıştığı problemlerde önceki yöntemden daha iyi performans gösterdiği ortaya konulmuştur. Ağ oldukça hızlı çalışmaktadır. 512x512'lik bir görüntünün bölümlenmesi, yeni bir GPU'da bir saniyeden daha kısa sürer [2].



Şekil 3.42. U-Net Mimarisi[2]

Şekil 3.42'de görülen gri renkli oklar kısa yol bağlantılarıdır. Mavi oklar evrişim, kırmızı oklar havuzlama, yeşil oklar üstörnekleme veya ters evrişim işlemlerini temsil eder. Mavi kutular çok kanallı özellik haritalarını gösterir. Kanal sayıları her bir kutunun üzerinde yazmaktadır. Kutuların solunda dikey olarak yazan sayılar genişlik ve yükseklik değerleridir. Mimarinin sağ tarafında yer alan mavi kutulara birleşik olarak görünen beyaz kutular ise mimarinin sol tarafında karşılık gelen konumdan kısayol bağlantıları vasıtasıyla kopyalanmış özellik haritalarını gösterir.

U-net mimarisinin geliştirilme amacı olan biyomedikal görevlerde binlerce eğitim görüntüsüne genellikle ulaşılamaz. Dolayısıyla benzer bir problemi çözmeye çalışan Ciresan ve ark. [74], her bir pikselin sınıf etiketini tahmin etmek için bir kayan pencere özelliği olan bir ağı eğitmişlerdir. Kayan pencere özelliği, bir pikselin etrafındaki pikselleri girdi olarak alan bir yerel bölge (yama) sağlanması ve böylece ilgili pikselin sınıf tahminini iyileştirme üzerine geliştirilen bir özelliktir.



Şekil 3.43. Cıresan Ve Ark. Tarafından Önerilen Kayan Pencere Yaklaşımı[74]

U-net mimarisindeki önemli bir özellik, yukarı örnekleme bölümünde, ağın bağlam bilgisini daha yüksek çözünürlüklü katmanlara aktarmasına izin veren çok sayıda özellik kanalına sahip olmasıdır. Şekil 3.43'de gösterilen mimariye dikkat edilirse altörnekleme için kullanılan evrişim katmanlarının sayısının çok fazla olduğu görülür. Altörnekleme için kullanılan son evrişim işleminden sonra derinliği 1024 olan özellik kanalı elde edilir. Sonuç olarak, altörnekleme ile üstörnekleme işlemleri, birbirlerine az ya da çok simetriktir ve u şekilli bir mimari sağlar. Son altörnekleme işleminden sonra elde edilen çıktı 28x28x1024 boyutlarındadır. U-net mimarisinde genişlik ve yükseklik değerleri 28'den küçük olan bir yapı bulunmaz. Bu, ağda tamamen bağlı katmanların olmadığı anlamına gelir. Bu noktada segmentasyon haritasının sadece girdi görüntüsündeki tam bağlamın mevcut olduğu pikselleri içerdiği söylenebilir.

U-net mimarisi genel olarak iki bölüme ayrılır. Sol taraf contracting path(daralan yol), sağ taraf ise expansive path(genişleyen yol) olarak isimlendirilir. Daralan yol tipik bir evrişimli ağ mimarisidir. Padding kullanmadan 3x3 boyutlarında, her iki evrişim katmanını takip eden ReLU aktivasyon fonksiyonları ile evrişim işlemi gerçekleştirilir. Devamında 2x2 boyutlarında havuzlama katmanı ile % 50 oranında bir alt örnekleme işlemi gerçekleştirilir. Her bir altörnekleme işleminden sonra derinlik, yani özellik kanallarının sayısı ikiye katlanır. Yukarıda bahsedilen işlem adımları daralan yol boyunca beş kez tekrar edilir ve daralan yolun en sonunda 28x28x1024 boyutlarında bir hacim oluşur. Genişleyen yoldaki her adım, genişlik ve yükseklik değerlerini iki katına çıkaran ve özellik kanalı sayısını yarı yarıya azaltan 2x2 boyutunda üst evrişim, devamında aktivasyon fonksiyonu olarak ReLU kullanılan 3x3 boyutunda evrişim işlemlerinden

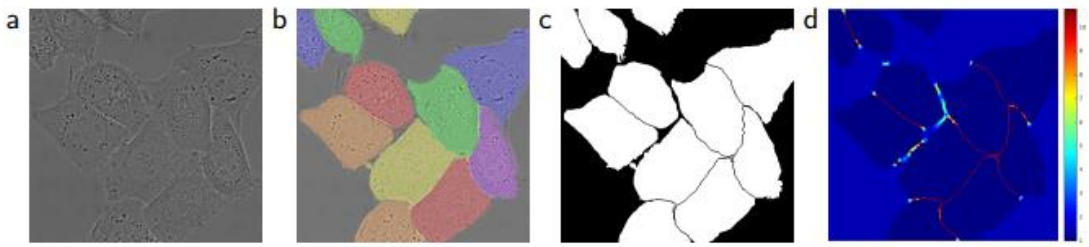
oluşur. 2x2'lik üst evrişim aynı zamanda mimarinin daralan yol tarafındaki simetriğinden kısayollar vasıtasıyla aktarılan özellik haritaları ile beslenir. Şekil 3.43'de aktarılan bu özellik haritaları beyaz kutular ile temsil edilmiştir. Şekil 3.43 dikkatle incelenirse bu aktarımlar sırasında aktarılan içeriğin hedef hacimine uygun olacak şekilde bir miktar kırpıldığı görülür. Mimarinin en sonunda segmentasyon haritasını elde etmek için 64 kanaldan oluşan özellik haritasına, 1x1 boyutunda evrişim uygulanır ve çıktıda istenen sayıda sınıftan oluşan segmentasyon çıkış haritası elde edilir.

Ağ, girdi görüntüleri ve bunlara karşılık gelen segmentasyon haritaları ile beraber skotastik gradyan inişi kullanılarak eğitilir. Padding uygulanmamış evrişim nedeniyle çıkıştaki segmentasyon haritalarının boyutu giriş görüntüsünün boyutundan küçük olur. Yükü en aza indirmek ve GPU belleğinden maksimum düzeyde yararlanmak için, büyük bir toplu iş boyutu yerine tek bir görüntü dikdörtgensel alanlara bölünür ve bu bölümler tek bir toplu iş olarak ele alınır.

Mimarinin en sonunda pikselleri sınıflandırmak için enerji fonksiyonu olarak adlandırılan bir fonksiyon kullanılır. Bu fonksiyon, son özellik haritası üzerinde, çapraz entropi kaybı fonksiyonu ile kombine edilmiş piksel tabanlı softmax fonksiyonu vasıtasıyla hesaplanır. Enerji fonksiyonu denklem 3.14'deki gibi tanımlanır.

$$E = \sum_{x \in \Omega} w(x) \log(p_{l(x)}(x)) \tag{3.14}$$

Burada *p*, softmax fonksiyonunu, *l*, her pikselin gerçek etiketini, w ise eğitimde bazı piksellerin önemini arttırmak için tanıtılan ağırlık haritasını temsil eder.



Şekil 3.44. Hücre Görüntülerinin U-Net Mimarisi İle Segmentasyonu[2]

Şekil 3.44’de a, ham görüntüyü, b, gerçek segmentasyon haritasını, c, ağın oluşturduğu segmentasyon haritasını, d ise sınır piksellerini temsil eder.

Sonuç olarak U-net mimarisi, farklı biyomedikal segmentasyon uygulamalarında çok iyi performans sağlar. Elastik deformasyonlarla veri büyütme sayesinde, yalnızca çok az açıklamalı görüntüye ihtiyaç duyar ve bir NVidia Titan GPU'da (6 GB) yalnızca 10 saatlik çok makul bir eğitim süresine sahiptir [2].

DeepLab: DeepLab [75], ilk olarak 2015 yılında tasarlanan ve açık kaynaklı, son teknoloji ürünü bir semantik segmentasyon modelidir. O zamandan beri modelde, DeepLab v2 [76], DeepLab v3 [77] ve en son DeepLab v3+ [78] modelleri ortaya konmuş ve bu modellerin her birinde önceki modele göre çok sayıda iyileştirme yapılmıştır. Bu modellerin arasında en son yayınlanan DeepLab v3+ modeli PASCAL VOC 2012 veri seti üzerinde %89 oranında başarım göstermiş ve o zamana kadar tasarlanmış en başarılı model olmuştur. Bu tez çalışmasında segmentasyon problemini çözmek için DeepLab v3+ modeli kullanılacaktır. Bu bölümde deeplab, deeplab v2 ve deeplab v3 modellerine sırasıyla değinilip son olarak deeplab v3+ modeli derinlemesine incelenecektir.

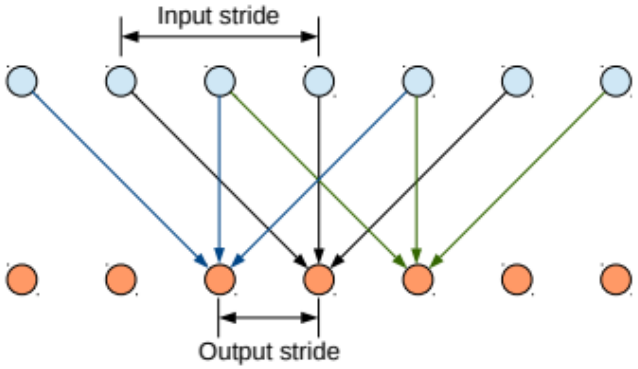
2015 yılında ortaya konulan ilk Deeplab modeli temel olarak evrişimli sinir ağları ile olasılıksal grafik modelleri yaklaşımlarının kombinasyonundan oluşan bir metot olarak ortaya konulmuştur. Evrişimli sinir ağları görüntü sınıflandırma ve görüntüdeki nesneleri tanıma görevlerinde başarılı olsada bu ağların son katmanı, yani çıktısı, nesne segmentasyonu için yeterli başarıyı gösteremez. Deeplab çalışmasında bu sorunun üstesinden gelmek için evrişimli sinir ağının son katmanı ile koşullu rastgele alanlar modelinin kombinasyonu ortaya konulmuştur.

Koşullu Rastgele Alanlar, çok yönlü sınıflandırıcılar tarafından hesaplanan sınıf puanlarını piksellerin ve kenarların veya süper piksellerin yerel etkileşimleri tarafından yakalanan düşük seviyeli bilgilerle birleştirmek için anlamsal bölümlemede yaygın olarak kullanılmıştır [75]. Bu çalışmada yazarlar, koşullu rastgele alanlar yönteminin hesaplama maliyetini kabul edilebilir seviyede tutmak için 2011 yılında yayınlanan Gauss kenar potansiyelleri ile tam bağlantılı koşullu rastgele alanlarlarda verimli çıkarım [79] isimli çalışmayı benimsemiştir.

İlk deeplab modeli, bu bölümde ayrıntılı olarak açıklanan tamamen evrişimli ağlar [5] modeline benzer şekilde çalışır. Tamamen evrişimli ağlar, bir evrişimli sinir ağının sonunda yer alan tamamen bağlı katmanlar yerine segmentasyon çıktısı üretmek için üst evrişim veya üst örnekleme denilen yapıları kullanır. Evrişim ve havuzlama işlemlerinin sonucu olarak düşen genişlik ve yükseklik boyutları bir takım ters evrişim teknikleri ile yeniden kazanılarak girdi ile aynı boyutta segmentasyon haritası elde edilir.

Evrişimli ağların segmetasyon problemlerine uyarlanması aşamasında yaşanan en büyük sorunlardan biri, yoğun evrişim ve havuzlama katmanları sonrasında kaybedilen yüksek miktarda anlamsal bilgi ve bu sırada ortaya çıkan yüksek hesaplama maliyetidir. İlk deeplab modelinde bu sorunun üstesinden gelmek için daha sonra atrous konvolüsyon ismini alacak olan hole(boşluk) algoritması yaklaşımı benimsenmiştir. Atrous kelimesi Fransızca à trous(delikleri olan) kelimelerinin birleşiminden gelmektedir.

Atrous algoritması, ondalıksız dalgacık dönüşümünün verimli hesaplanması için ilk olarak 1989 yılında yayınlanan Dalgacık Dönüşümü Yardımıyla Sinyal Analizi İçin Gerçek Zamanlı Bir Algoritma [80] isimli çalışmada ortaya konmuştur. Evrişimli ağlarda hole algoritması yada atrous algoritması olarak adlandırılan yapının yaptığı işlem temel olarak evrişim katmanlarında kullanılan filtrelerin arasına boşluklar ekleyerek filtreleri bir miktar genişletmektir. Bu sayede filtrenin görüş açısı genişletilmiş, böylece hem bir filtrenin aynı anda daha geniş alanı taraması sağlanmış hemde hesaplama maliyeti yüksek boyutlu girdiler için makul seviyeye çekilmeye çalışılmıştır. Uygulanan bu filtrenin parametreleriyle oynanarak elde edilen özellik haritasının boyutu da kolaylıkla belirlenebilir.



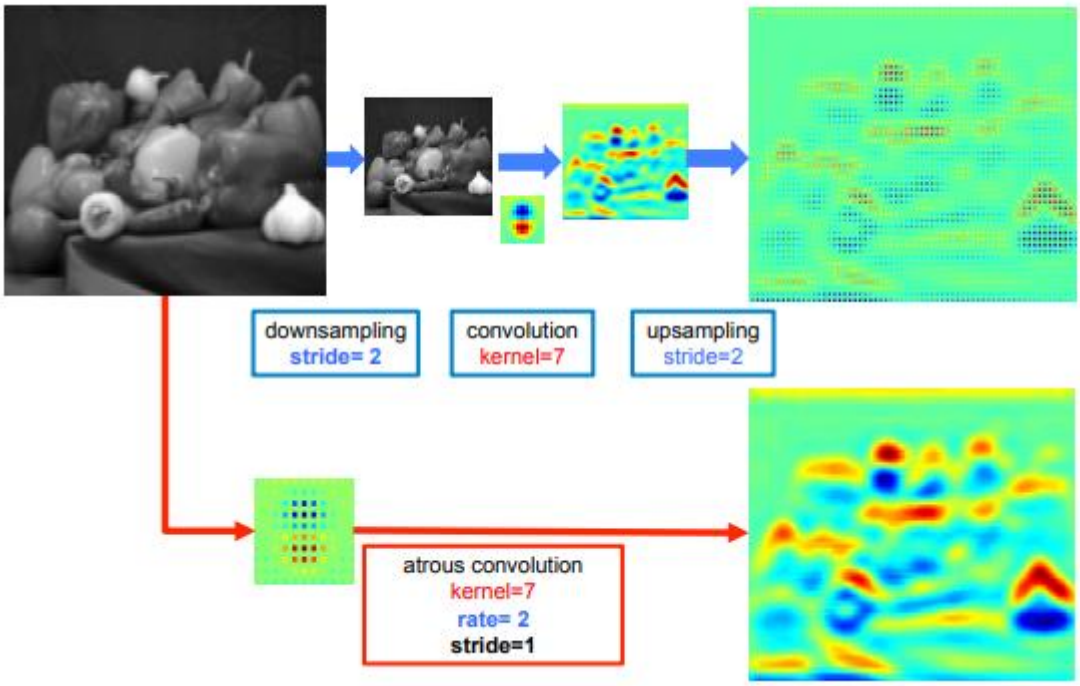
Şekil 3.45. Hole Algoritması Veya Atrous Algoritmasının Çalışma Mantığı[75]

Şekil 3.45'de hole algoritmasının çalışma mantığı basitçe gösterilmiştir. Üst satırda yer alan mavi daireler giriş katmanını, alt satır yer alan turuncu daireler çıkış katmanını gösterir. Uygulanan filtrenin çekirdek boyutu üç, giriş adım genişliği iki, ve çıkış adım genişliği değeri ise birdir. Giriş adım genişliği değeri filtreyi oluşturan birimlerin her birinin birbirinden ne kadar uzakta olduğunu ifade eder. Klasik konvolüsyonel ağlardaki filtrenin hareket ederken bir sonraki adımda ne kadar uzağa zıplayacağını ifade eden adım değeri ile karıştırılmamalıdır. Bir atrous filtrenin giriş adım değeri ve çıkış adım değeri parametreleri ile oynayarak çıkış hacminin boyutu kolaylıkla belirlenebilir. Bu durum evrişimli ağlarda büyük bir avantaj olarak kabul edilebilir.

Denklem 3.15'de x ile temsil edilen giriş özellik haritasına, w ile temsil edilen evrişim filtresinin uygulanması ve y ile temsil edilen çıktı özellik haritasındaki her bir i konumu için hesaplamanın nasıl yapıldığı gösterilmiştir.

$$y[i] = \sum_{k} x[i + r * k]w[k] \tag{3.15}$$

Burada r atrous evrişim filtresinde her bir ağırlık arasında yer alan boşluk değeridir.



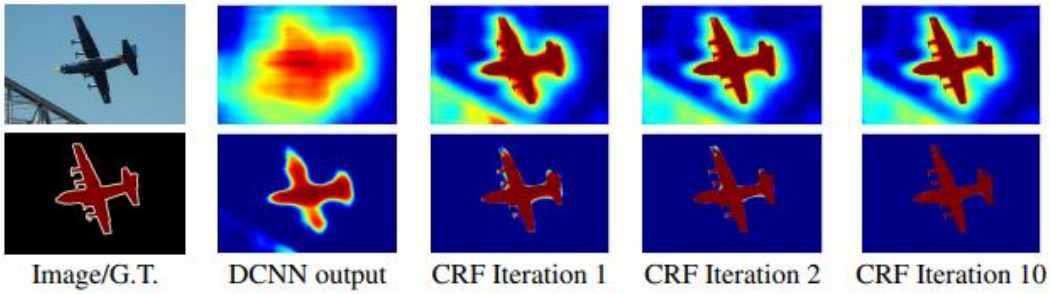
Şekil 3.46. Standart Evrişim ve Atrous Evrişim Sonuçları[76]

Şekil 3.46'da üst bölümde giriş görüntüsüne standart evrişim uygulanmış ve ortaya çıkan özellik haritası gösterilmiştir. Alt bölümde ise aynı giriş görüntüsüne atrous evrişim uygulanmış ve ortaya çok daha zengin bir özellik haritası çıkmıştır. Atrous evrişim işleminde, standart evrişim işleminde yer alan upsampling yani üst örnekleme adımı bulunmamaktadır. Atrous evrişimin ilgili parametrelerinin ayarlanması ile özellik haritasının giriş görüntüsü ile aynı boyutta oluşması sağlanmıştır. Bu örnekte evrişim filtresi olarak dikey gaussian türevi uygulanmıştır.

Atrous algoritması giriş hacmindeki farklı ölçekli nesneleri tanımada oldukça etkindir. Farklı ölçekteki atrous filtrelerin beraber kullanılmasıyla oluşturulan modüllerin görüntüdeki farklı ölçekteki nesnelerin tanınmasında üstün başarı gösterdiği görülmüştür.

Derin evrişimli sinir ağları tek başlarına şekil 3.47'de üst satır ikinci sırada görüldüğü gibi görüntüdeki bir nesnenin kaba sınırlarını belirlemede başarılı olabilir. Fakat görüntüdeki nesnelerin ayrıntılı sınırlarını belirlemede yetersiz kalırlar. Bunun temel nedeni birden fazla maksimum havuzlama katmanına sahip daha derin modellerin,

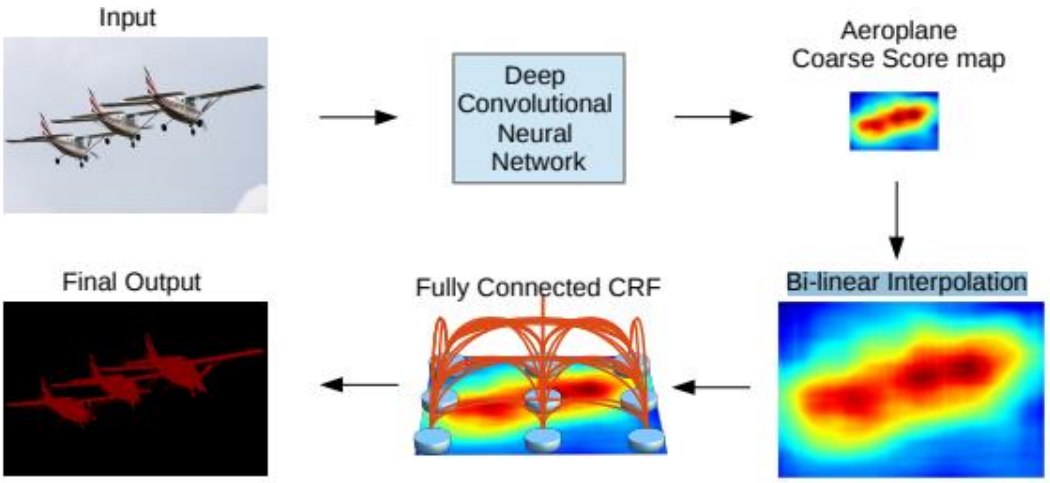
artan değişmezlikleri ve geniş alıcı alanları ile en üst çıktı seviyelerindeki skorlardan konum çıkarma sorununu daha zor hale getirmeleridir [75].



Şekil 3.47. Koşullu Rastgele Alanlar Yaklaşımı ile Segmentasyon Sonuçları[75]

Şekil 3.47'de üst satır evrişimli sinir ağının ürettiği skor haritasının ağın en sonunda yer alan softmax fonksiyonuna girmemiş halini, alt satır ise fonksiyona girmiş halini gösterir. İkinci sütun derin evrişimli sinir ağı tarafından üretilen kaba sınırları, devam eden sütunlar ise koşullu rastgele alanlar yöntemi tarafından iterasyonlar boyunca üretilen segmentasyon çıktılarını gösterir.

Koşullu rastgele alanlar yöntemi nesne sınırlarını tanımlamak için evrişimli ağ tarafından üretilen kaba sonuçları iyileştirmeye çalışır. Sinir ağı tarafından üretilen kaba sınır, koşullu rastgele alanlar işleminden geçtikten sonra daha rafine hale gelir.



Şekil 3.48. Deeplab Modeli Genel Yapısı[75]

Şekil 3.48'de görülen bi-linear enterpolasyon yani çift doğrulamalı interpolasyon işlemi evrişimli sinir ağı tarafından üretilen çıktının boyutunu giriş görüntüsü ile aynı boyuta yükseltmek için kullanılır. Modele hesaplama maliyeti olarak kayda değer bir yük getirmediği görülmüştür.

İlk deeplab modelinde eğitim aşamasında iki aşamalı bir yaklaşım benimsenmiştir. Birinci aşamada derin evrişimli sinir ağının eğitimi, ikinci aşama da ise koşullu rastgele alanlar yapısının eğitimi gerçekleştirilmişdir. Koşullu rastgele alanlar yapısının eğitimi sırasında evrişimli sinir ağı tarafından sağlanan tekli terimlerin sabitlendiği varsayılmıştır.

Modelde evrişimli sinir ağı için bölüm 3.4.3.'de kısaca değinilen ImageNet üzerinde önceden eğitilmiş olan VGG-16 modeli temel model olarak kullanılmıştır. Burada kullanılan VGG-16 modeli üzerinde, Pascal VOC 2012 verisetinin yirmi bir sınıflı çıktısına uygun olacak şekilde ince ayar yapılmıştır. Eğitimde skotastik dereceli alçalma, maliyet fonksiyonu olarak çapraz entropi, mini parti boyutu olarak yirmi girdi görüntüsü, başlangıç öğrenme değeri olarak 0.001 değeri kullanılmıştır. Her ikibin eğitim adımında bir öğrenme değeri 0.1 ile çarpılmıştır.

Deeplab mimarisinin ilk bölümünü oluşturan bu evrişimli ağın eğitimi tamamlandıktan sonra mimarinin ikinci bölümü olan koşullu rastgele alanlar yapısının eğitimi gerçekleştirilmiştir. Koşullu rastgele alanlar yapısının eğitimi, 2011 yılında ortaya konulan Gauss kenar potansiyelleri ile tam bağlantılı koşullu rastgele alanlarda verimli çıkarım [79] isimli çalışma çizgisinde gerçekleştirilmiştir.

Modelin çeşitli varyasyonlarının değerlendirilmesi önemli ölçüde PASCAL verisetinin doğrulama seti üzerinde gerçekleştirilmiştir. Bu değerlendirme sonuçları tablo 3.3'de görülmektedir.

Tablo 3.3. Deeplab Varyasyonlarının Pascal Voc 2012 Üzerinde Karşılaştırılması

Deeplab Varyasyonu	Ortalama IOU(%)
DeepLab	59.80
DeepLab-CRF	63.74
DeepLab-MSc	61.30
DeepLab-MSc-CRF	65.21
DeepLab-7x7	64.38
DeepLab-CRF-7x7	67.64
DeepLab-LargeFOV	62.25
DeepLab-CRF-LargeFOV	67.64
DeepLab-MSc-LargeFOV	64.21
DeepLab-MSc-CRF-LargeFOV	68.70

Tablo 3.3'de görüldüğü gibi deeplab varyasyonları arasında en iyi performans gösteren model, MSc, CRF ve LargeFOV yaklaşımlarını içeren modeldir.

MSc, bu bölümde bahsedilen tamamen evrişimli ağ modeline benzer şekilde, modelin orta katmanlarından son katmanlarına aktarılan kısayol bağlantıları vasıtasıyla gerçekleşir. MSc kullanmak tek başına modelin başarım performansını %1,5 oranında arttırır.

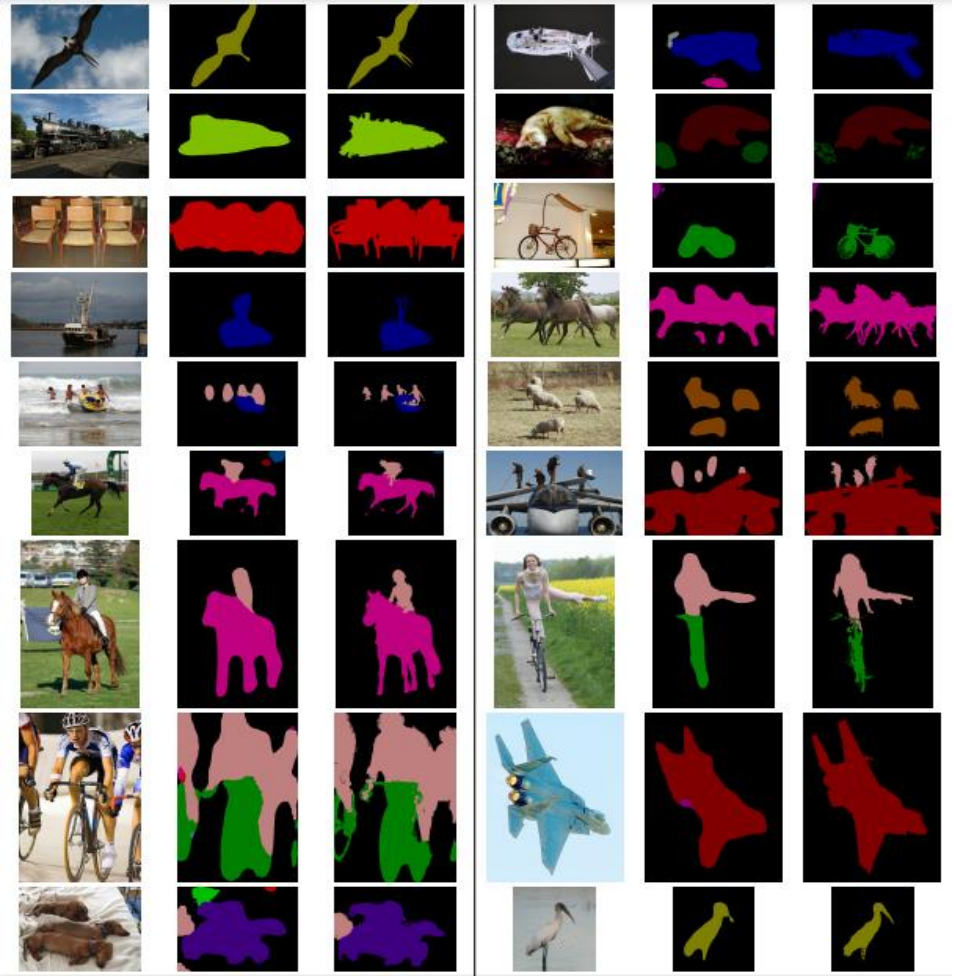
LargeFOV yaklaşımı temel olarak atrous evrişim filtrelerini ve bu filtrelerin çekirdek boyutu ve giriş adımı parametlerini kullanarak filtrelerin görüş alanını keyfi olarak değiştirmeye olanak verir. Giriş adımı parametresi filtrenin elemanları arasındaki adım mesafeleridir. Bu değer filtrenin görüş alanını belirler. Yüksek bir değer verilmesi filtrenin görüş alanınında büyümesi anlamına gelir.

Tablo 3.4'de farklı parametrelere sahip atrous filtreler ile modeldeli ilk tam bağlı katman üzerinde evrişim gerçekleştirilmiş ve bunların sonuçları gösterilmiştir. Özellikle son satırda yer alan DeepLab-CRF-LargeFOV varyasyonu küçük filtre boyutu, yüksek giriş adım değeri ile ilk satırda yer alan DeepLab-CRF-7x7 varyasyonu ile beraber en iyi başarımı göstermesine rağmen çok daha az maliyetli bir varyasyon olduğunu kanıtlamıştır. Bunun temel sebebinin kullanılan giriş adımı değerinin oniki gibi yüksek bir değer olmasının olduğu düşünülebilir. Özellikle görüntüdeki büyük ölçekli nesneleri daha az sayıda parametre ile yakalamak konusunda giriş adımı değerinin öneminin büyük olduğu söylenebilir. Bu tez çalışmasında, görüntülerdeki hayvanların görüntünün büyük

bir alanını kaplayacağını, yani büyük ölçekli nesneler olacağını öngörürsek bu yaklaşımı benimsememizin ağımızın başarım ve performansını arttıracağını düşünebiliriz.

Tablo 3.4. Deeplab Modelinin Farklı Parametre Değerleriyle Performans Sonuçları

Deeplab varyasyonu	Çekirdek boyutu	Giriş adımı	Alıcı alan büyüklüğü	Parametre sayısı	Ortalama IOU(%)	Eğitim hızı (görüntü/saniye)
DeepLab-CRF-7x7	7x7	4	224	134.3M	67.64	1.44
DeepLab-CRF	4x4	4	128	65.1M	63.74	2.90
DeepLab-CRF-4x4	4x4	8	224	65.1M	67.14	2.90
DeepLab-CRF-LargeFOV	3x3	12	224	20.5M	67.64	4.84



Şekil 3.49. İlk Deeplab Modelinin Pascal Voc-2012 Üzerindeki Sonuçları

Şekil 3.49'de ilk deeplab modelinin PASCAL VOC-2012 doğrulama seti üzerinde sonuçları görülmektedir. Burada her bir satırdaki ilk resim giriş resmini, ikinci resim evrişimli sinir ağı tarafından oluşturulan segmentasyon çıktısını, üçüncü resim ise evrişimli sinir ağı tarafından oluşturulan segmentasyon çıktısının koşullu rastgele alanlar yaklaşımı ile rafine edilmiş halini göstermektedir. Görüldüğü gibi ilk deeplab modeli için koşullu rastgele alanlar yaklaşımı modelin başarımını önemli ölçüde arttırmaktadır.

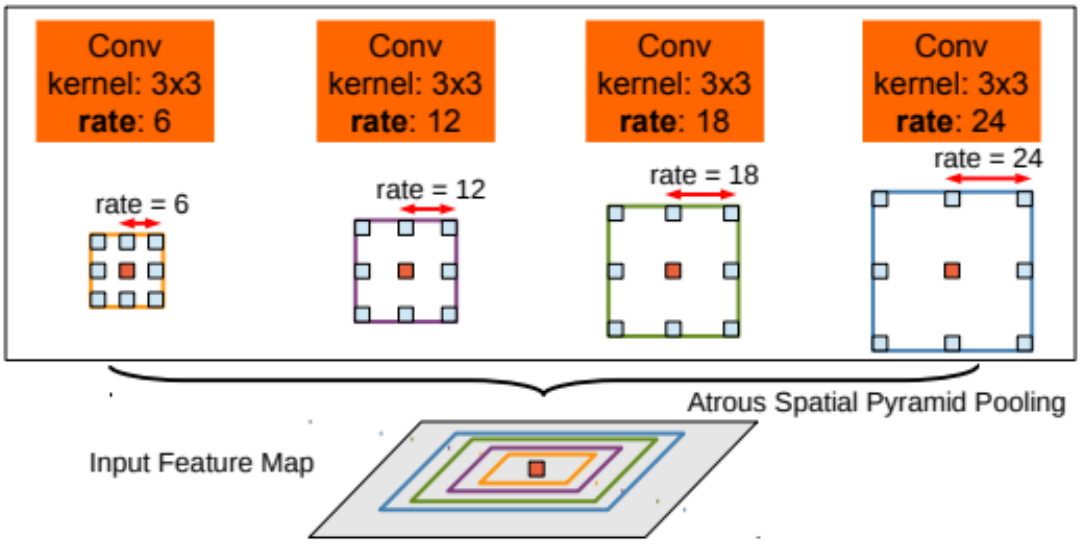
Sonuç olarak ilk deeplab modeli evrişimli sinir ağları ile koşullu rastgele alanlar yaklaşımının bir kombinasyonunu gerçekleştirerek semantik segmentasyon problemlerinde başarılı ve hesaplama maliyeti olarak makul sayılabilecek bir model olarak ortaya konulmuştur. Model geliştirilme aşamasında iken farklı varyasyonlar denenerek modelin başarımının arttırılmasına çalışılmıştır.

İkinci deeplab versiyonu ilkinden yaklaşık iki yıl sonra ortaya konulmuştur. Bu versiyonda ilk versiyona göre üç ana katkı gerçekleştirilmiştir. Bunlardan birincisi, aslında ilk versiyonda da kullanılan ve atrous evrişim denilen filtreleme tekniğinin yoğun tahmin görevleri için ilk versiyona göre daha etkin bir şekilde kullanılmasıdır. Atrous evrişim, derin öğrenme modeli boyunca özellik haritalarının çözünürlüğünü doğrudan kontrol etmek açısından önemlidir. Aynı zamanda görüntüdeki büyük ölçekli nesneleri yakalamak için ağdaki parametre sayısını arttırmadan filtrelerin görüş alanının genişletilebilmesini sağlar. İkinci katkı, görüntüde yer alan farklı ölçeklerdeki nesnelerin segmentasyonu için atrous uzamsal piramit havuzlama olarak isimlendirilen yapının kullanılmasıdır. Bu yapı kendisine gelen özellik haritalarını farklı örnekleme değerlerine sahip, yani farklı görüş alanına sahip filtrelerle işler ve böylece görüntüde yer alan farklı ölçekteki nesneleri etkin bir şekilde yakalar. Üçüncü katkı ise ilk modele benzer şekilde derin evrişimli sinir ağları ile olasılıksal grafik modellerin kombinasyonunun daha etkin bir biçimde ortaya konulmasıdır. Bu modelde de yine birinci modelde olduğu gibi koşullu rastgele alanlar yaklaşımı benimsenmiştir. Model PASCAL VOC-2012 test seti üzerinde %79.7 başarıma ulaşarak ilk versiyona göre yaklaşık %12'lik bir iyileşme göstermiştir.

Modelde yine temel olarak tamamen evrişimli sinir ağının son birkaç katmanında yer alan havuzlama katmanları kaldırılmış, yerlerine genellikle yüksek örnekleme değerlerine sahip üstörnekleme filtreleri içeren katmanlar eklenmiştir. Bu üstörnekleme filtreleri, filtre elemanları arasında boşluk içeren meşhur atrous filtreleridir. Atrous filtre

içeren katmanlar ile ortaya çıkarılan, nesne sınırlarını içeren yoğun özellik haritalarını, orijinal görüntü boyutuna yükseltmek için yine bir önceki modelde olduğu gibi çift doğrusal interpolasyon yöntemi kullanılmıştır.

İkinci deeplab versiyonunda ortaya konulan en büyük yenilik atrous uzamsal piramit havuzlama modülü olarak görülebilir. Bu modül temel olarak görüntüdeki aynı nesnelerin farklı ölçeklerdeki hallerini yakalamak için kullanılır. Bu konuda standart yol aynı nesneyi içeren görüntünün, farklı ölçeklerdeki hallerini evrişimli ağa göndermek ve oluşan skor veya özellik haritalarını toplamaktır. Bu yol başarım olarak kabul edilebilir olmasına rağmen hesaplama maliyeti açısından ağa ekstra yük getirir. ASPP ise tek ölçekli bir görüntüyü farklı boyuttaki atrous filtrelerden oluşan modül ile işler ve ortaya, görüntüdeki nesnelerin farklı ölçeklerdeki hallerini içeren özellik haritaları çıkarır. Bu yöntem klasik yöntemlere göre hesaplama maliyeti açısından çok daha etkindir. Kısaca ASPP, farklı boyutlarda ve giriş adımı değerlerinden oluşan atrous filtreleri içeren, birbirlerine paralel atrous evrişim katmanlarının kombinasyonundan oluşan bir yapı olarak tanımlanabilir.



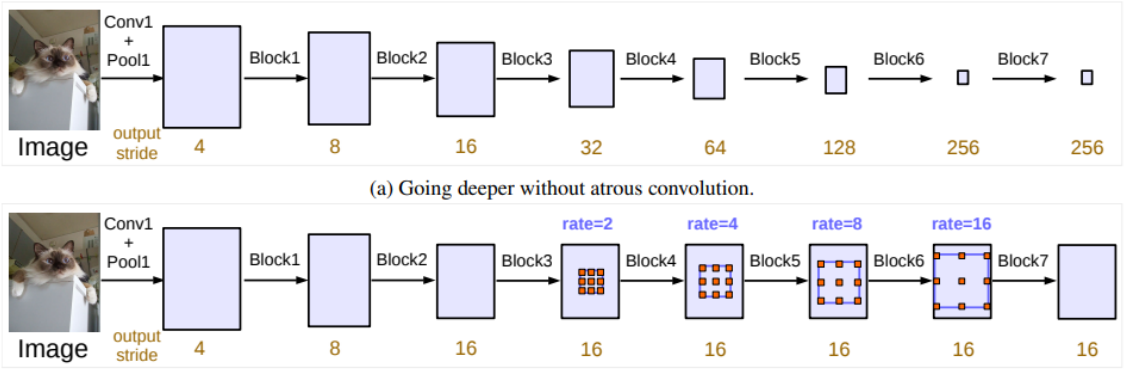
Şekil 3.50. ASPP Modül Yapısı[76]

İkinci deeplab versiyonunda temel model olarak VGG-16 ile beraber ResNet-101 modeli de eğitim sırasında kullanılmış ve sonuçları değerlendirilmiştir. Doğal olarak modellerdeki tam bağlantılı katmanlar kaldırılmış, yerlerine altörnekleme işlemlerini

belli bir seviyede tutan, özgün ağdaki her 32 piksel yerine her 8 pikselde bir öznitelik yanıtlarının hesaplanmasına izin vererek, atrous evrişimli katmanlar aracılığıyla özellik çözünürlüğünü arttıran ve böylece ağın derin katmanlarında çok fazla veri kaybının oluşmasını engelleyen bir yapı benimsenmiştir. Temel model olarak ResNet-101 modelinin kullanılması, VGG-16 modelini temel model olarak kullanan ilk deeplab versiyonuna göre daha iyi bir segmentasyon başarımının ortaya konmasını sağlamıştır.

2017 yılında yayınlanan üçüncü deeplab versiyonunda [77] temel olarak görüntüde farklı ölçeklerde yer alan nesnelerin segmentasyonu için kullanılan atrous evrişim yeniden ele alınmıştır. İkinci versiyonda ortaya konulan ASPP modülünün performansı arttırılmış, global kapsamda görüntü seviyesindeki özelliklerin kodlaması gerçekleştirilmeye çalışılmıştır. DeepLabv3 ismi ile ortaya konulan modelde bir önceki modele göre en büyük fark modelin sonunda yer alan koşullu rastgele alanlar yapısının artık kullanılmaması olmuştur. Model, PASCAL VOC 2012 veri setinde %85.7 başarım düzeyine ulaşarak deepLabv2 modeline göre yaklaşık %6'lık bir iyileşme göstermiş ve o güne kadar ortaya konulmuş diğer segmentasyon modelleri ile de karşılaştırılabilir bir başarıma erişmiştir.

DeepLabv3 mimarisinde temel model olarak ResNet mimarisi kullanılmıştır. ResNet mimarisinin son bloğu birkaç kez çoğaltılmış ve kademeli olarak düzenlenmiştir. Çoğaltılan bu bloklarda şekil 3.50'de görüldüğü gibi sırasıyla 2, 4, 8 ve 16 adım değerlerine sahip atrous evrişim filtreleri uygulanmıştır. Bunun arkasındaki neden herzaman olduğu gibi görüntüdeki global bilgiyi korurken detaylı bilgiyi de elde edebilmektir. Örneğin atrous evrişim kullanılmamış bir modelde son bloktaki özellik haritaları çok düşük çözünürlüklü olacağı için görüntüdeki global bilgiyi özetleyebilse bile detaylı bilgiyi, yani nesnelerin ayrıntılı sınırlarını ortaya koymakta başarısız olacaktır. Şekil 3.51'de ikinci bölümde görüldüğü gibi atrous evrişim kullanmak ağın son bloklarındaki özellik haritalarının yüksek çözünürlüklü kalmasını sağlamış ve giriş görüntüsündeki detaylı bilginin ve global bilginin elde edilmesini sağlayabilmiştir.



Şekil 3.51. Deeplabv3 Modelinde Atrous Evrişim Blokları [77]

Modelde yine bir önceki modelde olduğu gibi ASPP yapısı kullanılmıştır. Önceki modelden farklı olarak deeplabv3 modelinde ASPP yapısına, ince ayarı yapılmış toplu normalleştirme ve görüntü düzeyinde özellikler dahil edilmiştir. Toplu normalleştirme parametreleri belirlenirken 2015 yılında yayınlanan "Toplu normalleştirme: Dahili ortak değişken kaymasını azaltarak derin ağ eğitimini hızlandırma" [81] çalışması esas alınmıştır.

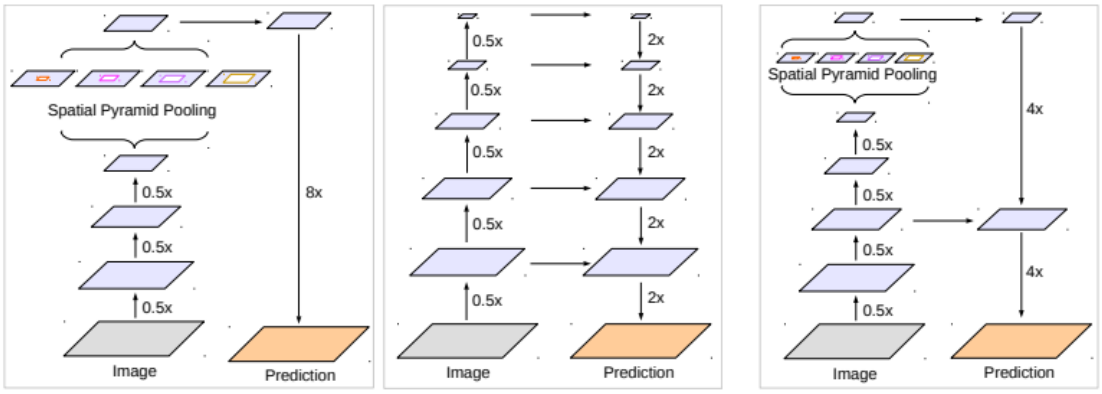
Önceki iki modelde olduğu gibi tüm görüntü yerine görüntüden parçalar kırpılıp bu parçalar eğitim için kullanılmıştır. Görüntüden kırpılan parçaların boyutu belirlenirken atrous evrişim filtrelerinin giriş adımı değerleri dikkate alınmıştır. Büyük giriş adımı değerine sahip bir atrous evrişim filtresini küçük boyutlu bir parçaya uygulamak modelin etkinliğini düşüreceği için giriş adımı ve kırpma büyüklüklerine dikkat edilmiştir. Bu sebeple kırpma büyüklüğü 513 piksel olarak belirlenmiştir.

2018 yılında ortaya konulan dördüncü DeepLab versiyonu [78] deepLabv3+ olarak isimlendirilmiştir. Bu modelde getirilen en büyük yenilik mimarinin kodlayıcı-kod temel yapısına göre dizayn edilmesi olmuştur. Önceki modeller görüntüdeki çok ölçekli bağlamı yakalayabilmek için özellik haritalarını çok ölçekli filtre ve havuzlama katmanları ile değerlendirmeye, devam eden modeller ise keskin sınırları kademeli bir şekilde belirlemeye çalışmışlardır. Bu çalışmada önerilen deepLabv3+ modeli, deepLabv3 modelinin geliştirilmiş bir versiyonu olarak, özellikle nesne sınırları boyunca segmentasyon sonuçlarını iyileştirmek için basit ama etkili bir kod çözücü modülü ekleyerek, bu iki yaklaşımın avantajlarını birleştiren bir çalışma olarak ortaya

konulmuştur. Temel model olarak Xception modeli kullanılmış, ASPP ve kod çözünü yapısına derinlemesine ayrılabilir evrişim katmanları uygulanmıştır. Model PASCAL VOC 2012 verisetinde %89 gibi bir başarım değerine ulaşarak o güne kadar ortaya konulan en başarılı segmentasyon modellerinden biri haline gelmiştir.

Çalışma boyunca görüntü işlemede kullanılan iki temel derin öğrenme yapısı olan uzamsal piramit havuzlama ve kodlayıcı-kod çözücü yapıları üzerinde durulmuştur. Bunlardan ilki farklı çözünürlükteki piramit havuzlama özelliklerini kullanarak görüntüdeki zengin bağlamsal bilgiyi yakalarken ikincisi nesnelerin keskin sınırlarını elde etmek amacıyla geliştirilmiştir.

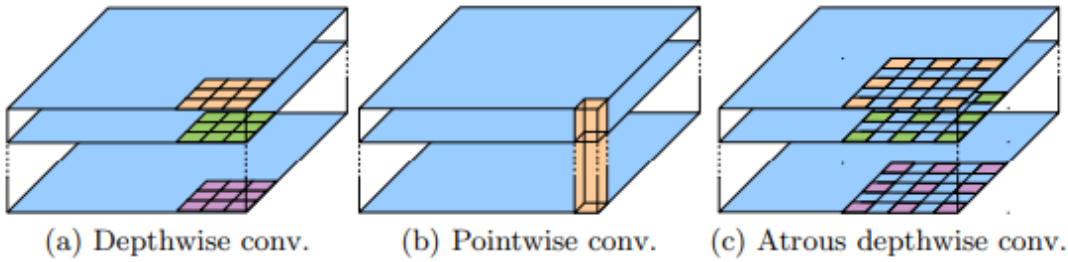
Son öznitelik haritasında zengin semantik bilgiler kodlanmış olsa da, ağ omurgası içinde havuzlama veya kademeli işlemlerle yapılan evrişimler nedeniyle nesne sınırları ile ilgili ayrıntılı bilgiler eksiktir. Bu, daha yoğun özellik haritalarını çıkarmak için atrous konvolüsyon uygulanarak hafifletilebilir. Yine de evrişimli ağların genel yapısı ve sınırlı GPU kapasiteleri nedeniyle giriş görüntüsünden dört veya sekiz kat küçük özellik haritalarını elde etmek hesaplama açısından engelleyicidir [78]. Bu noktada kodlayıcı-kod çözücü mimarileri kodlayıcı yolunda daha hızlı hesaplamaya izin verir ve kod çözücü yolundaki keskin nesne sınırlarını kademeli olarak kurtarır. Bunun temel sebebi bu mimarilerde kodlayıcı ve kod çözücü bölümlerinde yer alan katmanlar arasında gerçekleştirilen kısayol bağlantılarıdır. Bu mimarilere en iyi örneklerden birisi bölüm 4.3'de açıklanan U-Net mimarisidir. Özetlemek gerekirse bu çalışmada, uzamsal piramit havuzlama ve kodlayıcı-kod çözücü yapılarının avantajları bir araya getirilmiş, özellikle kodlayıcı modülünün zenginleştirilmesine çalışılmıştır. Bir araya gelerek oluşan bu yapı şekil 3.52'de (c) bölümünde görülebilir.



Şekil 3.52. Uzamsal Piramit Havuzlama Ve Kodlayıcı-Kod Çözücü Yapısı[78]

Bu modeldeki diğer bir yenilik, derinlemesine ayrılabilir evrişim katmanlarının mimariye dahil edilmesi olmuştur. Amacı, modelin segmentasyon başarım performansını korurken hesaplama maliyetini düşürmektir. Derinlemesine ayrılabilir evrişim yaklaşımı bir çok derin öğrenme mimarisinde benimsenmiştir.

Derinlemesine ayrılabilir evrişim temel olarak, standart bir evrişim işlemini, noktasal bir evrişim işlemini takip ederek derinlemesine bir evrişime göre çarpanlarına ayırma işlemi olarak tanımlanabilir. Bu evrişimli ağlarda görülen hesaplama karmaşıklığını önemli ölçüde azaltır. Noktasal evrişim, derinlemesine evrişimden elde edilen çıktıyı birleştirmek için kullanılırken derinlemesine evrişim, her bir giriş kanalı için bağımsız olarak bir uzaysal evrişim gerçekleştirir. Şekil 3.53'de görüldüğü gibi derinlemesine evrişim, atrous evrişim ile birleştirilmiştir ve adına atrous derinlemesine evrişim denilmiştir.



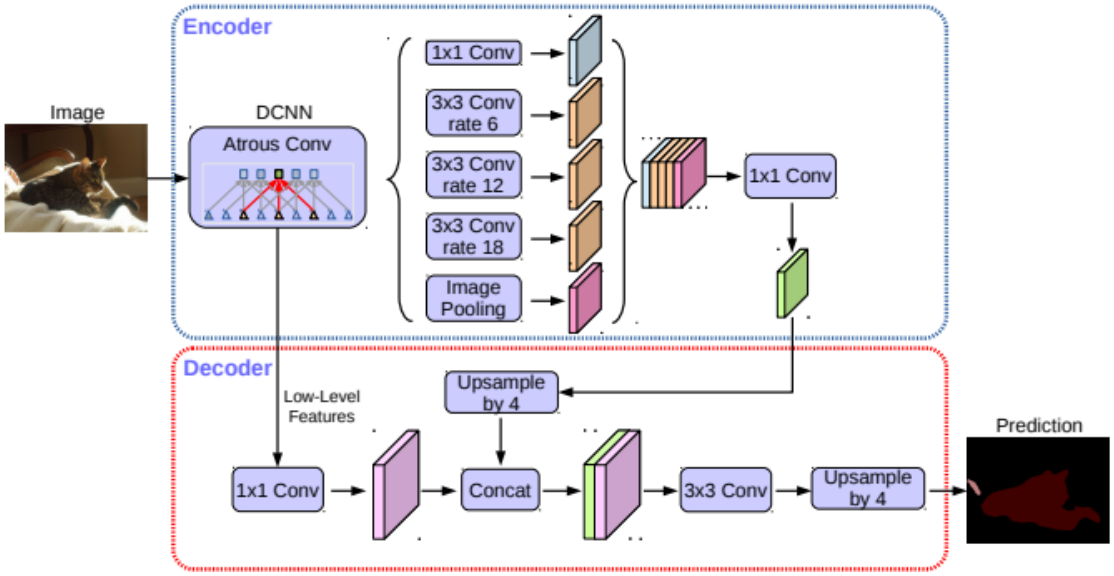
Şekil 3.53. Atrous Derinlemesine Evrişim Gösterimi[78]

Şekil 3.53 (a)'da görüldüğü gibi derinlemesine evrişim tek bir evrişim filtresini her bir giriş kanalına uygular. (b)'de görüldüğü gibi noktasal evrişim ise her kanaldan gelen derinlemesine evrişimin çıktılarını birleştirir. (c)'de atrous evrişim ile derinlemesine evrişimin adaptasyonundan oluşan atrous derinlemesine evrişim yapısı görülmektedir.

Mimarinin kodlayıcı bileşeni olarak deepLabv3 modeli tercih edilmiştir. İstenilen çözünürlükte özellik haritalarının çıkarılması için atrous evrişim filtreleri kullanılmıştır. Mimaride ortaya konulan çıkış adımı parametresi(output stride), giriş görüntüsü büyüklüğünün çıkış görüntüsü büyüklüğüne oranı olarak ifade edilir. Görüntü sınıflandırma ağlarında bu oran genellikle 32'dir. Yani bir görüntü sınıflandırma mimarisinin, en sonunda yer alan tamamen bağlı katmandan hemen önce oluşturduğu özellik haritasının çözünürlüğü giriş görüntüsünden 32 kat küçüktür. Bu durumda böyle bir mimari için çıkış adım değerinin 32 olduğu söylenebilir. Semantik segmentasyon görevlerinde ise çıkış adımı değeri genellikle 16 veya 8 olur. Bunun sebebi, detaylı bilgiyi kurtarabilmek için mimarinin sonunda daha yoğun bir özellik haritası istenmesidir. Bunu yapmak için klasik bir görüntü sınıflandırma mimarisinin en sonunda yer alan bir veya iki blok kaldırılır ve yerine atrous evrişim filtreleri içeren katmanlar eklenir. Sonuç olarak mimarinin kodlayıcı kısmının en sonunda yer alan son özellik haritasının çıkış adım değeri genellikle 16'dır ve bu özellik haritası, kodlayıcı kısmın nihai sonucunu temsil eder. Bu özellik haritası 256 kanala sahiptir ve bu sayede zengin semantik bilgiyi barındırır.

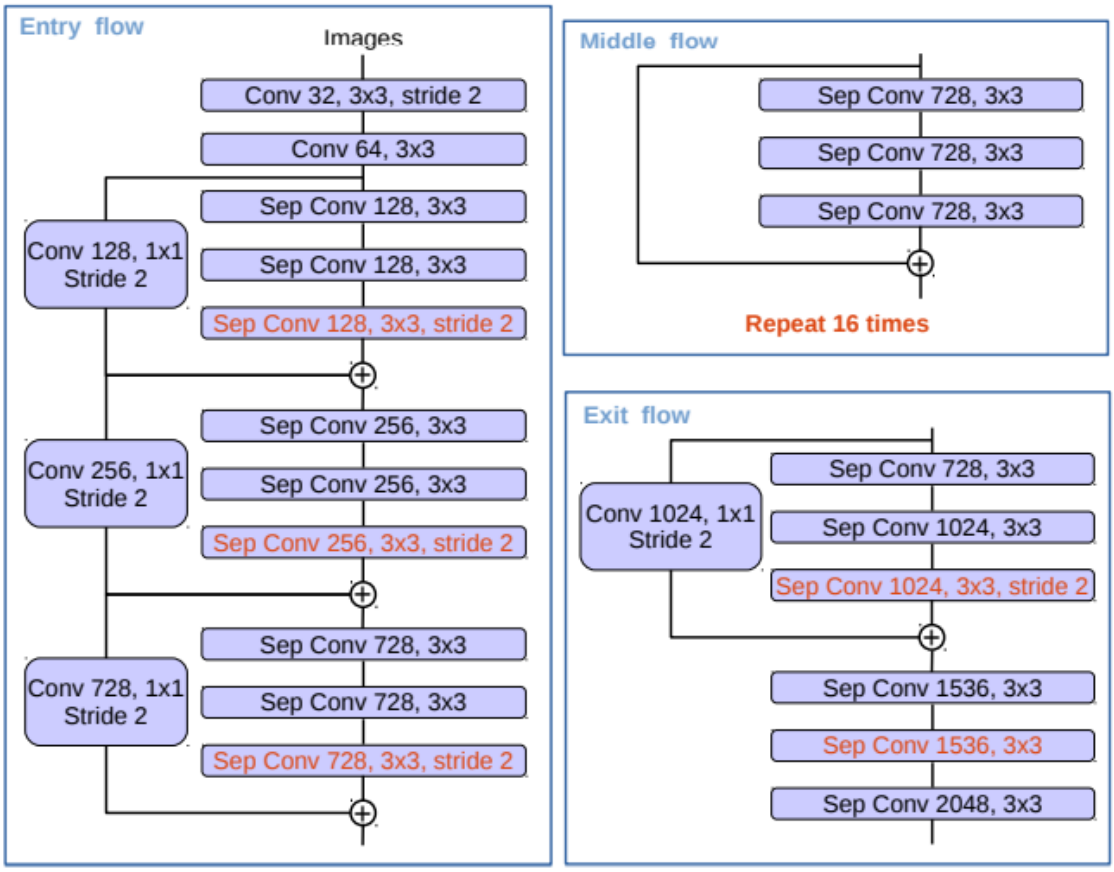
Önceki model olan deepLabv3 mimarisinde 16 kat alt örneklenmiş olan özellik haritalarını giriş görüntüsü ile aynı boyuta yükseltmek için çift doğrulamalı enterpolasyon kullanılır. Bu işlem deeplabv3+ mimarisi için temel bir kod çözücü modülü olarak kabul edilebilir. Tek başına böyle bir temel yaklaşım, ayrıntılı nesne sınırlarını ortaya koyma noktasında yetersiz kalacaktır. Bu sebeple mimaride, şekil 79'da alt bölümde görüldüğü gibi basit ama etkin bir kod çözücü modülü önerilmiştir. Kodlayıcı modülünün çıktıları olan özellikler çift doğrulamalı enterpolasyon kullanılarak ilk önce 4 kat yukarı örneklenmiş, devamında karşılık gelen düşük seviyeli özelliklerle birleştirilmiştir. Birleştirme işleminden sonra sonuçları daha rafine hale getirmek için 3x3 boyutunda bir evrişim işlemi uygulanmıştır. Bu evrişim işleminden sonra ortaya çıkan özellik haritası

çift doğrulamalı enterpolasyon kullanılarak tekrar 4 kat yukarı örneklenmiş ve sonuç olarak çözünürlük toplamda 16 kat yükseltilmiştir.



Şekil 3.54. Deeplabv3+ Mimarisinin Genel Yapısı[78]

Model geliştirilirken deneysel aşamada, temel model olarak, MSRA takımı tarafından modifiye edilerek, hizalanmış Xception [82] ismini alan model kullanılmıştır. MSRA takımı tarafından modifiye edilen Xception mimarisi görüntü sınıflandırma problemlerinde üstün performans göstermiştir. Bu sebeple deepLabv3+ mimarisi semantik segmentasyon problemlerini çözmek için bu mimariyi temel mimari olarak deneylemiştir. Temel olarak kullanılan bu mimaride birkaç değişikliğe gidilmiştir. Daha hızlı hesaplama ve bellek verimliliği için mimarinin girişindeki ağ yapısı üzerinde değişiklik yapılmıştır. Mimarideki tüm maksimum havuzlama katmanlarının yerine istenilen çözünürlükte özellik haritaları elde edilmesini sağlayan atrous derinlemesine evrişim katmanları kullanılmıştır. Mimariye ekstra toplu normalizasyon uygulanmıştır. 3x3 boyutunda her derinlemesine evrişim katmanına ReLu aktivasyon fonksiyonu eklenmiştir. Şekil 3.55'de ortaya çıkan bu temel mimari görülebilir.



Şekil 3.55. Deeplabv3+ İçin Modifiye Edilmiş Xception Mimarisi[78]

Modelin eğitim aşamasında önceki deeplab modellerinde olduğu gibi çoklu öğrenme oranı politikası, 513x513 piksel kırpma boyutu, ince ayarlı toplu normalizasyon, çıkış adımı değeri 16 ve eğitim sırasında rastgele ölçekli veri arttırma yaklaşımları benimsenmiştir.

Mimarinin ikinci kısmını oluşturan kod çözücü modülünde farklı tasarım seçenekleri için üç ana alan düşünülmüştür. Bunlardan birincisi kodlayıcı modülünden gelen, yüksek sayıda kanala sahip, düşük seviyeli özellik haritalarındaki kanal sayısını azaltmak için uygulanacak 1x1 evrişim alanıdır. İkincisi daha keskin segmentasyon sınırları elde etmek için kullanılacak olan 3x3 boyutundaki evrişim alanıdır. Üçüncü alan ise kodlayıcı modülünden gelen düşük seviyeli özelliklerden hangilerinin kullanılması gerektiğini ortaya koymaya çalışacak olan alandır.

Sonuç olarak deepLabv3+ mimarisi kodlayıcı-kod çözücü genel yapısı içerisinde ortaya konulmuş bir modeldir. Mimarinin kodlayıcı kısmı, zengin bağlamsal bilgiyi yakalamak için bir önceki mimari olan deepLabv3 mimarisini kullanır. Keskin nesne sınırlarını yakalamak için ise basit ama etkili bir kod çözücü yapısı kullanılır. Xception mimarisini temel model olarak kullanan deepLabv3+ mimarisi, PASCAL VOC 2012 test seti üzerinde %89 başarım seviyesine ulaşmıştır ve o güne kadar ortaya konulmuş en başarılı segmentasyon modellerinden biri olmuştur.



Şekil 3.56. Deeplabv3+ Mimarisinın Doğrulama Seti Üzerindeki Sonuçları[78]

Şekil 3.56'de son satırda yer alan görseller, deepLabv3+ mimarisinin tanımlanması zor nesneler üzerindeki başarısız segmentasyon sonuçlarını gösterir. Üstteki iki satırda ise başarılı sonuçlar görülmektedir.

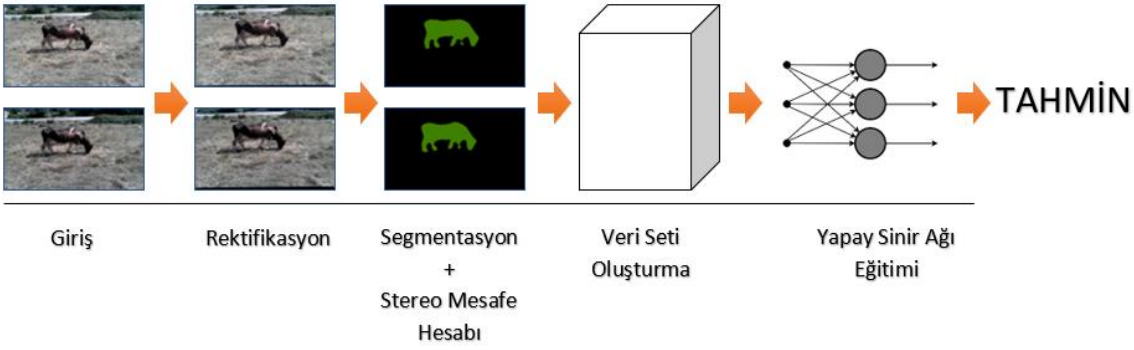
BÖLÜM 4. GERÇEKLEŞTİRİLEN ÇALIŞMA

Bu bölümde çalışma hakkında detaylı bilgiler verilecektir. Bu tez çalışmasında temel hedef stereo görme ve semantik segmentasyon yöntemlerini beraber kullanarak canlı hayvan ağırlını tahmin etmektir. Çalışmada Anaconda çerçevesi dahilinde kullanılabilen python programlama dili ve bununla beraber bilgisayar görüsü için bu programlama dilinde yaygın şekilde kullanılan openCV isimli kütüphane, kamera kalibrasyonu ve stereo düzeltme gibi problemler için kullanılacaktır. openCV kütüphanesinde kullanıma hazır şekilde bulunan fonksiyonlar sayesinde kamera kalibrasyonu, stereo düzeltme gibi problemler kolaylıkla çözülebilmektedir.

Çalışmanın yapay zekâ kısmında bulunan konvolüsyonel ağlar ve konvolüsyonel ağ tabanlı semantik segmentasyon problemleri için yine python programlama dilinde yaygın olarak kullanılan keras ve tensorflow kütüphanelerini kullanılacaktır. Bu kütüphanelerden özellikle keras, yazım kolaylığı sayesinde oldukça yaygın şekilde kullanılmaktadır. Keras, tensorflow kütüphanesinin üzerine kurulmuştur. Tensorflow kütüphanesinin nispeten karmaşık olan yazımını basitleştirmek amacı ile ortaya çıkan keras kütüphanesi yapay zekâ alanındaki birçok problemi nispeten basit sayılabilecek bir yazım zorluğu ile çözebilmektedir. Ayrıca keras ile yapılmış birçok uygulamaya kolaylıkla ulaşılabilmektedir.

Çalışmada donanım olarak iki adet standart kamera ve raspberry pi 4 ile kurulmuş bir stereo görme düzeneği ile yerel çalışmalar için Intel Core2Duo işlemcili, 4GB Ram, 128 GB SSD sabit disk ve NVIDIA GeForce 9300M GS ekran kartı donanımına sahip bir bilgisayar kullanılacaktır. Yoğun işlem gücü gerektiren uygulamalar için ise Google Colaboratory ortamında ücretsiz olarak sunulan GPU ve TPU donanımları kullanılacaktır.

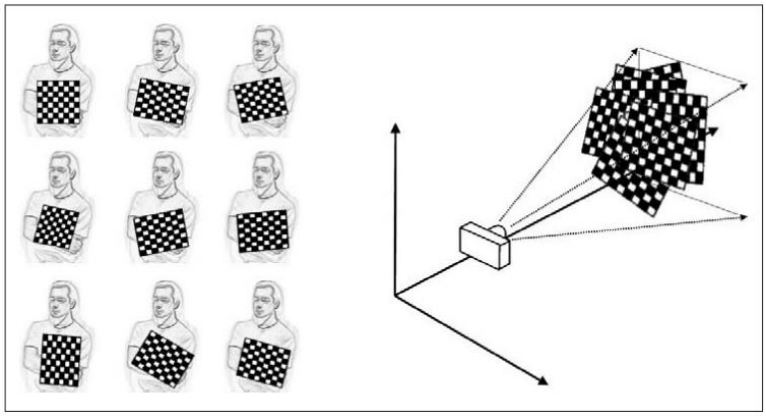
Çalışmanın temel iş akışı şekil 4.1'de görülmektedir.



Şekil 4.1. Temel İşlem Adımları

Çalışmaya ilk olarak, stereo düzenek oluşturmak için kullanılacak, birbirine eşdeğer olan iki adet standart web kamerası kalibre edilerek başlanmıştır. Kalibrasyon için 8x7 boyutlarında birbirlerine eşdeğer karelerden oluşan bir satranç tahtası kullanılmıştır. Bu kalibrasyon sonucunda elde edilen kalibrasyon matrisleri şekil 4.3 ve 4.4'de görülmektedir.

OpenCV kütüphanesini kullanarak bir kamerayı kalibre etmek için şekil ve ölçüleri bilinen bir nesne kullanmak gerekir. Yapılan çalışmalarda genellikle bir satranç tahtası veya içinde dairesel şekiller bulunan bir nesne kullanılır. Satranç tahtası kullanarak kamerayı kalibre etmek için ilk önce satranç tahtasının farklı mesafe ve açılardan fotoğrafları çekilir. Yaklaşık 20 adet fotoğraf kalibrasyon için yeterlidir. OpenCV'nin kameraya yaklaşımı 1997 yılında Heikkila J. ve Silven O. tarafından yapılan "A four-step camera calibration procedure with implicit image correction" [32] isimli çalışmaya dayanmaktadır.



Şekil 4.2. Satranç Tahtası İle Kamera Kalibrasyonu[30]

Çekilen bu fotoğraflar openCV kütüphanesindeki hazır fonksiyonlar ile değerlendirilerek kameranın içsel matrisi ve bozulma katsayıları belirlenir.

```
%YAML:1.0
---
K: !!opencv-matrix
   rows: 3
   cols: 3
   dt: d
   data: [ 6.4337010706337560e+02, 0., 3.0864333108498880e+02, 0.,
       6.4337870706186698e+02, 2.4275232703992199e+02, 0., 0., 1. ]
D: !!opencv-matrix
   rows: 1
   cols: 5
   dt: d
   data: [ -4.0514242606495124e-02, 4.3275361448902044e-01,
       -1.4725762338436636e-03, -3.9967784806535761e-04,
       -1.1362435835439808e+00 ]
```

Şekil 4.3. Sol Kamera Matrisi

```
%YAML:1.0
---
K: !!opencv-matrix
   rows: 3
   cols: 3
   dt: d
   data: [ 6.4953257394642299e+02, 0., 3.1007277457722569e+02, 0.,
       6.5020248176278869e+02, 2.4645928008411408e+02, 0., 0., 1. ]
D: !!opencv-matrix
   rows: 1
   cols: 5
   dt: d
   data: [ -4.3463180105729478e-02, 5.1107895555199911e-01,
       -8.6563249604373882e-04, -2.4190056294415394e-03,
       -1.3472817256105363e+00 ]
```

Şekil 4.4. Sağ Kamera Matrisi

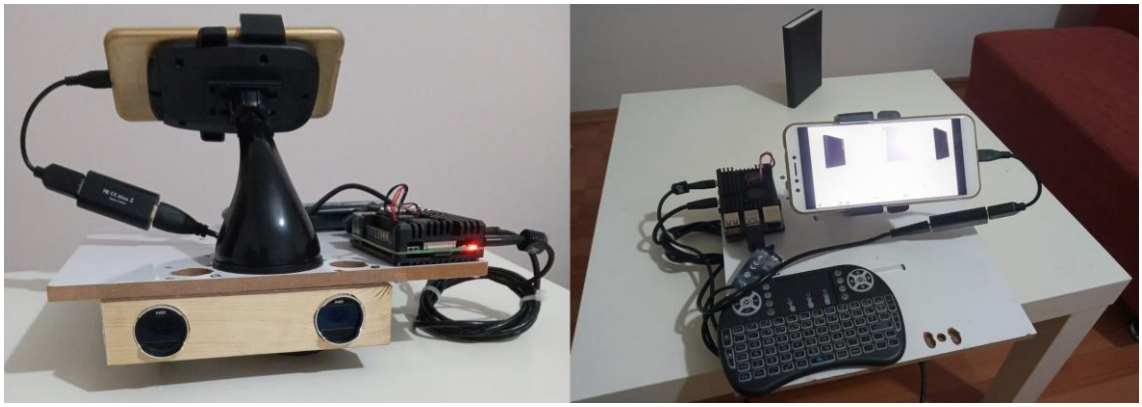
K: Kamera içsel matrisidir. f_x, f_y, c_x ve c_y değerleri bu matriste bulunur.

D: Bozulma katsayıları. k1, k2, p1, p2 ve k3 değerleri bu matriste bulunur.



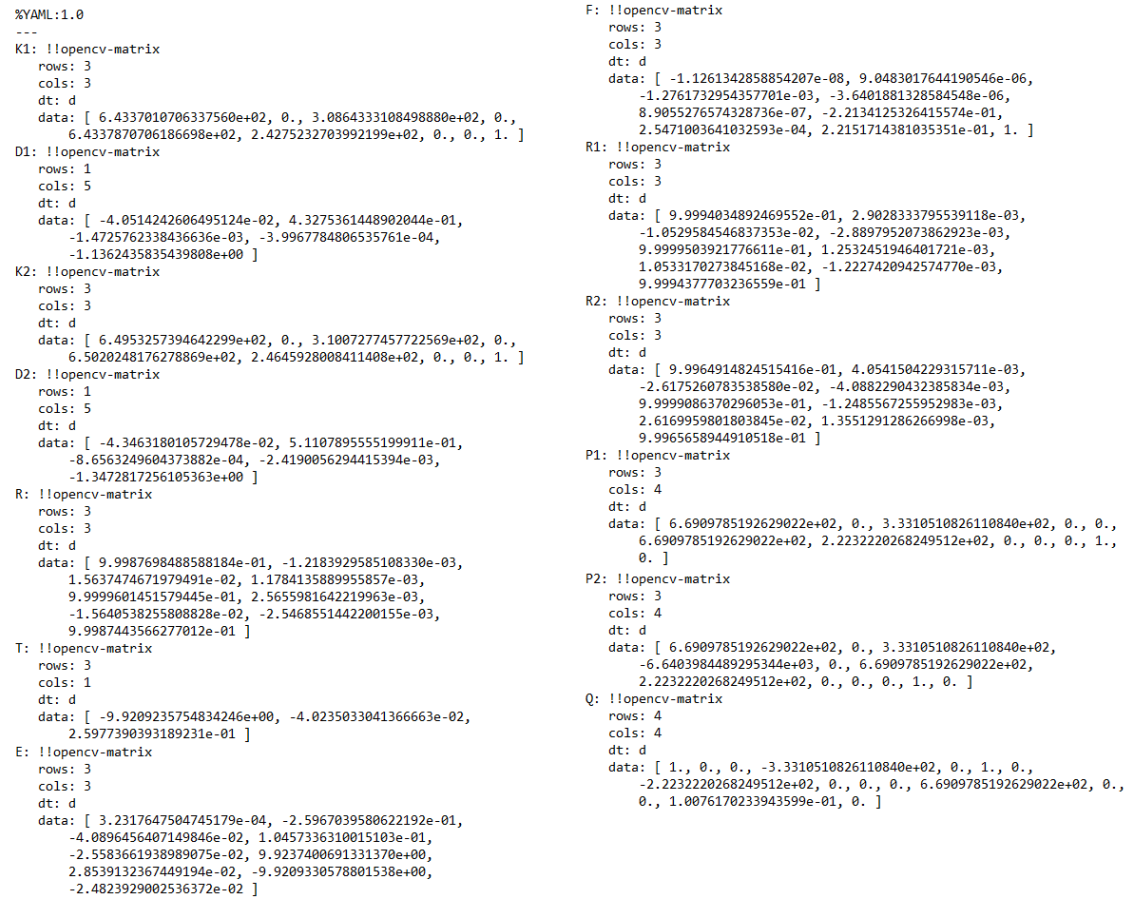
Şekil 4.5. Kalibrasyonsuz Ve Kalibrasyonlu Görüntüler [29]

Çalışmanın devamında bu iki kamera kullanılarak taşınabilir bir stereo düzenek hazırlanmıştır. Düzeneğin taşınabilir olması için raspbian 5.10 versiyonu işletim sistemine sahip bir raspberry pi 4 donanımı kullanılmıştır. Şekil 4.6'da hazırlnan bu düzenek görülmektedir.



Şekil 4.6. Görüntü Elde Etmek İçin Hazırlanan Stereo Düzenek

Stereo mesafe hesabının yapılabilmesi için stereo düzeneğinin de kalibre edilmesi ve stereo kalibrasyon matrisinin elde edilmesi gerekir. Kameraları kalibre etmek için kullanılan satranç tahtası kullanılarak stereo düzenekde kalibre edilmiştir. Kalibrasyon sonucu elde edilen stereo kalibrasyon matrisleri şekil 4.7'de görülmektedir. 3x1 boyutunda olan T matrisindeki ilk değer, iki kameranın optik merkezleri arasındaki mesafeyi santimetre cinsinden verir ve stereo hesabı için gereklidir.



Şekil 4.7. Stereo Kalibrasyon Matrisleri

K1: Birinci kameranın iç parametreleridir.

D1: Birinci kameranın bozulma katsayılarıdır.

K2: İkinci kameranın iç parametreleridir.

D2: İkinci kameranın bozulma katsayılarıdır.

R: Çıktı dönüş matrisidir. Çeviri vektörü T ile bu matris, birinci kameranın koordinat sisteminde verilen noktaları ikinci kameranın koordinat sistemindeki noktalara getirir. Daha teknik bir ifadeyle, R ve T dizisi, birinci kameranın koordinat sisteminden ikinci kameranın koordinat sistemine bir temel değişikliği gerçekleştirir. R ve T vektörü beraber değerlendirilirse bu, ikinci kamera koordinat sistemine göre birinci kameranın konumuna eşdeğerdir.

T: Öteleme matrisidir. Birinci kamera ile ikinci kamera arasındaki öteleme değerlerini X, Y ve Z eksenleri üzerinden verir.

E: Essential matrisidir. Fiziksel uzayda iki kamerayı ilişkilendiren öteleme ve döndürme hakkında bilgi içerir.

F: Fundemental matrisidir. E matrisine ek olarak F matrisi her iki kameranın iç parametrelerle ilgili bilgilerini de değerlerine ekler, böylece iki kamerayı piksel koordinatlarında ilişkilendirir.

R1: İlk kamera için 3x3 rotasyon matrisidir. Bu matris, düzeltilmemiş ilk kameranın koordinat sisteminde verilen noktaları, düzeltilmiş ilk kameranın koordinat sistemindeki noktalara getirir. Daha teknik terimlerle, düzeltilmemiş ilk kameranın koordinat sisteminden düzeltilmiş ilk kameranın koordinat sistemine bir temel değişikliği gerçekleştirir.

R2: İkinci kamera için çıktı 3x3 rotasyon matrisidir. Bu matris, düzeltilmemiş ikinci kameranın koordinat sisteminde verilen noktaları, düzeltilmiş ikinci kameranın koordinat sistemindeki noktalara getirir. Daha teknik terimlerle, düzeltilmemiş ikinci kameranın koordinat sisteminden düzeltilmiş ikinci kameranın koordinat sistemine bir temel değişikliği gerçekleştirir.

P1: İlk kamera için yeni (düzeltilmiş) koordinat sistemlerinde 3x4 projeksiyon matrisi çıktısıdır. Düzeltilmiş birinci kamera koordinat sisteminde verilen noktaları düzeltilmiş ilk kameranın görüntüsüne yansıtır.

P2: İkinci kamera için yeni (düzeltilmiş) koordinat sistemlerinde 3x4 projeksiyon matrisi çıktısıdır. Düzeltilmiş birinci kamera koordinat sisteminde verilen noktaları düzeltilmiş ikinci kameranın görüntüsüne yansıtır.

Q: Disparity değerleri üzerinden derinlik haritasına geçilmesini sağlar.

4.1. Veri Hazırlama

Çalışmanın devamında İstanbul'un çeşitli köylerine gidilerek, hazırlanan stereo düzenek ile 640x480 piksel boyutlarında, 85 adet büyükbaş hayvan fotoğrafı çekilmiştir.

Çalışmanın planlama aşamasında tek bir hayvanı önden, yandan, arkadan ve üstten olmak üzere 4 farklı açıdan çekmek planlamış fakat sahada karşılaşılan zorluklar sebebi ile çekim açısı yandan ve arkadan olmak üzere ikiye düşürülmüştür. Stereo düzenek, kullanılan batarya sebebiyle yakşalık 1 kg olduğu için hayvanları üstten çekmek mümkün olmamıştır. Önden çekimler ise hayvanların birbirlerinden çok farklı duruşlar sergilemesi ve aynı zamanda atlatılan birkaç tehlike sebebi ile iptal edilmiştir. Sonuç olarak stereo düzenek ile 85 adet hayvanın iki farklı açıdan fotoğrafları çekilmiş ve toplamda 170 çift olmak üzere 340 adet stereo fotoğraf elde edilmiştir. Bu stereo fotoğraf çiftlerinden bazıları şekil 4.8'de görülmektedir.



Şekil 4.8. Rektifikasyonsuz ve Rektifikasyonlu Stereo Fotoğraflar

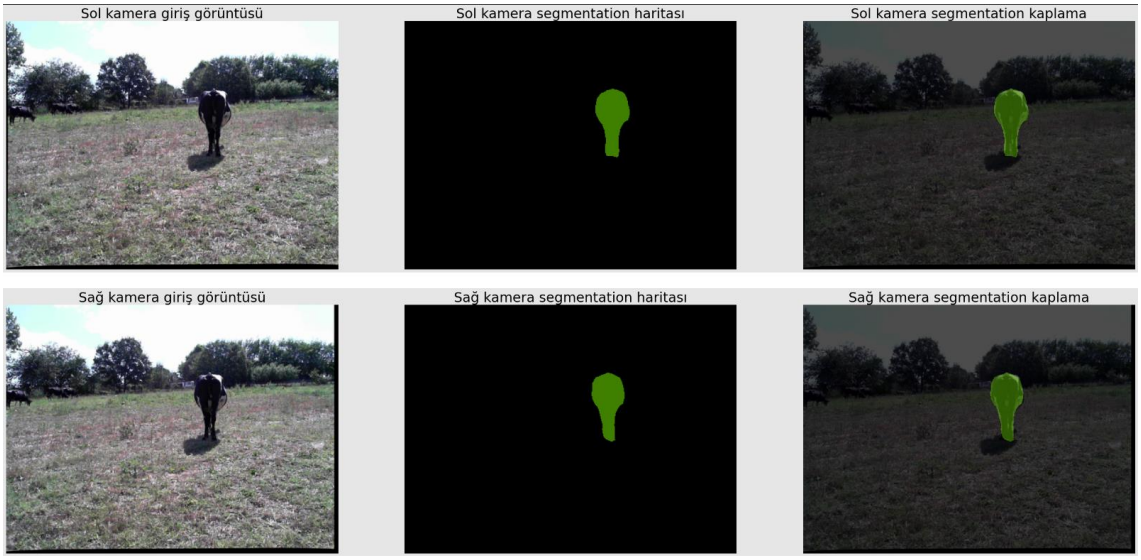
Şekil 4.8'deki fotoğraflardan a ve b sütünlarında bulunanlar kameralardan gelen ve herhangi bir rektifikasyon işlemine girmemiş ham fotoğraflardır. Stereo hesabının doğru şekilde yapılabilmesi için bu fotoğrafların, stereo kalibrasyon matrisleri ve openCV

kütüphanesi kullanılarak stereo düzeltme(rektifikasyon) işlemine sokulması gerekir. Şekilde c ve d sütünlarında ise aynı fotoğrafların rektifikasyon işlemi yapılarak düzeltişmiş halleri görülmektedir.

Çalışmanın segmentasyon adımında ise bölüm 3.5.4.'de açıklanan deeplab v3+ semantik segmentasyon modeli kullanılarak rektifiye edilmiş fotoğrafların tamamının segmentasyon çıktıları elde edilmiştir. Şekil 4.9 ve 4.10'da örnek segmentasyon sonuçları görülmektedir.



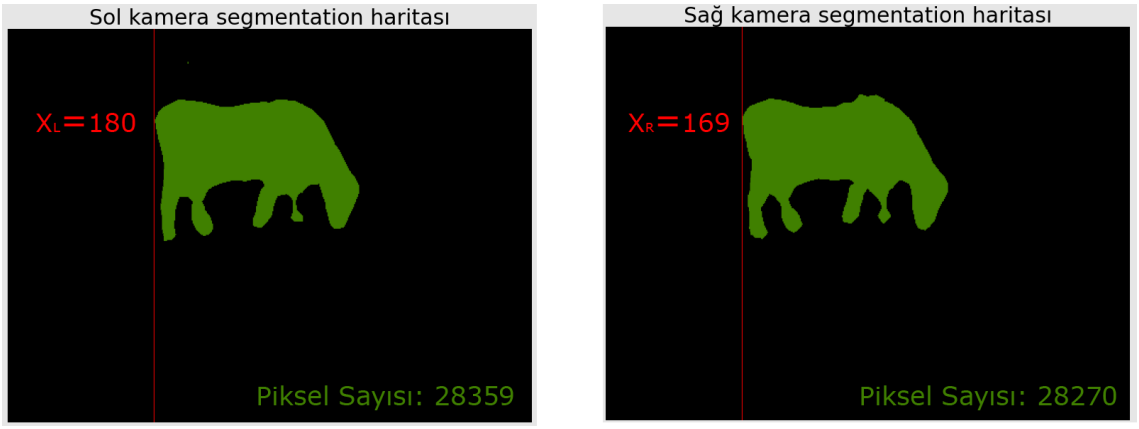
Şekil 4.9. Yandan Çekilmiş Stereo Görüntülerde Segmentasyon Sonuçları



Şekil 4.10. Arkadan Çelikmiş Stereo Görüntülerde Segmentasyon Sonuçları

Tüm segmentasyon işlemlerini tamamladıktan sonra her bir hayvan için sol ve sağ kameralardan gelen görüntülerin segmentasyon haritalarında, hayvanların kapladığı piksel sayılarını hesaplanmıştır. Yine bu segmentasyon haritaları kullanılarak her bir hayvanın kamera düzlemine olan mesafesi stereo hesaplama tekniği ile hesaplanmıştır. Tüm bu işlemler python programlama dilinde yazılan kaynak kodları ile tüm resimler için topluca otomatik olarak gerçekleştirilmiştir.

Segmentasyon haritaları üzerinden hayvanın kamera düzlemine olan mesafesini hesaplamak için görüntü çiftlerinin her birinde hayvanın sol sınırının piksel cinsinden X konumu belirlenmiştir. Sol kameradan gelen segmentasyon haritasındaki sınır değeri ile sağ kameradan gelen segmentasyon haritasındaki sınır değeri birbirinden çıkarılarak bölüm 3.2.'de ifade edilen disparity(d) değeri hesaplanmıştır.



Şekil 4.11. Disparity Değeri ve Hayvanın Görüntüdeki Piksel Sayıları

Şekil 4.11'de görüldüğü gibi X_L ve X_R değerlerini birbirinden çıkarılırsa disparity(d) değeri 11 piksel olarak bulunur. Bu değerle beraber, stereo kalibrasyon matrislerinden gelen odak uzaklığı(f) ve öteleme matrisinden gelen kameralar arasındaki öteleme değeri(T) kullanılarak mesafe değeri kolayca hesaplanabilir. Şekil 4.7'deki stereo kalibrasyon matrisleri incelenirse sol kamera için f değerinin X cinsinden 643.37010706337560 piksel, Y cinsinden ise 643.37870706186698 piksel olduğu görülür. Sağ kamera için ise f değerleri sırasıyla 649.53257394642299 ve 650.20248176278869'dir. Odak uzaklığının X ve Y cinsinden birbirinden farklı olmasının sebebi bölüm 3.1.'de ilgili denklemlerle beraber ortaya konularak ifade edilmiştir. Çalışmada sadece yatay eksende hesaplama yapılacağı için X eksenindeki odak uzaklıkları yeterli olacaktır. Her iki kamera için ortak bir odak uzaklığı değeri elde etmek için her iki kameranın X cinsinden odak uzaklıklarının ortalaması alınmış ve bu değer 646.451340504899295 olarak bulunmuştur. T öteleme değeri, şekil 4.7'deki stereo kalibrasyon matrislerinde görüldüğü gibi -9.9209235754834246'dir. Bu değer f değerinden farklı olarak santimetre cinsindendir. Negatif olmasının sebebi kameraların konumlarının birbirlerine göre negatif olarak kabul edilmesidir. Denklemlerde bu değer mutlak değer olarak kullanılmıştır.

$$Z = \frac{fT}{d} \rightarrow \frac{646.451340504899295 * 9.9209235754834246}{11} = 583.035 \qquad (4.1)$$

Denklem 4.1'de yapılan hesaplamaya göre şekil 4.11'de gösterilen örnek hayvanın kamera düzlemine olan mesafesi 583 santimetredir.

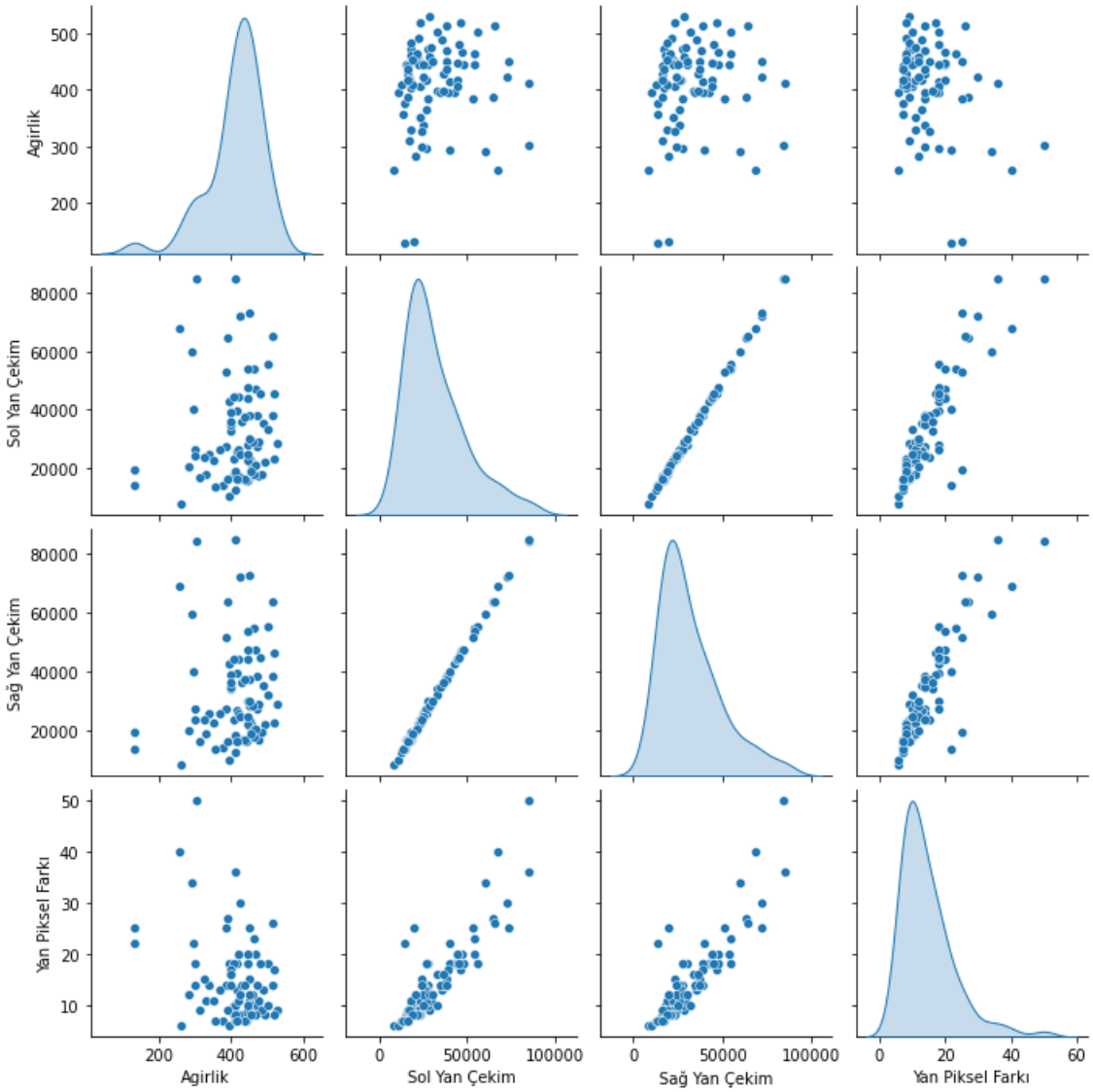
Her bir hayvan için mesafe değerlerini hesapladıktan sonra hayvanların görüntülerde kapladığı piksel sayısını da belirlenmiş ve tüm bu değerler kullanılarak tablo 4.1'de görüldüğü gibi her bir hayvan için bir satır olmak üzere toplamda 85 hayvan için 85 satırdan oluşan bir veri seti oluşturulmuştur. Tabloda son 10 satır görülmektedir. Mesafe değerleri metre cinsinden yazılmıştır.

Tablo 4.1. Stereo Görüntüleri Kullanarak Oluşturduğumuz Veriseti

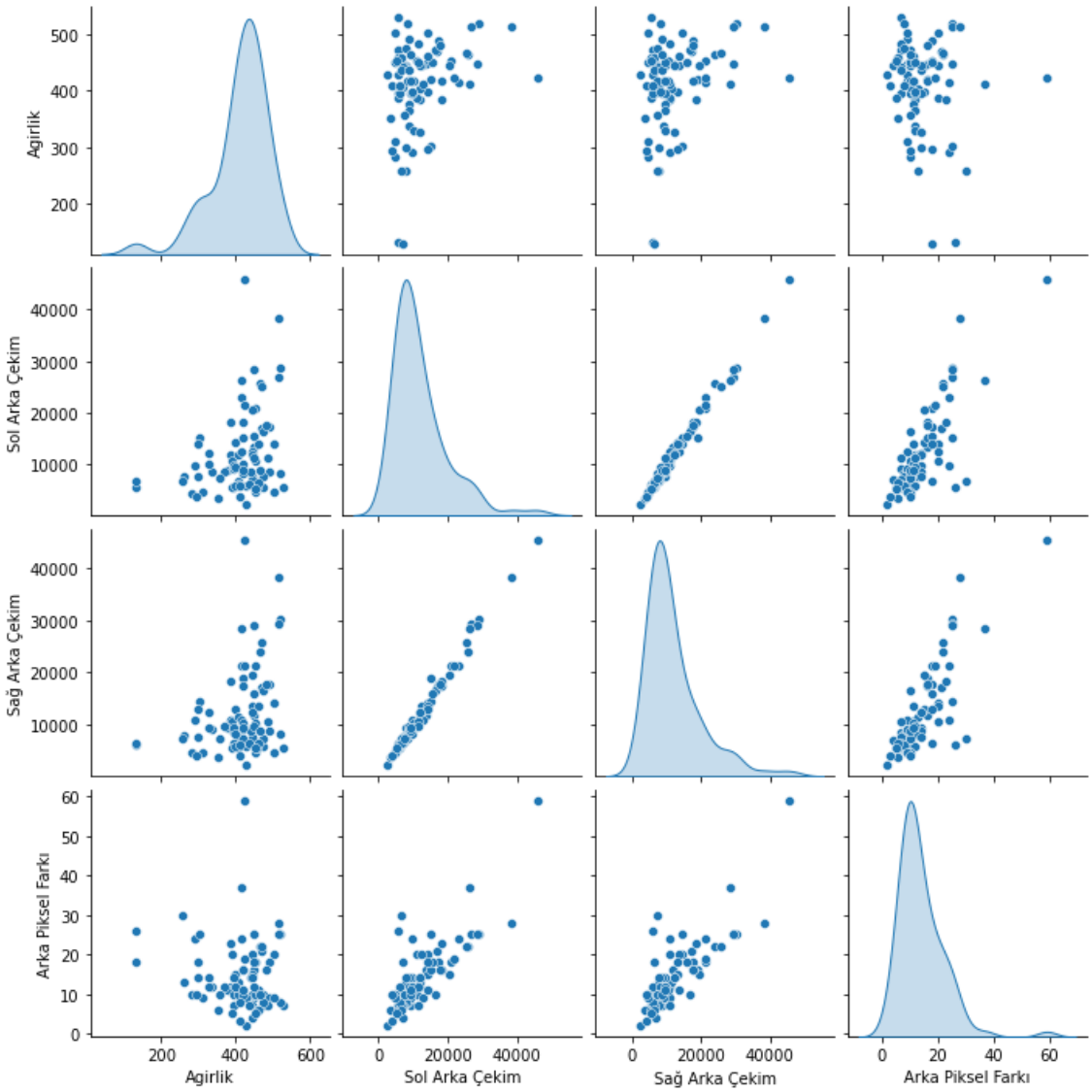
id	Sol Yan Çekim	Sağ Yan Çekim	Yan Piksel Farkı	Yan Mesafe	Sol Arka Çekim	Sağ Arka Çekim	Arka Piksel Farkı	Arka Mesafe	Agirlik
76	20346	19837	12	5.344484204	4541	4541	10	6.413381045	283
77	35896	36164	16	4.008363153	9569	8195	12	5.344484204	398
78	30093	30029	12	5.344484204	11402	9606	12	5.344484204	450
79	24082	23725	14	4.580986461	7662	7535	14	4.580986461	300
80	16450	16536	9	7.125978939	4717	4537	9	7.125978939	312
81	13613	13582	7	9.161973	7379	7372	10	6.413381	357
82	40195	40141	22	2.915173	3831	3852	10	6.413381	294
83	21014	20703	8	8.016726	9017	8616	10	6.413381	465
84	18571	18758	8	8.016726	5314	5374	5	12.826762	453
85	16099	16051	7	9.161973	9067	9352	11	5.830346	417

Tablo 4.1'de sol yan çekim sütununda sol kameradan gelen rektifikasyonlu görüntüde hayvanın kapladığı alanın piksel sayısı, sağ yan çekim sütununda ise hayvanın sağ kameradan gelen rektifikasyonlu görüntüde kapladığı alanın piksel sayısı verilmiştir. Yan piksel farkı sütununda ise sol ve sağ kameralardan gelen görüntülerde hayvanın görüntülerdeki piksel farkı yani şekil 4.11'de görüldüğü gibi disparity değeri verilmiştir. Yan mesafe sütunu da bu piksel farkı kullanılarak yapılan stereo hesabı sonucunda hayvanın kamera düzlemine olan uzaklığı metre cinsinden verilmiştir. Sol arka çekim, sağ arka çekim, arka piksel farkı ve arka mesafe değerleri ise hayvanın arkadan çekilmiş görüntüleri kullanılarak aynen yandan çekilmiş görüntülerde olduğu gibi hesaplanmıştır.

Şekil 4.12 ve 4.13'de veriseti grafiksel olarak ifade edilmiştir. Yandan ve arkadan çekimler birbirinden bağımsız olduğu için her iki çekim sonuçları farklı grafikler ile gösterilmiştir.



Şekil 4.12. Yandan Çekim Verileri



Şekil 4.13. Arkadan Çekim Verileri

Şekil 4.12'de verisetinin sol yan çekim, sağ yan çekim, yan piksel farkı ve ağırlık verileri görselleştirilmiştir. Aynı görselleştirme şekil 4.13'de arkadan çekilen fotoğraf verileri için yapılmıştır. Stereo mesafe değeri zaten piksel farkı kullanılarak hesaplandığı için tekrar bu grafiklere koyma gereği duyulmamıştır. Burada dikkat edilmesi gereken hayvanın görüntülerde kapladığı piksel sayısı, yani segmentasyon verisinin tek başına bir anlam ifade etmemesidir. Yani bir hayvan hafif ağırlıkta olsa bile eğer kamera düzlemine yakın bir yerden fotoğraflanmışsa görüntüde çok yer kaplayacaktır. Fakat denkleme stereo mesafe değişkeninini de katarsak o zaman değerlerimiz anlamlı olacaktır. Hayvanın görüntüde kapladığı piksel sayısı ile beraber kamera düzlemine olan mesafesi

de fazla ise ozaman o hayvanın ağırlığının fazla olduğu söylenebilir. Dolayısı ile şekil 4.12 ve 4.13'deki grafikler tek başına ağırlık ile ilgili bir fikir edinmemizi sağlayamaz. Sadece en sağdaki sütunlarda ikinci ve üçüncü satırlara bakarsak piksel farkı ile sol ve sağ çekimlerdeki piksel sayısının doğru orantılı olduğu görülür. Buda gayet doğaldır çünkü herhangi bir cisim stereo kamera düzlemine yakın fotoğraflanırsa her iki görüntü çiftinde çok yer kaplar ve bu iki görüntüdeki piksel farkıda yüksek olur. Bölüm 3.2.'de yer alan şekil 3.9 incelenirse kamera düzlemine yakın nesnelerin piksel farkının yani disparity değerinin fazla, uzak olan nesnelerin ise disparity değerinin az olduğu görülür.

Çalışma, giriş bölümünde de bahsedildiği gibi segmentasyon verisi ile beraber stereo mesafe verisini beraber kullanarak ağırlık tahmini yapmak üzerinedir. Dolayısı ile her bir hayvandan elde edilen bu iki veri tek bir fonksiyonda bir araya getirilmiştir. Bu fonksiyon kullanılarak veri seti daha anlamlı hale getirilebilir. Yan ve arka değerler için ayrı ayrı olmak üzere, veri setinde olmayan, yan değer ve arka değer isminde iki adet fonksiyon tanımlanmıştır. Bu fonksiyonlar, hayvanın görüntülerde kapladığı piksel sayısı ile aynı hayvanın stereo mesafesini hesaplamak için kullanılan disparity değerlerini beraber içermelidir.

$$yan\ Değer = \frac{sol\ yan\ çekim + sağ\ yan\ çekim}{yan\ piksel\ farkı} \tag{4.2}$$

$$arka\ Değer = \frac{sol\ arka\ çekim + sağ\ arka\ çekim}{arka\ piksel\ farkı} \tag{4.3}$$

Denklem 4.2.'de tanımlanan yan değer isimli değişken yandan çekilmiş görüntü çiftindeki piksel sayılarının toplamının disparity değerine bölünmesi ile elde edilmiştir. Bunun nedeni, piksel sayılarının ağırlık ile doğru orantılı, disparity değerinin ise ağırlık ile ters orantılı olmasıdır. Disparity değerinin yükselmesi demek hayvanın kamera düzlemine yakın bir noktadan fotoğraflandığı anlamına gelmektedir. Bu sebeple disparity

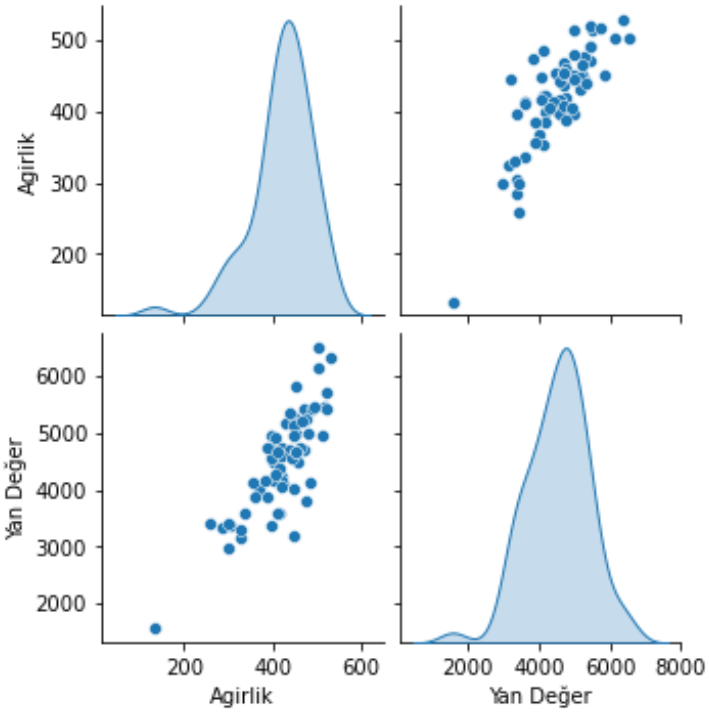
arttıkça piksel sayısı sabit kalsa bile hayvanın daha hafif bir ağırlığa sahip olması gerekir. Aynı işlemler denklem 4.3.'de arka değer fonksiyonu içinde geçerlidir.

Yan ve arka değeri her bir hayvan için hesaplanarak veri seti güncellenmiştir.

Tablo 4.2. Güncellenmiş Veri Seti

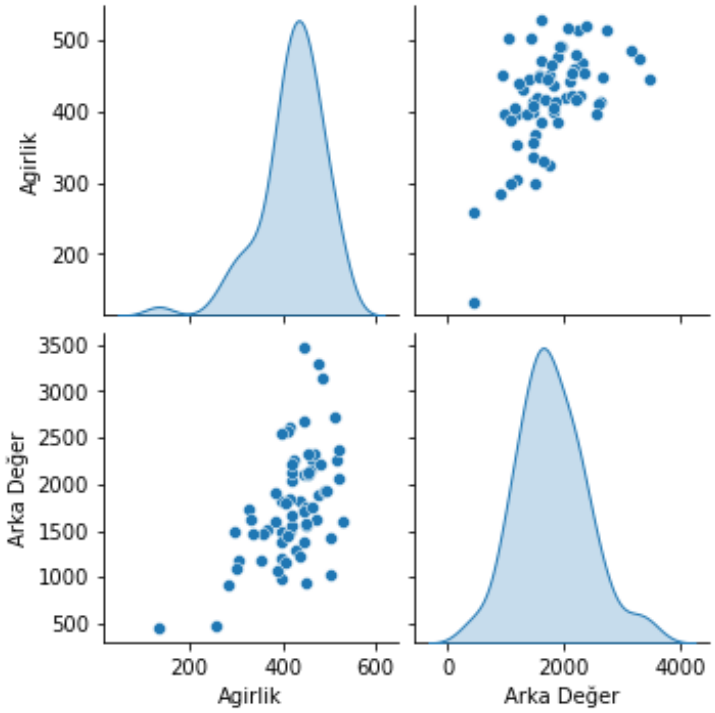
id	Sol Yan Çekim	Sağ Yan Çekim	Yan Piksel Farkı	Yan Mesafe	Sol Arka Çekim	Sağ Arka Çekim	Arka Piksel Farkı	Arka Mesafe	Agirlik	Yan Deger	Arka Değer
76	20346	19837	12	5.344404	4541	4541	10	6.4181045	283	3348.58	908.2
77	35896	36164	16	4.008363	9569	8195	12	5.3444204	398	4503.75	1480.333
78	30093	30029	12	5.344204	11402	9606	12	5.3444204	450	5010.166	1750.667
79	24082	23725	14	4.580981	7662	7535	14	4.5809461	300	3414.784	1085.5
80	16450	16536	9	7.125979	4717	4537	9	7.1259939	312	3665.111	1028.22
81	13613	13582	7	9.16197	7379	7372	10	6.413381	357	3885	1475.1
82	40195	40141	22	2.91517	3831	3852	10	6.413381	294	3651.636	768.3
83	21014	20703	8	8.01673	9017	8616	10	6.413381	465	5214.625	1763.3
84	18571	18758	8	8.01673	5314	5374	5	12.82676	453	4666.125	2137.6
85	16099	16051	7	9.16197	9067	9352	11	5.830346	417	4592.857	1674.45

Şekil 4.14 ve 4.15'de, oluşturulan bu yan değer ve arka değer fonksiyonlarının ürettikleri değerlerin hayvanların ağırlıkları ile olan ilişkileri ayrı ayrı görselleştirilmiştir.



Şekil 4.14. Yan Değer Ve Ağırlık İlişkisi

Şekil 4.14'de yer alan grafikde görüldüğü gibi yan değerler ağırlık ile doğrudan bağlantılıdır. Sadece bu değerler kullanılarak bile stereo fotoğrafı çekilmiş bir hayvan için ağırlık tahmini yapılabilir. Çalışmada, hem bu şekildeki sınırlı bilgiler ile hem de tüm veri seti kullanılarak yapay sinir ağları eğitilmiş ve bu ağlarla tahminler yapılarak sonuçları karşılaştırılmıştır.



Şekil 4.15. Arka Değer Ve Ağırlık İlişkisi

Şekil 4.15'de arka değer ve ağırlık arasındaki ilişkiyi ifade eden grafik bulunmaktadır. Şekil 4.14 ve 4.15'deki grafikler karşılaştırılırsa yan değerin ağırlık ile olan ilişkisinin arka değere göre daha tutarlı olduğu görülür. Bunun sebebi bir hayvanın yandan çekim görüntüsünün onun vucüt yapısını daha iyi ortaya koymasıdır. Arkadan çekilen fotoğraflarda ise hayvanın ağırlık tahmini yandan çekimlere göre daha zayıf kalacaktır.

4.2. Veri Normalizasyonu

Hazırlanan veri seti incelenirse birbirinden çok farklı ölçeklerde verilerin olduğu görülür. Görüntüdeki piksel sayıları binlerle ifade edilirken, stereo mesafeler birkaç metre ile, disparity değerleri ise 5-30 piksel aralığında ifade edilmektedir. Bu şekildeki girdilerle bir ağın eğitilmesi hem çok zaman alabilir hemde ağ yeterince başarıya ulaşamayabilir. Bu tür farklı ölçeklerdeki verilerin, normalizasyon denilen bir takım tekniklerle işlenerek birbirlerine yakın değerler olarak ifade edilmeleri gerekir. Bunun temel nedeni, bu özelliklerin model ağırlıklarıyla çarpılmasıdır. Dolayısıyla çıktıların ve gradyanların ölçeği, girdilerin ölçeğinden etkilenir. Literatürde kullanılan farklı

normalizasyon teknikleri mevcuttur. Bu çalışmada tensorflow kütüphanesinde hazır olarak bulunan normalization fonksiyonu ile veri normalleştirilmesi yapılmıştır. Tablo 4.3'de pembe renkli satıralarda ham veriler, beyaz renkli satırlarda ise bir üstte verilmiş ham verilerin normalize edilmiş halleri görülmektedir.

Tablo 4.3. Örnek Normalizasyon Sonuçları

Sol Yan Çekim	Sağ Yan Çekim	Yan Piksel Farkı	Yan Mesafe	Sol Arka Çekim	Sağ Arka Çekim	Arka Piksel Farkı	Arka Mesafe
53850	54433	23	2.79	25761	23903	22	2.92
1.4	1.4	1.16	-1.26	2.06	1.76	1.25	0.77
37897	38291	14	4.58	16978	17075	21	3.05
0.4	0.42	-0.01	-0.45	0.78	0.78	1.11	-0.74
24641	24474	12	5.34	21629	21273	19	3.38
-0.4	-0.41	-0.27	-0.11	1.45	1.38	0.81	-0.67
47105	47196	20	3.21	25168	25835	22	2.92
0.95	0.96	0.77	-1.07	1.97	2.03	1.25	-0.77
27869	29934	18	3.56	7089	6826	4	16.03
-0.21	-0.08	0.51	-0.91	-0.67	-0.67	-1.41	2.22

4.3. Yapay Sinir Ağlarının Tasarımı ve Eğitimi

Bu çalışmada üç adet farklı yapay sinir ağı tasarlanıp eğitilmiştir. Birinci ağ sadece sol yan piksel, sağ yan piksel ve yan piksel farkı değerleri ile eğitip tahminleme yapılmıştır. İkinci ağ sol arka piksel, sağ arka piksel ve arka piksel farkı değerleri ile eğitip tahminleme yapılmıştır. Son ağ ise veri setimizindeki tüm değerler ile eğitip tahminleme yapılmıştır.

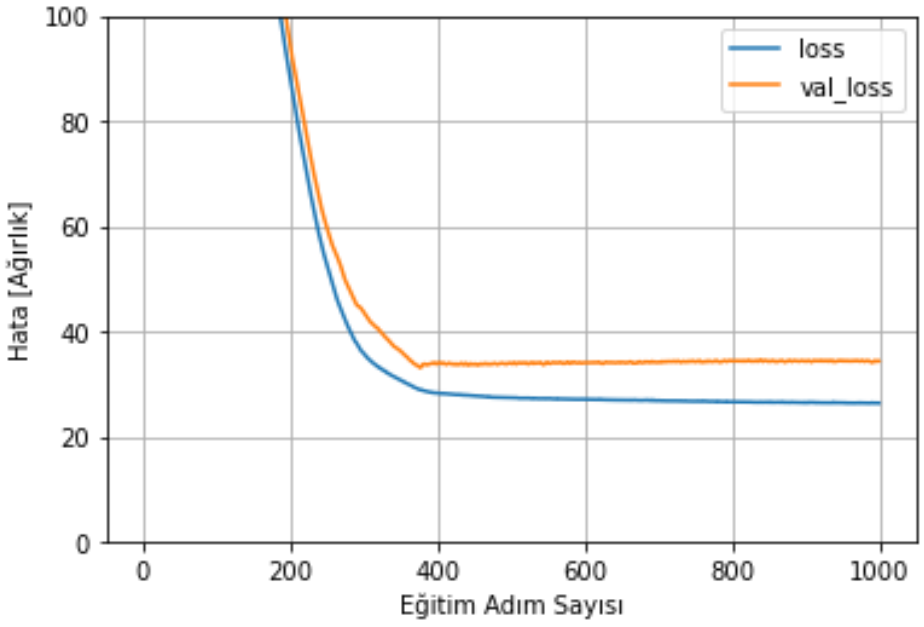
Veri setinin %90'ı yapay sinir ağlarını eğitmek için, %10'u ise test için ayrılmıştır. Doğrulama verileri ise %90'lık eğitim verisinin %20'lik kısmından oluşmuştur.

İlk ağ 3 elemanlı giriş katmanına, devamında 64 node'dan oluşan 2 adet gizli katmana ve 1 elemandan oluşan çıkış katmanına sahip bir tam bağlı ağdır. Toplamda 4488 parametreye sahiptir.

Tablo 4.4. Birinci Yapay Sinir Ağının Mimari Özeti

```
Model: "sequential_3"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
normalization (Normalization (None, 3)                 7.
_________________________________________________________________
dense_5 (Dense)              (None, 64)                256
_________________________________________________________________
dense_6 (Dense)              (None, 64)                4160
_________________________________________________________________
dense_7 (Dense)              (None, 1)                 65
=================================================================
Total params: 4,488
Trainable params: 4,481
Non-trainable params: 7
```

İlk ağın gizli katmanlarında aktivasyon fonksiyonu olarak ReLU fonksiyonu kullanılmıştır. Loss fonksiyonu olarak ortalama mutlak hata fonksiyonu, optimizer olarak keras kütüphanesinde hazır bulunan Adam optimizeri, öğrenme değeri olarak 0.001 değeri kullanılmıştır. Eğitim adım sayısı 1000 olarak yeterli görülmüştür. Bu parametreler ile alınan sonuçlar ışığında oluşturulan kayıp grafiği şekil 4.16'da görülmektedir.



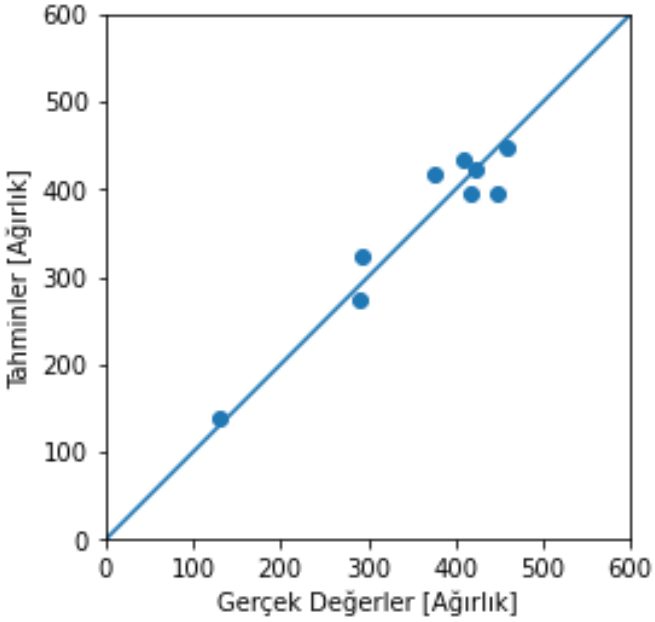
Şekil 4.16. Birinci Yapay Sinir Ağının Kayıp Grafiği

Sadece yandan çekilen fotoğraflar ile elde edilen değerlerle eğitilen ilk ağda Şekil 4.16'da görüldüğü gibi kayıp ve doğrulama kaybı değerleri yaklaşık 30 civarındadır. Bu ağın test seti üzerinde gerçekleştirdiği tahminler tablo 4.5'de görülmektedir.

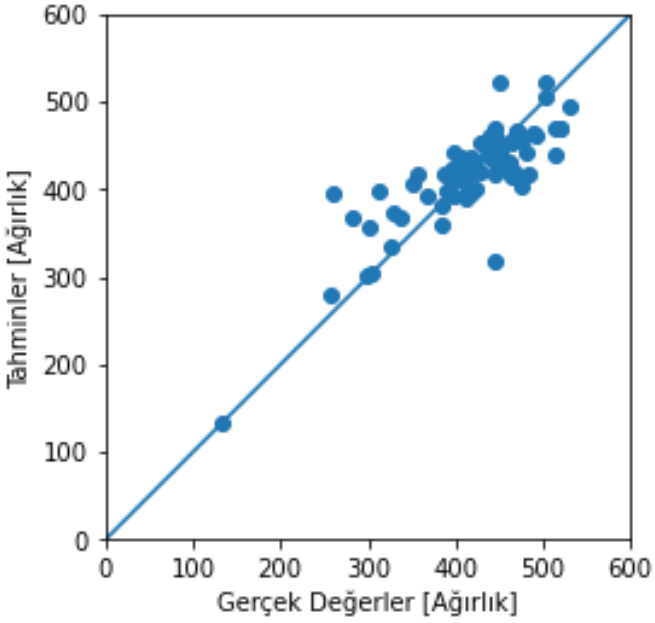
Tablo 4.5. Birinci Ağın Tahmin Değerleri

id	Sol Yan Çekim	Sağ Yan Çekim	Yan Piksel Farkı	Gerçek Agirlik	Tahmin Edilen Ağırlık
9	60146	59403	34	291	275.083
21	14031	14031	7	376	418.701
36	72247	71788	30	423	421.776
44	44624	44198	20	418	396.788
47	43968	44156	20	447	394.972
64	20079	19978	8	458	446.577
67	18592	18507	8	409	434.73
70	13850	13713	22	131	139.36
81	40195	40141	22	294	324.351

Tablo 4.5'de görüldüğü gibi sadece yandan çekilmiş fotoğraflarla eğitilen ağın test seti üzerinde yaptığı tahminlerde hata miktarları genel olarak 20-40 kilogram arasında değişmektedir.

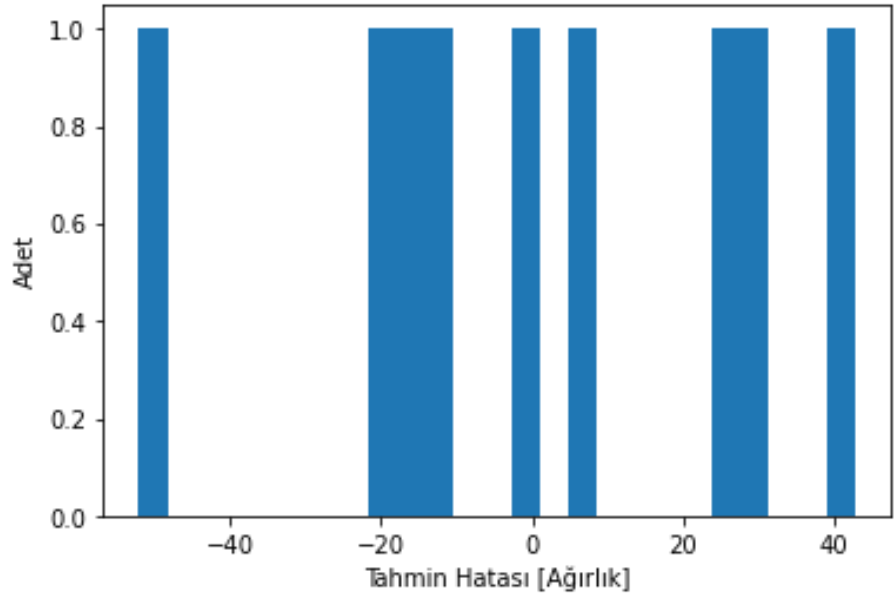


Şekil 4.17. Birinci Ağın Test Seti Üzerinde Yaptığı Tahminler

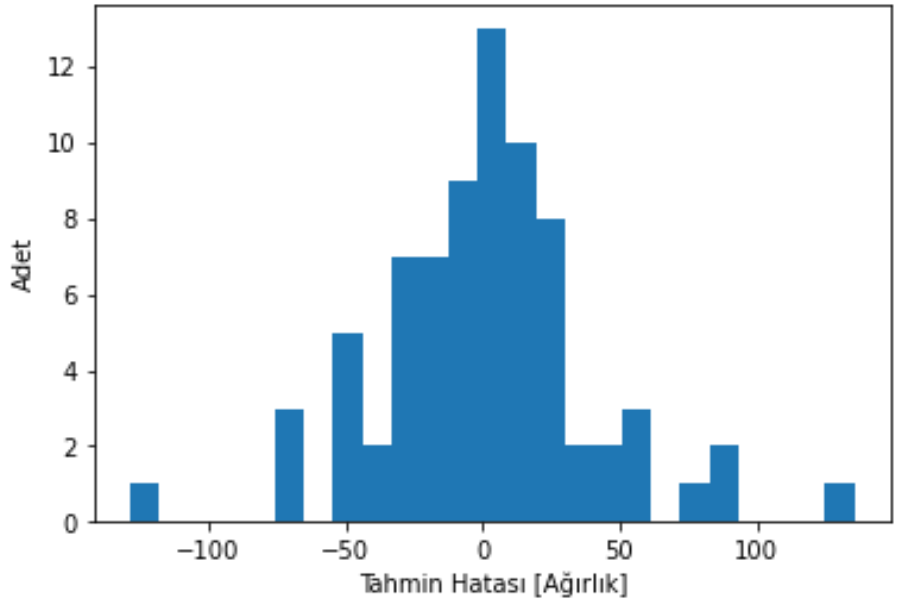


Şekil 4.18. Birinci Ağın Eğitim Seti Üzerinde Yaptığı Tahminler

Şekil 4.18'de gerçek değerler ve tahmin edilen değerler eğitim seti için grafiksel olarak gösterilmiştir.



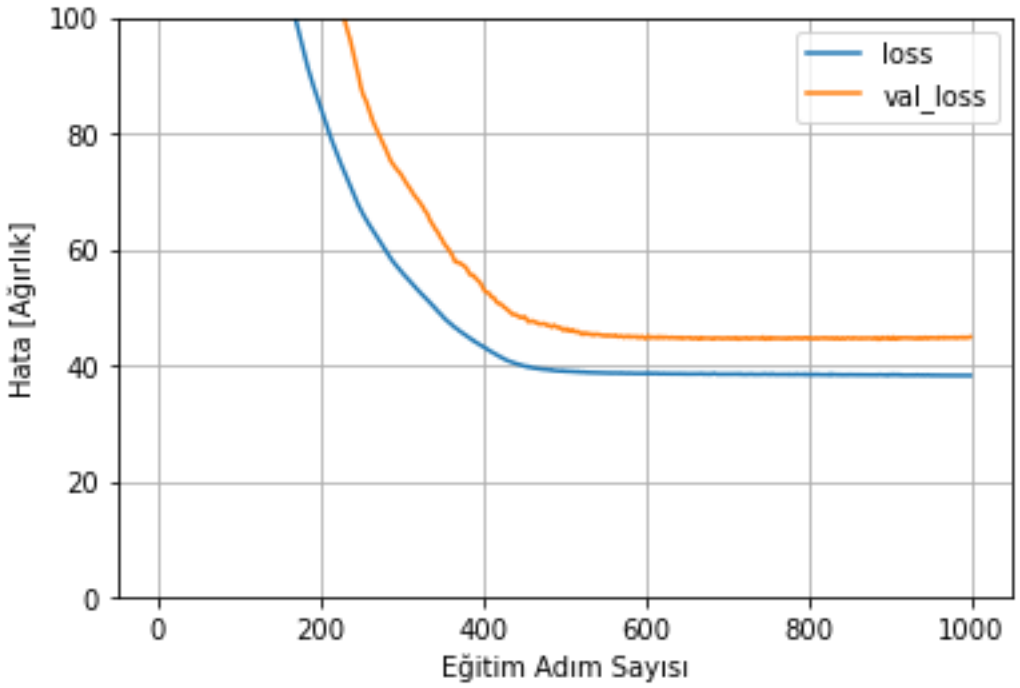
Şekil 4.19. Birinci Ağın Test Seti Üzerinde Hata Dağılımı Grafiği



Şekil 4.20. Birinci Ağın Eğitim Seti Üzerinde Hata Dağılımı Grafiği

Şekil 4.20'de görüldüğü gibi sadece yandan çekilmiş fotoğraflardan elde edilen verilerle eğitilen ağın yaptığı tahminler genel olarak ± 50 kilogram arasında değişmektedir.

Çalışmanın devamında ikinci bir yapay sinir ağı tasarlanmış ve bu ağda arkadan çekilmiş fotoğraflar ile eğitilerek sonuçlar karşılaştırılmıştır. İkinci ağ mimari olarak ilk ağ ile aynıdır. Hiper parametreler de olduğu gibi korunmuştur. Böylece parametre sayısı da eşit kalmıştır. Bu sebeple mimarinin özet tablosu tekrar buraya konulmamıştır. Bu şartlar altında arkadan çekilmiş fotoğraflardan elde ettiğimiz verilerle eğittimiz ağın başarımı şekil 4.21'deki grafikte görülmektedir.



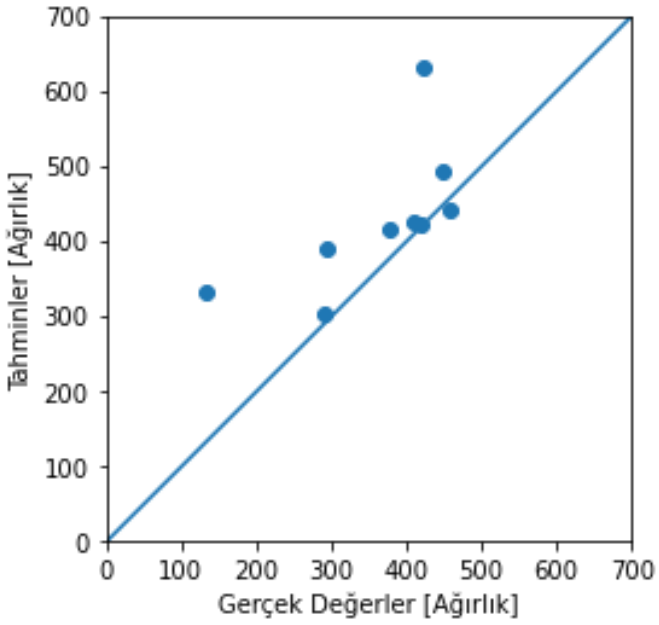
Şekil 4.21. İkinci Yapay Sinir Ağının Kayıp Grafiği

Şekil 4.21'de görüldüğü gibi sadece arkadan çekilmiş fotoğraflardan elde edilen verilerle eğitilen ağda hata miktarları 40 civarındadır. Bu değer birinci ağa göre daha fazladır. Bunun sebebi sadece arkadan çekilmiş fotoğrafların hayvanın genel vücut ölçülerini ortaya koyma noktasındaki yetersizliktir. Ağın test seti üzerinde gerçekleştirdiği tahminler tablo 4.6 üzerinde görülmektedir.

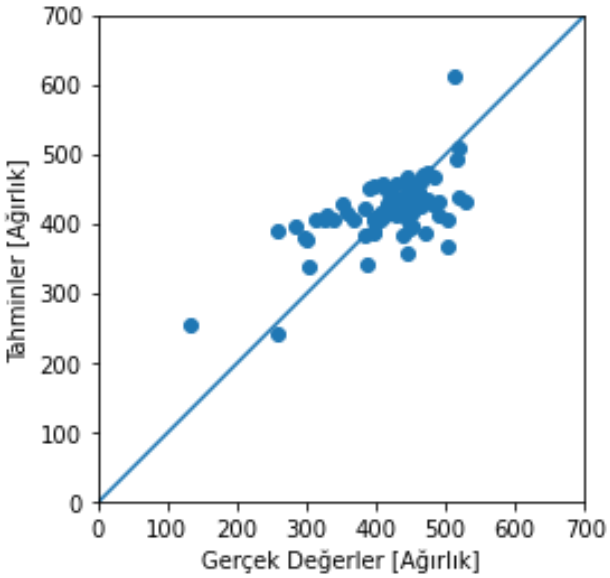
Tablo 4.6. İkinci Ağın Tahmin Değerleri

id	Sol Arka Çekim	Sağ Arka Çekim	Arka Piksel Farkı	Agirlik	Tahmin Edilen Ağırlık
9	9819	10676	24	291	302.667
21	8818	9227	11	376	415.957
36	45864	45521	59	423	632.537
44	10000	10126	11	418	422.559
47	28421	29072	25	447	492.185
64	5726	5492	6	458	441.392
67	6002	6056	8	409	425.119
70	6753	6239	18	131	332.303
81	3831	3852	10	294	391.844

Tablo 4.6'da görüldüğü gibi sadece arkadan çekilmiş fotoğraflarla eğitilen ağın yaptığı tahminlerde hata miktarları genel olarak 20-40 kilogram arasında değişmekle beraber 36, 70 ve 81 id numaralı hayvanlarda hata miktarları dramatik şekilde artmıştır. Bu durum, arkadan çekilmiş fotoğraflarla eğitilen ağın güvenirliğini önemli ölçüde düşürür.

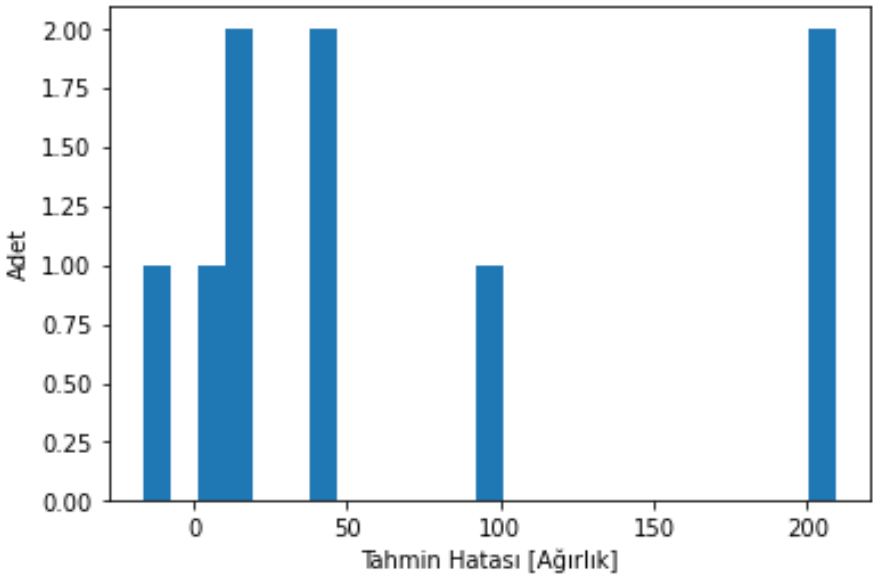


Şekil 4.22. İkinci Ağın Test Seti Üzerinde Yaptığı Tahminler

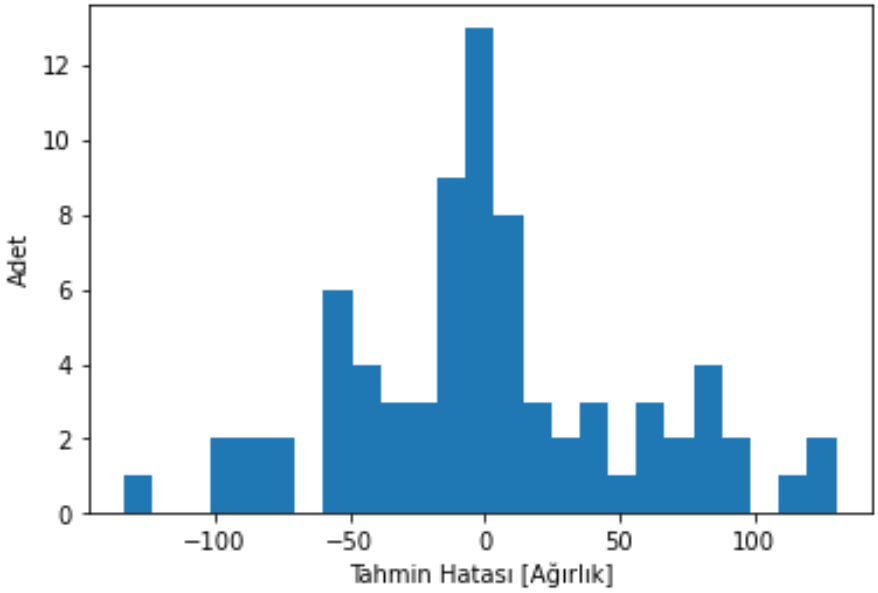


Şekil 4.23. İkinci Ağın Eğitim Seti Üzerinde Yaptığı Tahminler

Şekil 4.23'de gerçek değerler ve tahmin edilen değerler eğitim seti için grafiksel olarak gösterilmiştir.



Şekil 4.24. İkinci Ağın Test Seti Üzerinde Hata Dağılımı Grafiği



Şekil 4.25. İkinci Ağın Eğitim Seti Üzerinde Hata Dağılımı Grafiği

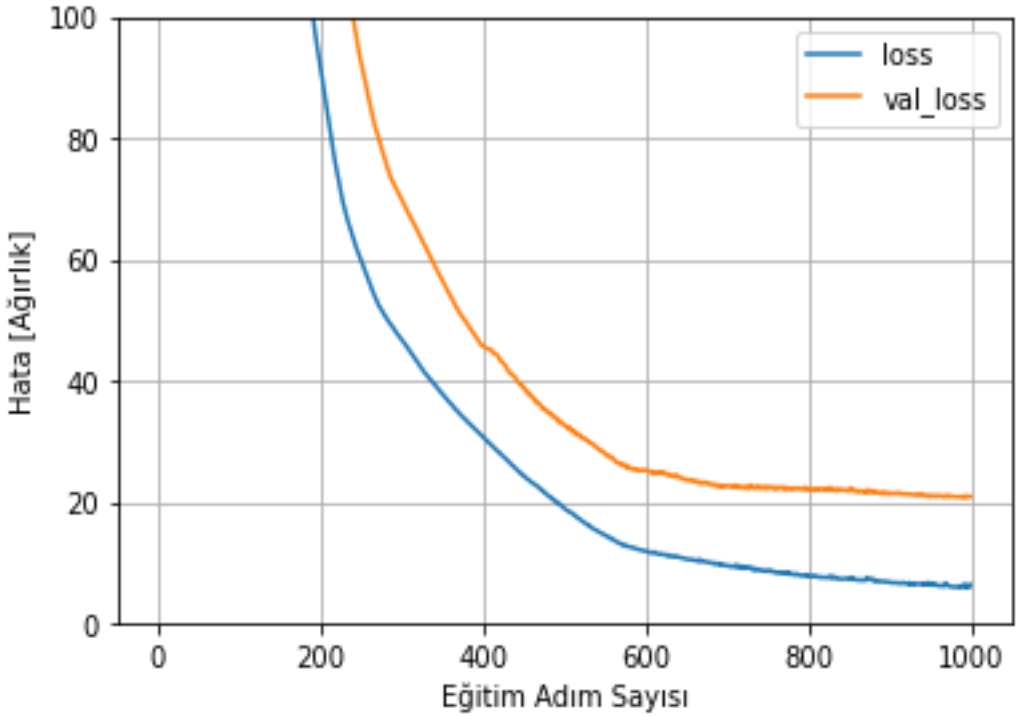
Şekil 4.25'de görüldüğü gibi sadece arkadan çekilmiş fotoğraflardan elde edilen verilerle eğitilen ağın yapıtı tahminler çok yüksek miktarlarda hatalar içermektedir.

Çalışmanın devamında üçüncü bir ağ tasarlanmış ve tüm veri seti kullanılarak bu ağ eğitilmiştir. İlk iki ağdan farklı olarak bu ağın giriş boyutu 3 değil, tüm veri setini kapsayacak şekilde 8 olmuştur. Geri kalan mimari ve parametreler olduğu gibi korunmuştur.

Tablo 4.7. Üçüncü Yapay Sinir Ağının Mimari Özeti

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
normalization (Normalization (None, 8)                 17
_________________________________________________________________
dense (Dense)                (None, 64)                576
_________________________________________________________________
dense_1 (Dense)              (None, 64)                4160
_________________________________________________________________
dense_2 (Dense)              (None, 1)                 65
=================================================================
Total params: 4,818
Trainable params: 4,801
Non-trainable params: 17
```

Tablo 4.7'de görüldüğü gibi ağdaki giriş sayısı 3'den 8'e yükselmiş olduğundan dolayı ağdaki parametre sayısı 4818'e yükselmiştir. Tüm veri seti ile eğitilen bu ağın başarım düzeyi şekil 4.26'de görülmektedir.



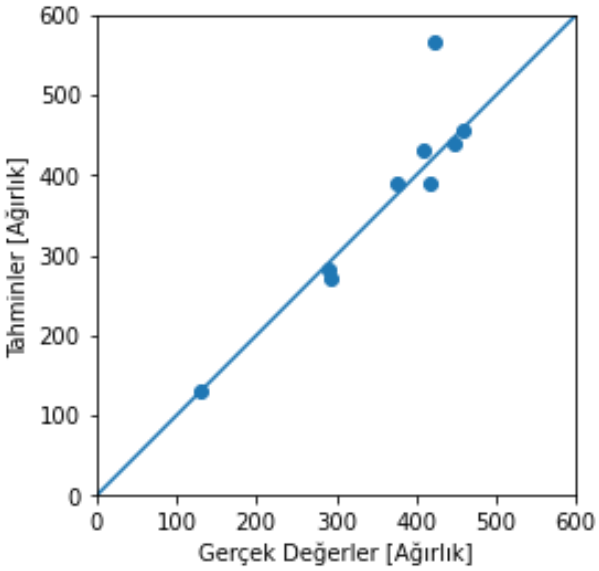
Şekil 4.26. Üçüncü Yapay Sinir Ağının Kayıp Grafiği

Şekil 4.26'de görüldüğü gibi tüm veri seti ile eğitilen yapay sinir ağında kayıp değerleri ilk iki ağa göre çok daha düşüktür. Bu ağın ilk iki ağa göre daha başarılı olmasının sebebi, hayvanların her iki açıdan çekilmiş fotoğraflarından elde edilen verilerle eğitilmiş olmasıdır. Bu durumda, imkanlar dahilinde üstten de fotoğraflar çekilebilseydi daha başarılı bir ağ elde edilebileceği sonucuna varılabilir. Ayrıca veri setindeki hayvan sayısını arttırmanın ve veri dağılımını daha ideal hale getirmenin de daha başarılı sonuçlar elde edilmesini sağlayacağı söylenebilir. Şimdi bu ağın test seti üzerinde gerçekleştirdiği tahminleri tablo üzerinde görelim.

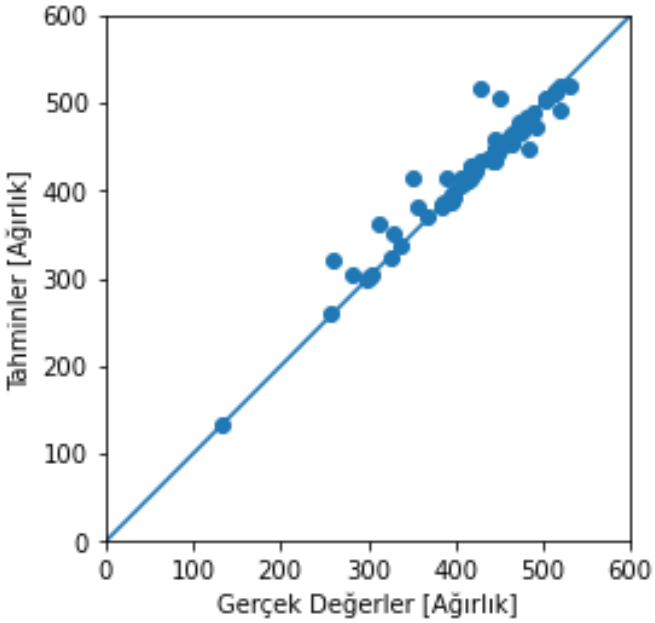
Tablo 4.8. Üçüncü Ağın Tahmin Değerleri

id	Sol Yan Çekim	Sağ Yan Çekim	Yan Piksel Farkı	Yan Mesafe	Sol Arka Çekim	Sağ Arka Çekim	Arka Piksel Farkı	Arka Mesafe	Gerçek Agirlik	Tahmin Edilen Ağırlık
9	60146	59403	34	1.88629	9819	10676	24	2.672242	291	281.71
21	14031	14031	7	9.16197	8818	9227	11	5.830346	376	389.12
36	72247	71788	30	2.13779	45864	45521	59	1.087014	423	566.15
44	44624	44198	20	3.20669	10000	10126	11	5.830346	418	389.92
47	43968	44156	20	3.20669	28421	29072	25	2.565352	447	438.57
64	20079	19978	8	8.01673	5726	5492	6	10.68897	458	456.09
67	18592	18507	8	8.01673	6002	6056	8	8.016726	409	430.47
70	13850	13713	22	2.91517	6753	6239	18	3.562989	131	130.8
81	40195	40141	22	2.91517	3831	3852	10	6.413381	294	272.02

Tablo 4.8'de görüldüğü gibi çoğu durumda tahminler 10-20 kg hata payı ile gerçekleşerek başarılı olmasına rağmen 36 id numaralı hayvanda başarısız bir tahmin gerçekleşmiştir. Bu hayvanın verilerine dikkat edilirse diğer hayvanlara göre hem yandan hemde özellikle arkadan kamera düzlemine çok daha yakın olarak fotoğraflandığı görülür. Fotoğraflama aşamasında hayvanların ürkmesi sebebiyle genel olarak orta ve uzak mesafelerden fotoğraflar çekilebilmiştir. Veri setinde yakından fotoğraflanan hayvanların sayısının az olması, yapay sinir ağının yakın hayvanlar için yeterince eğitilemediğinin düşünülmesine sebep olmuştur. Bu noktada daha kapsamlı ve veri dağılımının daha homojen olduğu bir veri seti ile eğitilen bir ağın çok daha başarılı olacağı söylenebilir.

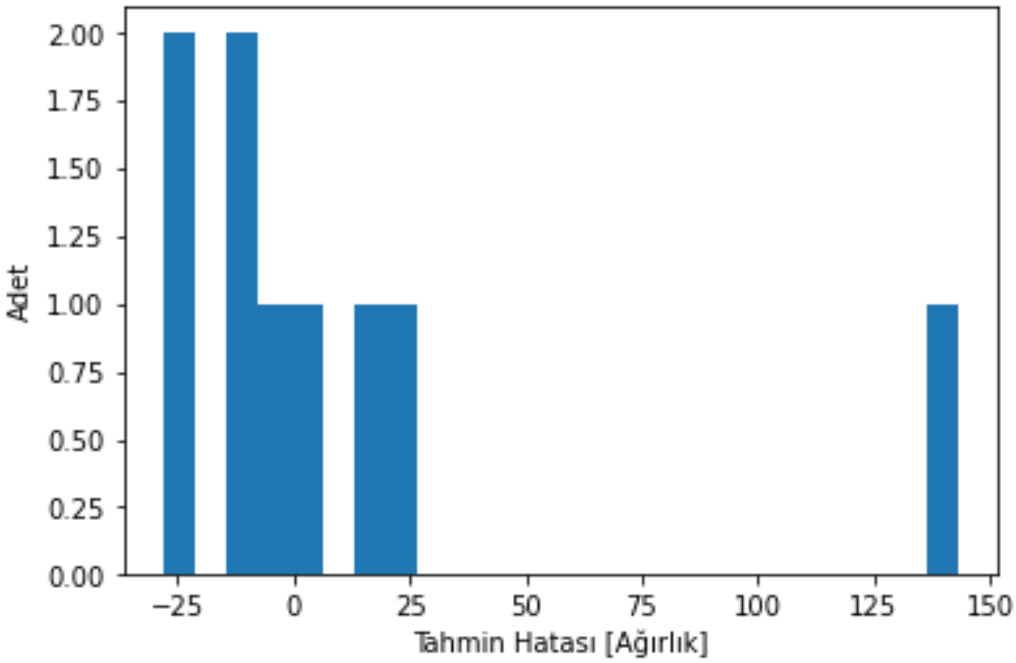


Şekil 4.27. Üçüncü Ağın Test Seti Üzerinde Yaptığı Tahminler

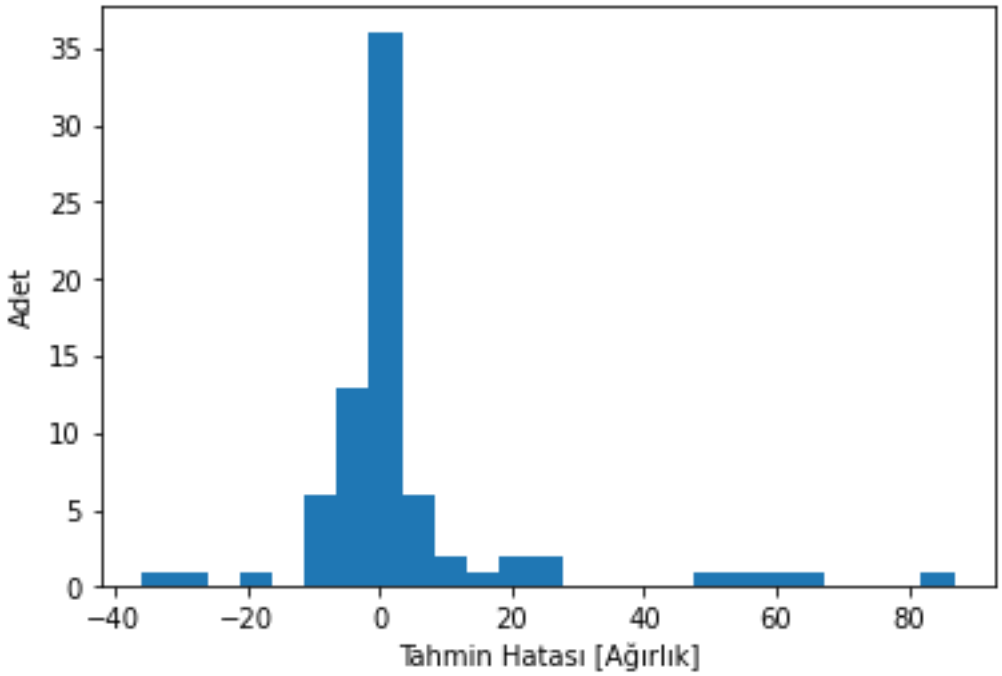


Şekil 4.28. Üçüncü Ağın Eğitim Seti Üzerinde Yaptığı Tahminler

Şekil 4.28'de gerçek değerler ve tahmin edilen değerler eğitim seti için grafiksel olarak gösterilmiştir.



Şekil 4.29. Üçüncü Ağın Test Seti Üzerinde Hata Dağılımı Grafiği



Şekil 4.30. Üçüncü Ağın Eğitim Seti Üzerinde Hata Dağılımı Grafiği

Şekil 4.30'da eğitim seti üzerinden yapılan tahminlerin hata dağılımı grafiği görülmektedir. Genel görünüm hataların ± 20 kilogram etrafında toplandığıdır. Yüksek

miktarda hatanın olduğu tek tahmin kamera düzlemine çok yakın çekilen hayvandan kaynaklanmaktadır.

BÖLÜM 5. SONUÇ

Bu çalışmada literatürde stereo görme ve semantik segmentasyon olarak isimlendirilen yöntemler kullanılarak canlı hayvan ağırlığı tahmini gerçekleştirilmeye çalışılmıştır. Çalışma kapsamında stereo görme düzeneği hazırlanmış, bu düzenek ile ağırlıkları önceden bilinen 85 adet büyükbaş hayvanın stereo fotoğrafları elde edilmiştir. Bu fotoğraflardaki hayvanların, semantik segmentasyon modellerinden olan ve deeplab v3+ olarak isimlendirilen bir derin öğrenme modeli ile segmentasyon haritaları çıkarılmıştır. Devamında bu segmentasyon haritaları kullanılarak her bir hayvanın görüntüde kapladığı piksel sayıları ve stereo mesafe hesaplama tekniği kullanılarak kamera düzlemine olan mesafeleri hesaplanmıştır. Elde edilen bu veriler bir araya getirilerek bir veri seti oluşturulmuştur. Veri seti, hayvanların yandan ve arkadan olmak üzere iki farklı açıdan çekilen fotoğraflarından elde edilen verilerden oluşturulmuştur. Bu veri seti kullanılarak, mimari açıdan birbirinin benzeri üç farklı yapay sinir ağı eğitilmiştir.

Birinci sinir ağı sadece yandan çekilen görüntülerden elde edilen verilerle, eğitilmiştir. Bu sinir ağının yaptığı ağırlık tahminlerinde hata miktarı ±40 kilogram civarındadır. İkinci sinir ağı arkadan çekilen görüntülerden elde edilen verilerle eğitilmiştir. Bu sinir ağının yaptığı ağırlık tahminlerinde hata miktarı ±100 kilogram civarındadır. Hata miktarının yüksek olması arkadan çekilen görüntülerin hayvanın vücut ölçüleri hakkında yeterli bilgiyi içermemesidir. Üçüncü sinir ağı ise tüm veri setindeki verilerle eğitilmiştir. Bu sinir ağının yaptığı ağırlık tahminlerinde hata miktarı ±20 kilograma kadar düşmüştür.

Eğitilen sinir ağları karşılaştırıldığında tüm veri seti ile eğitilen sinir ağının ilk iki sinir ağına göre önemli ölçüde başarılı olduğu görülmüştür. Bu noktada daha çok açıdan çekilmiş fotoğraflarla oluşturulan veri setleri ile eğitilecek sinir ağlarının daha da başarılı olacağı sonucu çıkarılabilir. Ayrıca veri setinin nitelik ve niceliğinin arttırılmasınında sinir ağlarının daha başarılı tahminler yapacağını söylemek mümkündür. Veri setinde sınırlı sayıda temsil edilen, hafif ağırlıkta olan hayvanların ağırlık tahminlerinin nispeten

başarısız olduğu görülmüştür. Bu nedenle daha kapsamlı ve homojen dağılımlı bir veri seti oluşturmanın modelin performansını önemli ölçüde arttıracağı da söylenebilir. Ayrıca hayvanın cinsi de ağırlığına direk olarak etki etmektedir. Örneğin farklı cins iki hayvanın vücut ölçülerinin tamamen aynı olduğu varsayılırsa bu iki hayvanın ağrılıkları birbirinden farklı olacağı görülür. Bu noktada çalışmada, hayvanın cinsini tanıyan ek bir derin öğrenme metodu geliştirlerek her cins için ayrı bir eğitim gerçekleştirilerek modellerin başarımları arttırılabilir.

KAYNAKÇA

[1] O. Alpan and A. R. Aksoy, *Sığır Yetiştiriciliği Ve Besiciliği*, 6th ed. Kars: Milsan Basın San. AŞ, 2012.

[2] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 9351, pp. 234–241, 2015, doi: 10.1007/978-3-319-24574-4_28.

[3] M. Soyaslan, E. Nart, and Ö. Çetin, "Stereo kamera sisteminde aykırılık haritaları yardımıyla nesne uzaklıklarının tespit edilmesi," *SAÜ Fen Bilim. Enstitüsü Derg.*, vol. 20, no. 2, p. 111, 2015, doi: 10.16984/saufenbilder.77095.

[4] N. Bandi, R. B. Tunyogi, Z. Szabo, E. Farkas, and C. Sulyok, "Image-Based Volume Estimation Using Stereo Vision," *SISY 2020 - IEEE 18th Int. Symp. Intell. Syst. Informatics, Proc.*, pp. 55–60, 2020, doi: 10.1109/SISY50555.2020.9217089.

[5] E. Shelhamer, J. Long, and T. Darrell, "Fully Convolutional Networks for Semantic Segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 4, pp. 640–651, 2017, doi: 10.1109/TPAMI.2016.2572683.

[6] S. KOÇAK, B. YÜCEER, M. UĞURLU, and C. ÖZBEYAZ, "BALA TARIM İŞLETMESİNDE YETİŞTİRİLEN HOLŞTAYN İNEKLERDE BAZI VERİM ÖZELLİKLERİ," *Lalahan Hayvancılık Araştırma Enstitüsü Derg.*, vol. 47, no. 1, pp. 9–14, Jun. 2007.

[7] N. Brandl and E. Jørgensen, "Determination of live weight of pigs from dimensions measured using image analysis," *Comput. Electron. Agric.*, vol. 15, no. 1, pp. 57–72, 1996, doi: 10.1016/0168-1699(96)00003-8.

[8] C. P. Schofield, J. A. Marchant, R. P. White, N. Brandl, and M. Wilson, "Monitoring pig growth using a prototype imaging system," *J. Agric. Eng. Res.*, vol. 72, no. 3, pp. 205–210, 1999, doi: 10.1006/jaer.1998.0365.

[9] L. De Wet, E. Vranken, A. Chedad, J. M. Aerts, J. Ceunen, and D. Berckmans, "Computer-assisted image analysis to quantify daily growth rates of broiler chickens," *Br. Poult. Sci.*, vol. 44, no. 4, pp. 524–532, 2003, doi: 10.1080/00071660310001616192.

[10] A. R. Kristensen, "Information from on-line live weight assessment for optimal selection of slaughter pigs for market," *Fourth Eur. Conf. Eur. Fed. Inf. Technol. Agric. Food Environ.*, no. July, pp. 754–759, 2003.

[11] J. Wu, R. Tillett, N. McFarlane, X. Ju, J. P. Siebert, and P. Schofield, "Extracting

the three-dimensional shape of live pigs using stereo photogrammetry," *Comput. Electron. Agric.*, vol. 44, no. 3, pp. 203–222, 2004, doi: 10.1016/j.compag.2004.05.003.

[12] D. Stajnko, M. Brus, and M. Hočevar, "Estimation of bull live weight through thermographically measured body dimensions," *Comput. Electron. Agric.*, vol. 61, no. 2, pp. 233–240, 2008, doi: 10.1016/j.compag.2007.12.002.

[13] Y. Wang, W. Yang, P. Winter, and L. Walker, "Walk-through weighing of pigs using machine vision and an artificial neural network," *Biosyst. Eng.*, vol. 100, no. 1, pp. 117–125, 2008, doi: 10.1016/j.biosystemseng.2007.08.008.

[14] P. Negretti, G. Bianconi, S. Bartocci, S. Terramoccia, and M. Verna, "Determination of live weight and body condition score in lactating Mediterranean buffalo by Visual Image Analysis," *Livest. Sci.*, vol. 113, no. 1, pp. 1–7, 2008, doi: 10.1016/j.livsci.2007.05.018.

[15] M. B. R. Mollah, M. A. Hasan, M. A. Salam, and M. A. Ali, "Digital image analysis to estimate the live weight of broiler," *Comput. Electron. Agric.*, vol. 72, no. 1, pp. 48–52, 2010, doi: 10.1016/j.compag.2010.02.002.

[16] S. Tasdemir, A. Urkmez, and S. Inal, "Determination of body measurements on the Holstein cows using digital image analysis and estimation of live weight with regression analysis," *Comput. Electron. Agric.*, vol. 76, no. 2, pp. 189–197, 2011, doi: 10.1016/j.compag.2011.02.001.

[17] P. Menesatti, C. Costa, F. Antonucci, R. Steri, F. Pallottino, and G. Catillo, "A low-cost stereovision system to estimate size and weight of live sheep," *Comput. Electron. Agric.*, vol. 103, pp. 33–38, 2014, doi: 10.1016/j.compag.2014.01.018.

[18] A. K. Mortensen, P. Lisouski, and P. Ahrendt, "Weight prediction of broiler chickens using 3D computer vision," *Comput. Electron. Agric.*, vol. 123, pp. 319–326, 2016, doi: 10.1016/j.compag.2016.03.011.

[19] C. Shi, G. Teng, and Z. Li, "An approach of pig weight estimation using binocular stereo system based on LabVIEW," *Comput. Electron. Agric.*, vol. 129, pp. 37–43, 2016, doi: 10.1016/j.compag.2016.08.012.

[20] M. F. Hansen, M. L. Smith, L. N. Smith, K. Abdul Jabbar, and D. Forbes, "Automated monitoring of dairy cow body condition, mobility and weight using a single 3D video capture device," *Comput. Ind.*, vol. 98, pp. 14–22, 2018, doi: 10.1016/j.compind.2018.02.011.

[21] O. Nir, Y. Parmet, D. Werner, G. Adin, and I. Halachmi, "3D Computer-vision system for automatically estimating heifer height and body mass," *Biosyst. Eng.*, vol. 173, pp. 4–10, 2018, doi: 10.1016/j.biosystemseng.2017.11.014.

[22] X. Song, E. A. M. Bokkers, P. P. J. van der Tol, P. W. G. Groot Koerkamp, and S. van Mourik, "Automated body weight prediction of dairy cows using 3-dimensional vision," *J. Dairy Sci.*, vol. 101, no. 5, pp. 4448–4459, 2018, doi:

10.3168/jds.2017-13094.

[23] S. Suwannakhun and P. Daungmala, “Estimating Pig Weight with Digital Image Processing using Deep Learning,” *Proc. - 14th Int. Conf. Signal Image Technol. Internet Based Syst. SITIS 2018*, no. 5, pp. 320–326, 2018, doi: 10.1109/SITIS.2018.00056.

[24] A. Ter-Sarkisov, J. Kelleher, B. Earley, M. Keane, and R. Ross, “Beef cattle instance segmentation using fully convolutional neural network,” *arXiv*, pp. 1–11, 2018.

[25] S. Tasdemir and I. A. Ozkan, “Ann Approach for Estimation of Cow Weight Depending on Photogrammetric Body Dimensions,” *Int. J. Eng. Geosci.*, no. 1, pp. 36–44, 2019, doi: 10.26833/ijeg.427531.

[26] A. Cominotte *et al.*, “Automated computer vision system to predict body weight and average daily gain in beef cattle during growing and finishing phases,” *Livest. Sci.*, vol. 232, no. August 2019, p. 103904, 2020, doi: 10.1016/j.livsci.2019.103904.

[27] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks,” *Adv. Neural Inf. Process. Syst. 25 (NIPS 2012)*, 2012.

[28] D. A. Forsyth and J. Ponce, *Computer Vision A Modern Approach(Second Edition)*. PEARSON, 2012.

[29] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision(Second Edition)*. Cambridge University Press, 2004.

[30] G. Bradski and A. Kaehler, *Learning OpenCV---Computer Vision with the OpenCV Library (Bradski, G.R. et al.; 2008)[On the Shelf]*, vol. 16, no. 3. 2009.

[31] D. Brown, “Decentering Distortion of Lenses - The Prism Effect Encountered in Metric Cameras can be Overcome Through Analytic Calibration,” *Photom. Eng.*, vol. 32, no. 3, pp. 444–462, 1966.

[32] J. Heikkila and O. Silven, “A Four-step Camera Calibration Procedure with Implicit Image Correction,” *Infotech Oulu Dep. Electr. Eng. Univ. Oulu*, pp. 1106–1112, 1997.

[33] B. G. Cumming and G. C. DeAngelis, “THE PHYSIOLOGY OF STEREOPSIS,” *Ann Rev Neurosci*, vol. 24, pp. 203–238, 2001.

[34] N. M. Tuan, Q. Van Hau, S. Chin, and S. Park, “In-situ concrete slump test incorporating deep learning and stereo vision,” *Autom. Constr.*, vol. 121, no. October 2020, p. 103432, 2021, doi: 10.1016/j.autcon.2020.103432.

[35] S. I. Kim and J. C. Namgung, “Distance Measurement of Object using LASER Pointer in the Stereo Vision System,” *J. Korea Inf. Commun. Soc.*, vol. 27.

[36] M. Domnguez-Morales, A. Jimnez-Fernndez, R. Paz-Vicente, A. Linares-Barranco, and G. Jimnez-Moreno, "Stereo Matching: From the Basis to Neuromorphic Engineering," *Curr. Adv. Stereo Vis.*, pp. 1–22, 2012, doi: 10.5772/45901.

[37] Ç. Elmas, *YAPAY SİNİR AĞLARI*. Ankara: Seçkin Yayınevi, 2003.

[38] K. L. Priddy and P. E. Keller, *Artificial Neural Networks: An Introduction*. 2009.

[39] V. V. NABİYEV, *Yapay Zeka*, 5th ed. Ankara: Seçkin Yayınevi, 2016.

[40] S. J. Russell and P. Norvig, *Artificial intelligence A Modern Approach*, Third., vol. 4. PEARSON, 2010.

[41] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2323, 1998, doi: 10.1109/5.726791.

[42] "Convolutional Neural Networks (CNNs / ConvNets)," *Stanford University*, 2020. .

[43] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 8689 LNCS, no. PART 1, pp. 818–833, 2014, doi: 10.1007/978-3-319-10590-1_53.

[44] C. Szegedy *et al.*, "Going deeper with convolutions," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 07-12-June, pp. 1–9, 2015, doi: 10.1109/CVPR.2015.7298594.

[45] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc.*, pp. 1–14, 2015.

[46] F. Lateef and Y. Ruichek, "Survey on semantic segmentation using deep learning techniques," *Neurocomputing*, vol. 338, pp. 321–348, 2019, doi: 10.1016/j.neucom.2019.02.003.

[47] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2016-Decem, pp. 770–778, 2016, doi: 10.1109/CVPR.2016.90.

[48] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 9908 LNCS, pp. 630–645, 2016, doi: 10.1007/978-3-319-46493-0_38.

[49] Z. Wu, C. Shen, and A. van den Hengel, "Wider or Deeper: Revisiting the ResNet Model for Visual Recognition," *Pattern Recognit.*, vol. 90, pp. 119–133, 2019, doi: 10.1016/j.patcog.2019.01.006.

[50] A. Casanova, G. Cucurull, M. Drozdzal, A. Romero, and Y. Bengio, “On the iterative refinement of densely connected representation levels for semantic segmentation,” *IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Work.*, vol. 2018-June, pp. 1091–1100, 2018, doi: 10.1109/CVPRW.2018.00144.

[51] A. Valada, J. Vertens, A. Dhall, and W. Burgard, “AdapNet: Adaptive semantic segmentation in adverse environmental conditions,” *Proc. - IEEE Int. Conf. Robot. Autom.*, pp. 4644–4651, 2017, doi: 10.1109/ICRA.2017.7989540.

[52] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, “Aggregated residual transformations for deep neural networks,” *Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017*, vol. 2017-Janua, pp. 5987–5995, 2017, doi: 10.1109/CVPR.2017.634.

[53] S. R. Bulo, L. Porzi, and P. Kontschieder, “In-place Activated BatchNorm for Memory-Optimized Training of DNNs,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 5639–5647, 2018, doi: 10.1109/CVPR.2018.00591.

[54] X. Liang, E. Xing, and H. Zhou, “Dynamic-Structured Semantic Propagation Network,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 752–761, 2018, doi: 10.1109/CVPR.2018.00085.

[55] T. Pohlen, A. Hermans, M. Mathias, and B. Leibe, “Full-resolution residual networks for semantic segmentation in street scenes,” *Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017*, vol. 2017-Janua, pp. 3309–3318, 2017, doi: 10.1109/CVPR.2017.353.

[56] A. Paszke, A. Chaurasia, S. Kim, and E. Culurciello, “ENet: A Deep Neural Network Architecture for Real-Time Semantic Segmentation,” pp. 1–10, 2016, [Online]. Available: http://arxiv.org/abs/1606.02147.

[57] E. Romera, J. M. Alvarez, L. M. Bergasa, and R. Arroyo, “ERFNet: Efficient Residual Factorized ConvNet for Real-Time Semantic Segmentation,” *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 1, pp. 263–272, 2018, doi: 10.1109/TITS.2017.2750080.

[58] S. Mehta, M. Rastegari, A. Caspi, L. Shapiro, and H. Hajishirzi, “ESPNet: Efficient spatial pyramid of dilated convolutions for semantic segmentation,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 11214 LNCS, pp. 561–580, 2018, doi: 10.1007/978-3-030-01249-6_34.

[59] L. Deng, M. Yang, H. Li, T. Li, B. Hu, and C. Wang, “Restricted Deformable Convolution-Based Road Scene Semantic Segmentation Using Surround View Cameras,” *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 10, pp. 4350–4362, 2020, doi: 10.1109/TITS.2019.2939832.

[60] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” *Proc. IEEE Comput. Soc.*

Conf. Comput. Vis. Pattern Recognit., pp. 580–587, 2014, doi: 10.1109/CVPR.2014.81.

[61] R. Girshick, "Fast R-CNN," *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 2015 Inter, pp. 1440–1448, 2015, doi: 10.1109/ICCV.2015.169.

[62] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, 2017, doi: 10.1109/TPAMI.2016.2577031.

[63] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 2, pp. 386–397, 2020, doi: 10.1109/TPAMI.2018.2844175.

[64] X. Li, T. Lai, S. Wang, Q. Chen, C. Yang, and R. Chen, "Feature pyramid networks for object detection," *Proc. - 2019 IEEE Intl Conf Parallel Distrib. Process. with Appl. Big Data Cloud Comput. Sustain. Comput. Commun. Soc. Comput. Networking, ISPA/BDCloud/SustainCom/SocialCom 2019*, pp. 1500–1504, 2019, doi: 10.1109/ISPA-BDCloud-SustainCom-SocialCom48970.2019.00217.

[65] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path Aggregation Network for Instance Segmentation," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 8759–8768, 2018, doi: 10.1109/CVPR.2018.00913.

[66] F. Visin *et al.*, "ReSeg: A Recurrent Neural Network-Based Model for Semantic Segmentation," *IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Work.*, pp. 426–433, 2016, doi: 10.1109/CVPRW.2016.60.

[67] Y. Wang, J. Liu, Y. Li, J. Yan, and H. Lu, "Objectness-aware semantic segmentation," *MM 2016 - Proc. 2016 ACM Multimed. Conf.*, pp. 307–311, 2016, doi: 10.1145/2964284.2967232.

[68] L. C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, 2016, doi: 10.1109/TPAMI.2017.2699184.

[69] D. M. Vo and S. W. Lee, "Semantic image segmentation using fully convolutional neural networks with multi-scale images and multi-scale dilated convolutions," *Multimed. Tools Appl.*, vol. 77, no. 14, pp. 18689–18707, 2018, doi: 10.1007/s11042-018-5653-x.

[70] Z. Huang, X. Wang, J. Wang, W. Liu, and J. Wang, "Weakly-Supervised Semantic Segmentation Network with Deep Seeded Region Growing," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 7014–7023, 2018, doi: 10.1109/CVPR.2018.00733.

[71] S. Zheng *et al.*, "Conditional random fields as recurrent neural networks," *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 2015 Inter, pp. 1529–1537, 2015, doi: 10.1109/ICCV.2015.179.

[72] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2016-Decem, pp. 770–778, 2016, doi: 10.1109/CVPR.2016.90.

[73] Y. Li, H. Qi, J. Dai, X. Ji, and Y. Wei, “Fully convolutional instance-aware semantic segmentation,” *Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017*, vol. 2017-Janua, pp. 4438–4446, 2017, doi: 10.1109/CVPR.2017.472.

[74] D. C. Cireşan, A. Giusti, L. M. Gambardella, and J. Schmidhuber, “Deep neural networks segment neuronal membranes in electron microscopy images,” *Adv. Neural Inf. Process. Syst.*, vol. 4, pp. 2843–2851, 2012.

[75] L. C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, “SEMANTIC IMAGE SEGMENTATION WITH DEEP CONVOLUTIONAL NETS AND FULLY CONNECTED CRFS,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, 2015, doi: 10.1109/TPAMI.2017.2699184.

[76] L. C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, “DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, 2017, doi: 10.1109/TPAMI.2017.2699184.

[77] L. C. Chen, G. Papandreou, F. Schroff, and H. Adam, “Rethinking atrous convolution for semantic image segmentation,” *arXiv*, 2017.

[78] L. C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, “Encoder-decoder with atrous separable convolution for semantic image segmentation,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 11211 LNCS, pp. 833–851, 2018, doi: 10.1007/978-3-030-01234-2_49.

[79] P. Krähenbühl and V. Koltun, “Efficient inference in fully connected crfs with Gaussian edge potentials,” *Adv. Neural Inf. Process. Syst. 24 25th Annu. Conf. Neural Inf. Process. Syst. 2011, NIPS 2011*, pp. 1–9, 2011.

[80] M. Holschneider, R. Kronland-Martinet, J. Morlet, and P. Tchamitchian, “A Real-Time Algorithm for Signal Analysis with the Help of the Wavelet Transform,” pp. 286–297, 1989, doi: 10.1007/978-3-642-97177-8_28.

[81] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” in *32nd International Conference on Machine Learning, ICML 2015*, 2015, vol. 1.

[82] J. Dai *et al.*, “Deformable Convolutional Networks,” *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 2017-Octob, pp. 764–773, 2017, doi: 10.1109/ICCV.2017.89.