

T.C.
FIRAT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ



**DERİN ÖĞRENME TABANLI OPTİK AKIŞ İLE HAREKET
TESPİTİ**

Ammar ASLAN

Yüksek Lisans Tezi

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

Kuramsal Temeller Bilim Dalı

TEMMUZ 2021

T.C.
FIRAT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

Bilgisayar Mühendisliği Anabilim Dalı

Yüksek Lisans Tezi

DERİN ÖĞRENME TABANLI OPTİK AKIŞ İLE HAREKET TESPİTİ

Tez Yazarı

Ammar ASLAN

Danışman

Doç. Dr. Galip AYDIN

TEMMUZ 2021

ELAZIĞ

T.C.
FIRAT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

Bilgisayar Mühendisliği Anabilim Dalı

Yüksek Lisans Tezi

Başlığı: Derin Öğrenme Tabanlı Optik Akış ile Hareket Tespiti
Yazarı: Ammar ASLAN
İlk Teslim Tarihi: 21.06.2021
Savunma Tarihi: 26.07.2021

TEZ ONAYI

Fırat Üniversitesi Fen Bilimleri Enstitüsü tez yazım kurallarına göre hazırlanan bu tez aşağıda imzaları bulunan jüri üyeleri tarafından değerlendirilmiş ve akademik dinleyicilere açık yapılan savunma sonucunda OYBİRLİĞİ ile kabul edilmiştir.

İmza

Danışman: Doç. Dr. Galip AYDIN Onayladım
Fırat Üniversitesi, Mühendislik Fakültesi

Başkan: Dr. Öğr. Üyesi Yasin SÖNMEZ Onayladım
Batman Üniversitesi, Teknik Bilimler MYO

Üye: Dr. Öğr. Üyesi Orhan YAMAN Onayladım
Fırat Üniversitesi, Teknoloji Fakültesi

Bu tez, Enstitü Yönetim Kurulunun/...../20..... tarihli toplantısında tescillenmiştir.

İmza

Doç. Dr. Kürşat Esat ALYAMAÇ
Enstitü Müdürü

BEYAN

Fırat Üniversitesi Fen Bilimleri Enstitüsü tez yazım kurallarına uygun olarak hazırladığım “Derin Öğrenme Tabanlı Optik Akış ile Hareket Tespiti” Başlıklı Yüksek Lisans Tezimin içindeki bütün bilgilerin doğru olduğunu, bilgilerin üretilmesi ve sunulmasında bilimsel etik kurallarına uygun davrandığımı, kullandığım bütün kaynakları atıf yaparak belirttiğimi, maddi ve manevi desteği olan tüm kurum/kuruluş ve kişileri belirttiğimi, burada sunduğum veri ve bilgileri unvan almak amacıyla daha önce hiçbir şekilde kullanmadığımı beyan ederim.

26.07.2021

Ammar ASLAN



ÖNSÖZ

Hareket tespiti, güvenlik sistemleri, otonom sistemlerde, sağlık alanında, trafikte, askeri alan gibi hayatımızın önemli alanlarında ihtiyaç olarak sıklıkla karşımıza çıkan bir gereksinimdir. Hareket tespiti, video verisi üzerinde art arda gelen iki görüntünün, birbirinden farkı alınmasından en popüler konulardan olan derin öğrenme yöntemlerine varıncaya dek birçok yöntem bulunmaktadır. Hareket tespit ve takip yöntemlerinin alt yapısında sıklıkla kullanılan optik akış yöntemleri hareket tahmini için önemli bir yere sahiptir. Öte yandan derin öğrenme son dönemlerde gösterdiği üstün başarıları ile adından söz ettirmeyi başarmıştır. Klasik yöntemlere derin öğrenme tabanlı yöntemler uygulanarak sonuçların performansı artırılabilir. Performans ve doğruluk açısından öneme sahip olmasından ve gelişen teknolojik dönemde daha iyi performansa sahip olması adına derin öğrenme tabanlı optik akış hesaplaması ile hareket tespit yöntemleri bu tez çalışmasında ana konusunu oluşturmaktadır.

Bu tez çalışmasında zaman gözetmeksizin bilgi, beceri ve desteğini esirgemeyen çok değerli hocam ve tez danışmanım sayın **Doç. Dr. Galip AYDIN**'a çok teşekkür ederim.

Hep yanımda olan ve desteğini esirgemeyen eşim Tuba ASLAN'a, değerli arkadaşlarım İsmail Hakkı YILDIZ, Abdullah Yıldırım'a, başta annem Vahide ASLAN, babam Zeki ASLAN ve desteğini esirgemeyen değerli ağabeyim Nedim ASLAN olmak üzere tüm aileme sonsuz teşekkürlerimi sunarım.

Bu tezimi, oğlum Ömer Eymen ASLAN'a ve ithaf ediyorum.

Ammar ASLAN

ELAZIĞ, 2021

İÇİNDEKİLER

Sayfa

ÖNSÖZ.....	iv
İÇİNDEKİLER	v
ÖZET	vii
ABSTRACT	viii
ŞEKİLLER LİSTESİ	ix
TABLolar LİSTESİ	xii
SİMGELER VE KISALTMALAR	xiii
1. GİRİŞ	1
2. LİTERATÜR ÖZETİ.....	7
3. DERİN ÖĞRENME	19
3.1. Yapay Sinir Ağları.....	20
3.1.1. Tek Katmanlı Algılayıcılar.....	22
3.1.2. Çok Katmanlı Algılayıcılar	24
3.1.3. Eğitici (Gözetimli Supervised) ve Eğitici (Gözetimsiz Unsupervised) Öğrenme.....	24
3.1.4. İleriye ve Geriye Yayılım.....	24
3.2. Derin Öğrenme Model Yapısı	26
3.2.1. Model Gelişiminde Karşılaşılan Problemler	26
3.2.2. Model Gelişiminde İyileştirme Fikirleri.....	27
3.2.3. Regularizasyon	27
3.2.4. Optimizasyon	28
3.3. Evrişimli Sinir Ağları (Convolutional Neural Networks).....	29
3.3.1. Evrişimli Sinir Ağı Katmanları	30
3.3.2. Evrişimli Sinir Ağı Mimarisi.....	32
3.4. Özyinelemeli Sinir Ağları (Recurrent Neural Networks)	36
3.5. Kapsül Ağları (Capsule Networks).....	37
4. OPTİK AKIŞ.....	38
4.1. Optik Akış Yön ve Şiddet Haritası	38
4.2. Optik Akış Denklemi.....	39
4.3. Seyrek Optik Akış	41
4.3.1. Lucas-Kanade Optik Akış Yöntemi	41
4.3.2. Horn ve Schunck Optik Akış Yöntemi.....	44
4.4. Yoğun Optik Akış.....	46
4.4.1. Farneback Optik Akış Yöntemi.....	46
5. UYGULAMALAR.....	47
5.1. Optik Akış Veri Kümeleri	47
5.1.1. MPI Sintel Veri Kümesi	47
5.1.2. KITTI Veri Kümesi	48
5.1.3. Flying Chairs (Uçan Sandalyeler) Veri Kümesi	49

5.2. Geliştirme Ortamı ve Optik Akış Değerlendirme Parametreleri	50
5.2.1. Colaboratory (Google Colab)	50
5.2.2. TensorBoard Gösterim Aracı	51
5.2.3. Son Nokta Hatası (End-point Error).....	51
5.2.4. Kayıp (Loss-L1).....	52
5.3. İlk Derin Öğrenme Tabanlı Optik Akış Mimarisi (FlowNet)	52
5.3.1. Mimari Yapısı	52
5.3.2. Modellerin Eğitilmesi ve Sonuçları.....	53
5.3.3. Görsel Çıktılar	57
5.4. Derin Öğrenme Tabanlı Optik Akış Mimarisi (FlowNet2)	59
5.4.1. Mimari Yapısı	59
5.4.2. Model Eğitimi ve Sonuçlar.....	59
5.4.3. Görsel Çıktılar	60
5.5. CNN ve RNN Tabanlı Optik Akış Mimarisi -RAFT Mimarisi	62
5.5.1. Mimari Yapısı	62
5.5.2. Görsel Çıktılar	62
6. BULGULAR VE TARTIŞMA.....	65
7. SONUÇLAR.....	68
KAYNAKLAR.....	70
ÖZGEÇMİŞ	

ÖZET

Derin Öğrenme Tabanlı Optik Akış ile Hareket Tespiti

Ammar ASLAN

Yüksek Lisans Tezi

FIRAT ÜNİVERSİTESİ

Fen Bilimleri Enstitüsü

Bilgisayar Mühendisliği Anabilim Dalı

Temmuz 2021, Sayfa: xiii + 74

Hareket tespiti ve tahmini, güvenlik(bina/işyeri), otonom araçlar, sağlık, trafik, askeri gibi birçok alanda karşımıza çıkan önemli uygulamaları olan bir kavramdır. Hareket tespiti sırasında ortam bozulmaları, ışık değişimleri, nesnenin hızlı hareketi ve gürültüler hareket tespitini zorlaştırmaktadır. Hareket tespitinde kullanılan yöntemlerin ve algoritmaların performansı, doğru hareket tespiti yapma açısından son derece önemlidir. Hareket tespiti yöntemlerinin temelini oluşturan ve birçok algoritmada baz alınan optik akış yaklaşımları, hareket tahmininde de önemli bir yere sahiptir. Hareket tespitinde kullanılan ve 2015 yılından günümüze kadar uzanan derin öğrenme tabanlı optik akış hesaplaması ile hareket tespit yöntemlerinin bilinmesi, mevcut yöntemlere göre performansı ve doğruluğunun karşılaştırılması açısından son derece önemlidir. Derin öğrenme tabanlı optik akış hesaplaması ile hareket tahmini yöntemleri ve optik akış için kullanılan veri kümeleri bu çalışmanın ana konusunu oluşturmaktadır.

Çalışmanın amacı, optik akış hesaplamasında derin öğrenme yöntemlerini kullanarak başarı elde etmiş çalışmaları incelemek, bazı algoritmaları yeniden eğiterek klasik yöntemler ile karşılaştırmak ve farklı video görüntüleri üzerinde performans analizleri gerçekleştirmektir.

Bu çalışmada, geleneksel yaklaşımlardan günümüz derin öğrenme tabanlı yaklaşımlara kadar hareket tespitinin temelini oluşturan optik akış hesaplama yöntemleri ile ilgili literatür taraması yapılmıştır. Geleneksel yaklaşımlardan olan Lucas-Kanade, Horn-Schunck, Farneback yöntemleri farklı hareketli görüntülere uygulanarak elde edilen sonuçlar karşılaştırılmıştır. Daha sonra derin öğrenme tabanlı yaklaşımlarından FlowNet, FlowNet2 ve RAFT yöntemleri Sintel, KITTI veri setleri üzerinde eğitilerek uç nokta hatası ve kayıp değerleri farklı hiper parametreler kullanılarak grafik, tablo ve görsel çıktılar vasıtasıyla karşılaştırılmıştır. Çalışmada son dönemlerde popüler olan denetimsiz öğrenme ile optik akış hesaplama yaklaşımları, denetimli öğrenme yaklaşımları ile karşılaştırarak değerlendirmelerde bulunulmuştur.

Anahtar Kelimeler: Optik Akış, Derin Öğrenme, Hareket Tespiti, Hareket Tahmini

ABSTRACT

Motion Detection with Deep Learning Based Optical Flow

Ammar ASLAN

Master's Thesis

FIRAT UNIVERSITY

Graduate School of Natural and Applied Sciences

Department of Computer Engineering

July 2021, Pages: xiii + 74

It is a concept that has important applications in many areas such as motion detection and prediction, security (building / workplace), autonomous vehicles, health, traffic, and military. During motion detection, ambient disturbances, light changes, rapid movement of the object and noises make motion detection difficult. The performance of the methods and algorithms used in motion detection is extremely important for accurate motion detection. Optical flow approaches, which form the basis of motion detection methods and are based on many algorithms, also have an important place in motion estimation. Knowing the deep learning-based optical flow calculation and motion detection methods used in motion detection and dating back to 2015 is extremely important in terms of comparing performance and accuracy with current methods. Deep learning based optical flow computation and motion estimation methods and datasets used for optical flow are the main subject of this study.

The aim of the study is to examine the studies that have achieved success by using deep learning methods in optical flow calculation, to compare some algorithms with classical methods by retraining, and to perform performance analyzes on different video images.

In this study, a literature review was conducted on optical flow calculation methods, which form the basis of motion detection, from traditional approaches to today's deep learning-based approaches. The traditional approaches, Lucas-Kanade, Horn-Schunck, Farneback methods, were applied to different motion pictures and the results obtained were compared. Then, the deep learning-based approaches FlowNet, FlowNet2 and RAFT methods were trained on Sintel, KITTI datasets and the endpoint error and loss values were compared using different hyperparameters through graphs, tables, and visual outputs. In the study, unsupervised learning, and optical flow calculation approaches, which are popular recently, were compared with supervised learning approaches.

Keywords: Optical Flow, Deep Learning, Motion Detection, Motion Estimation

ŞEKİLLER LİSTESİ

	Sayfa
Şekil 2.1. FlowNetS ve FlowNetC model yapıları [12]	10
Şekil 2.2. FlowNet2 model yapısı [13]	11
Şekil 2.3. IHA için optik akış ile hareket tespiti diyagramı	12
Şekil 2.4. Üç seviyeli piramit ağı yapısı [11].....	12
Şekil 2.5. PWCNet model yapısı [14].....	13
Şekil 2.6. RAFT özellik çıkarma ağı [16]	14
Şekil 2.7. RAFT'ın genel model yapısı [16]	15
Şekil 2.8. Gösterim akış katman yapısı [40]	15
Şekil 2.9. Video-CNN mimari yapısı [40]	16
Şekil 3.1. Derin öğrenme Google trend veri grafiği (Türkiye)	19
Şekil 3.2. Derin öğrenme Google trend veri grafiği (Dünya)	20
Şekil 3.3. YSA nöron yapısı	21
Şekil 3.4. Yapay sinir ağının genel yapısı.....	21
Şekil 3.5. Lojistik regresyon yapısı.....	22
Şekil 3.6. Sigmoid fonksiyonun grafiği	23
Şekil 3.7. YSA uygulanacak veriler ve yapıları (Renkler, veriyi ve ağırlıkları ifade etmektedir).	25
Şekil 3.8. Model değerlendirme metrikleri	28
Şekil 3.9. Evrişim katmanı işlemleri	30
Şekil 3.10. Konvolisyon uygulanan görüntü.....	31
Şekil 3.11. Ortalama ve en büyük havuzlama yapısı	32
Şekil 3.12. CNN katmanlarının tamamını içeren örnek bir model [63]	32
Şekil 3.13. LeNet-5 mimari yapısı [64]	33
Şekil 3.14. AlexNet mimari yapısı [65]	33
Şekil 3.15. VGG-16 mimari yapısı [68].....	34
Şekil 3.16. DenseNet modeli mimarisi [70].....	35
Şekil 3.17. Mimarilerin ImageNet verilerine göre derinlikleri ve hata oranları.....	36
Şekil 3.18. Özyinelemeli ağ parça yapısı [71]	37
Şekil 3.19. RNN mimari yapısı [71]	37

Şekil 4.1.	Optik akış renk ve şiddet haritası [76]	39
Şekil 4.2.	Optik akış denklemi	39
Şekil 4.3.	Shi-Tomasi filtresi [77].....	42
Şekil 4.4.	LK algoritmasının akış diyagramı	43
Şekil 4.5.	LK uygulama çıktıları	43
Şekil 4.6.	HS algoritmasının akış diyagramı.....	45
Şekil 4.7.	HS uygulama çıktıları (a: yumuşatma katsayısı)	45
Şekil 4.8.	Farneback uyguladığımız çıktıları	46
Şekil 5.1.	Sintel veri kümesinin görüntü ve etiketli akış görüntüleri	48
Şekil 5.2.	KITTI veri kümesine ait görüntüler ve veri kümesinde bulunan etiketli veriler.....	49
Şekil 5.3.	Uçan Sandalye veri kümesine ait görüntüler ve veri kümesinde bulunan etiketli veriler	50
Şekil 5.4.	Tensorboard gösterim aracı çıktı ekranı.....	51
Şekil 5.5.	Rafine işleminin detayı gösterimi [12].....	52
Şekil 5.6.	FlowNetS ağının KITTI veri setine göre Epe değerleri (Epoch Size 1000).....	53
Şekil 5.7.	FlowNetS ağının KITTI veri setine göre kayıp (loss) değerleri (Epoch Size 1000)	54
Şekil 5.8.	FlowNetC ağının KITTI veri setine göre Epe değerleri (Epoch Size 1000)	54
Şekil 5.9.	FlowNetC ağının KITTI veri setine göre kayıp (loss) değerleri (Epoch Size 1000).....	55
Şekil 5.10.	FlowNetC ağının KITTI veri setine göre Epe değerleri (Epoch Size 2700)	55
Şekil 5.11.	FlowNetC ağının KITTI veri setine göre kayıp (loss) değerleri (Epoch Size 2700).....	56
Şekil 5.12.	FlowNetS ağının KITTI veri setine göre Epe değerleri (Epoch Size 2700).....	56
Şekil 5.13.	FlowNetS ağının KITTI veri setine göre kayıp (loss) değerleri (Epoch Size 2700)	56
Şekil 5.14.	KITTI veri seti eğitim verilerine göre FlowNet çıktıları.....	57
Şekil 5.15.	Uçan Sandalyeler veri kümesine göre FlowNet uygulama çıktıları.....	58
Şekil 5.16.	Uçan sandalye verileri ile optik akış tahmini ve çıktıları.....	58
Şekil 5.17.	Sintel veri seti eğitim verilerine FlowNet2 ortalama kayıp (L1)	59
Şekil 5.18.	Sintel veri seti eğitim verilerine FlowNet2 EPE	60
Şekil 5.19.	FlowNet2, Sintel veri setine göre çıktılar	60
Şekil 5.20.	Uçan Sandalyeler veri seti eğitimine göre optik akış tahmini.....	61
Şekil 5.21.	Uçan Sandalyeler veri seti eğitimine göre optik akış tahmini.....	61
Şekil 5.22.	Uçan sandalye veri setine göre eğitilmiş ağırlıklara Sintel veri seti uygulama çıktıları	62
Şekil 5.23.	KITTI veri seti verilerine RAFT çıktıları	63

Şekil 5.24.	Sintel veri seti eğitim verilerine RAFT çıktıları	63
Şekil 5.25.	Sintel veri seti eğitim verilerine RAFT çıktıları	64
Şekil 6.1.	FlowNetS, FlowNetC, FlowNet2 ve RAFT modellerinin çıktıları	65
Şekil 6.2.	Klasik ve derin öğrenme tabanlı yaklaşımların aynı giriş görüntülerine göre çıktıları	66



TABLÖLAR LİSTESİ

	Sayfa
Tablo 3.1. YSA Hiper parametrelerin, değışime göre performansı (iterasyon 300)	26
Tablo 3.2. ImageNet verilerine göre derin öğrenme mimarilerin başarımları [66]	35
Tablo 4.1. LK’de farklı filtre boyutlarına göre çıktı karşılaştırılması	44
Tablo 5.1. Optik akış veri kümeleri ve içerikleri	47
Tablo 5.2. FlowNetS ve FlowNetC KITTI veri seti için EPE karşılaştırılması	57
Tablo 5.3. FlowNetS ve FlowNetC KITTI ve Uçan Sandalye veri seti için karşılaştırılması	57
Tablo 6.1. Sintel veri seti için RAFT ve FlowNet2 modellerinin karşılaştırmaları.....	65
Tablo 6.2. Sintel veri seti üzerinde elde edilen mimari ve başarımları [21].....	66

SİMGELER VE KISALTMALAR

Simgeler

Σ	: Toplam sembolünü temsil eder.
d	: Türev işlemini temsil eder.
∂	: Kısmi türev işlemini temsil eder.

Kısaltmalar

ANN	: Artificial neural network
CNN	: Evrişimli sinir ağları
EPE	: Average end-point error (son nokta hatası)
GPU	: Graphics Processing Unit
HS	: Horn ve Schunck
HSV	: Hue Saturation Value
IHA	: İnsansız hava araçları
KITTI	: Karlsruhe Institute of Technology and Toyota Technological Institute
LK	: Lucas ve Kanade
LSTM	: Long short – term memory
PCA	: Principal component analysis
PNG	: Portable network graphics
RNN	: Recurrent neural network
TPU	: Tensor processing unit
YDT	: Yüksek derecedeki terimler
YSA	: Yapay sinir ağları

1. Giriş

Hareket, bir nesnenin bulunduğu noktadan başka bir noktaya yer değiştirmesi olarak tanımlanmaktadır. Günümüzde bilgisayarlı görü teknolojilerinin gelişmesi ile hareketin tahmini ve tespiti üzerine yapılan çalışmalar ve araştırmalar artmıştır. Görüntü işlemenin önemli konularından biri olan hareket tespitinin birçok kullanım alanı bulunmaktadır. Kullanım alanları, güvenlik (bina/tesis vb.) alanlarında, trafik kontrol ve gözetiminde, uzay biliminde, sağlık alanında, insansız hava araçlarında (IHA) ve askeri alanlardır. Özellikle hareketli tespiti, çeşitli güvenlik ve askeri uygulamalarda problem çözmede etkili sonuçlar elde etmeye başarısı ile bu alandaki araştırma ve uygulamalar artmıştır [1,2].

Hareket tespiti, günümüzde bilgisayarlı görü alanlarında sıklıkla çalışılan önemli konulardan bir tanesidir [2]. Hareket tespiti, bir kaynaktan (kamera, sensör gibi) alınan veriler üzerinde bir noktadan başka bir noktaya hareket eden nesnenin hareketini, hızını ve durumunu izleme olarak tanımlanabilir. Hareket tespiti sırasında oluşabilecek ışık değişimi ve yetersizliği, nesnenin doğrusal olamayan hareketleri, nesnenin hızlı yer değiştirmesi, nesne veya ortamın yapısal bozulması, nesnenin kameran açısının dışına çıkması gibi sorunlara karşı hareket tespiti başarılı bir şekilde çalışması son derece önemlidir [3].

Hareket tespit yöntemlerinin başarılı sonuçlar elde edilmesi için son dönemlerde yapılan çalışmaların incelenmesi yorumlanması, karşılaştırılması açısından değerlidir. Hareket tespitinin başarılı sonuç elde edilebilmesi için birçok farklı yöntem geliştirilmiştir. Bu yöntemler, video verisi üzerinde art arda gelen iki görüntünün arasındaki farkın alınmasından, en popüler konulardan olan derin öğrenme denetimli ve denetimsiz öğrenme yöntemlerine varıncaya dek birçok yöntem geliştirilmiştir [4].

Yapay zeka, makine öğrenmesi ve derin öğrenme gibi alanların gelişmeleri, donanım (Graphics Processing Unit, Tensor Processing Unit) gücüne bağlı olarak önemli noktalara gelmiştir [5]. Bu disiplinler, farklı alanlarda kullanılmaya ve insanın zorlandığı veya erişemediği alanlarda çözümler üretmeyi başarmıştır. Günümüzde önemli bir yeri olan hareket tespiti, derin öğrenmenin gelişimi ile karmaşık yapılardaki klasik yöntemlere oranla daha başarılı sonuçlar elde etmeyi başarmıştır. Klasik yöntemlere derin öğrenme tabanlı yöntemler uygulanarak sonuçların performansı artırılabilir.

Optik akış, videoda art arda gelen görüntüdeki piksellerin yer değiştirmesi olarak tanımlanmaktadır. Optik akış hesaplaması, hareketin yön ve hız şiddetini belirlemede kullanılan yöntemlerin temelidir [6]. Optik akış, iki boyutlu görüntüler üzerinde ve eşleşen pikseller odaklı çalışmaktadır. Hareket tespiti yöntemlerinin birçoğu optik akışı baz alınarak geliştirilmiştir. Klasik optik akış yaklaşımları iki ana varsayım üzerine dayanmaktadır. Bunlardan ilki, pikselin iki farklı resimde aynı renge sahip olması; ikincisi ise, hareketin düşük hızda olma varsayımdır. Bu iki

varsayım optik akış denkleminin elde edilmesini sağlamaktadır. Optik akış tahmini, seyrek ve yoğun optik akış olmak üzere iki ana gruba ayrılmaktadır. Seyrek optik akışta, belirli pikseller(grup) üzerinde işlem yapılırken; yoğun optik akışta ise, tüm pikseller için hesaplama yapılmaktadır. Yoğun optik akışa, Farneback [7] seyrek optik akışa da Horn-Schunck (HS) [8] ve Lucas-Kanade (LK) [9] yöntemleri örnek verilebilir. HS ve LK tarafından yapılan çalışmalar, hareket tespiti ile ilgilenenlerin dikkatini çekmiş ve optik akış tahmini konusunda çalışmalara yön vermişlerdir. Optik akış denklemi, iki bilinmeyenli tek bir denklemden oluşmaktadır. Bu iki bilinmeyenli denklemin çözülmesi için birden fazla denkleme ihtiyaç vardır. İki bilinmeyenli denklemin çözümü için geleneksel kabul edilen LK ve HS yöntemleri, geleneksel yöntemler olarak karşımıza çıkmaktadır. İki geleneksel yöntem de farklı varsayımlar tanımlayarak iki bilinmeyenli denklemi çözmüşlerdir. LK yönteminin temel aldığı varsayım, küçük bir pencerede optik akışı sabit kabul etme üzerine kurulmaktadır. Horn Schunck yöntemi ise, komşu piksellerin aynı akışa sahip olduğu varsayımı ile hesaplama yapmaktadır.

Derin öğrenme ile optik akış hesaplama yaklaşımlarından önce klasik yöntemlerin dışında optik akış için birçok yöntem önerilmiştir. DeepFlow [10] ve EpicFlow [11] yaklaşımları optik akış hesaplamasında yüksek başarılar elde etmiş yöntemlerdir. Ancak derin öğrenme tabanlı optik akış ile hareket tespiti yöntemlerinin bulunması ve başarıları ile bu yaklaşımları tercih edilen yöntemler arasından kalmayı başaramamıştır.

Evrişimli sinir ağları (CNN) son zamanlarda, özellikle hareket tespiti ile ilgili çeşitli bilgisayarlı görü görevlerinde başarılı sonuçlar elde etmiştir. Derin öğrenmede verinin eğitimi, optik akışın kalitesini artırmaktadır. Bundan dolayı büyük veri setleri üzerinde çalışmalar yapılmıştır. Optik akış tahmini için denetimli ve denetimsiz öğrenme yöntemlerinden birçoğu kullanılarak başarılı sonuçlar elde edilmiştir. Denetimli öğrenme yaklaşımlarından olan FlowNet [12] ve FlowNet2 [13] optik akış tahmini için evrişimli sinir ağını kullanan öncü çalışmalardır. Daha sonra geliştirilen yöntemler arasında en dikkat çeken yaklaşımlar, PWC-Net [14], DeepFlowNet [15], RAFT [16] ve SPyNet [17] gibi optik akış hesaplama yöntemleridir. Derin öğrenme ile optik akışın hesaplanmasında, özellikle elde edilmesi zor olan büyük miktarlarda eğitim verisi gerekmektedir. Bundan dolayı sentetik veriler üretilip bu veriler üzerinde eğitimler yapılmıştır. FlowNet, PWC-Net ve daha pek çok farklı testlerde birbirlerinden daha iyi performans gösteren değişkenlerle birlikte, optik akış problemlerini derin öğrenme ile çözmek günümüzde son derece popüler bir konu olmuştur [18]. Bu popüler yaklaşımlar özet olarak aşağıdaki gibidir:

Dosovitskiy ve arkadaşları [12] denetimli olarak optik akış tahmin problemini çözeabilen ilk uygun CNN yaklaşımını geliştirmiştir. Bu yaklaşıma FlowNet adını vermişlerdir. Yapılan çalışmada iki mimari önerilmiş ve bu iki mimari birbiri ile karşılaştırılmıştır. Birincisi, tek girişli ama art arda iki resim alarak çalışan genel mimari; ikincisi ise, aynı anda iki görüntü girişi alarak özellik vektörlerini ilişkilendiren bir mimaridir. Gerçek veri setleri bir CNN'yi eğitmek için

yeterince büyük olmadığından, sentetik “Uçan Sandalyeler” veri seti oluşturmuşlardır. Farklı veri setleri üzerinde başarımlar elde edilmiştir. Yaptıkları çalışmada evrişimli sinir ağ mimarilerinin tasarımındaki son gelişmelere dayanarak, iki giriş görüntüsünün optik akışını doğrudan tahmin etmek için bir ağ eğitmenin mümkün olduğunu göstermişlerdir. Uçan Sandalyelerin test setinde CNN'ler DeepFlow [10] ve EpicFlow [11] gibi en gelişmiş derin öğrenme tabanlı olmayan optik akış yöntemlerinden bile daha iyi performans elde edildiği görülmektedir.

İlg ve arkadaşları [13], 2016 yılında optik akış yöntemini eğitim verisi üzerine odaklanarak doğruluk ve hızda önemli iyileştirmeler elde etmişlerdir. Model mimarisinde FlowNet mimarisinde kullanılan FlowNetS ve FlowNetC ağ yapıları farklı işlemlere tabi tutularak akışı elde edilmiştir. FlowNet2, FlowNet mimarisinin önerenler tarafından oluşturulmuş ve FlowNet'ten daha başarılı sonuçlar gösteren bir yöntemdir.

Sun ve arkadaşları [14] 2017 yılında kompakt ama etkili bir CNN modeli tasarlamışlardır. Bu modele PWC-Net adını vermişlerdir. Özellik ve resim piramit yaklaşımını kullanarak optik akışı elde etmişlerdir. FlowNet modeline göre PWC-Net modeli %11 daha doğrulukla ve iki kat daha hızlı olarak sonuçlar elde etmişlerdir. Çalışmalarında optik akış tahmini yapmak için CNN'lerin yakından ilişkili iki önemli yönü olan model ve eğitim üzerine odaklanmışlardır.

Ranjan ve arkadaşları [17], klasik bir uzaysal piramit formülasyonunu derin öğrenme ile birleştirerek SpyNet adını verdikleri yeni bir derin öğrenme tabanlı optik akış yöntemi geliştirmişlerdir. Geliştirilen bu yöntem, uçtan uca denetimli öğrenme yapısıyla çalışmaktadır. Uzamsal piramidin her seviyesinde, yalnızca derin ağı eğitmek yerine akışı tahmin etmek için derin bir sinir ağı eğitmenin daha verimli olacağını önermişlerdir. Yöntemleri, her ağın yapacak daha az işi ve ağdaki hareketin az olması sebebiyle büyük hareketler için etkili sonuçlar üretmiştir. Yaptıkları çalışmada FlowNet'ten daha verimli sonuçlar elde etmiş ve klasik optik akış yöntemlerini derin öğrenme ile birleştirerek yeni yaklaşım ortaya çıkarmışlardır.

Wang ve arkadaşları [19], bir görüntü karesi içerisindeki hareketi tespit edilmesi istenen nesneyi derin öğrenme tabanlı yöntem ile tespit etmişlerdir. Bunu yaparken iki önemli problem ile karşılaşmışlardır. Birincisi nesnenin bulunduğu video karesinde nesnenin çözünürlüğün yeterli olmaması ve bu durumdan dolayı nesne takibini zorlaşması. İkinci olarak ise takip edilmek istenen nesnenin video içerisinde çok hızlı hareket etmesidir. Bu problemleri verileri uzamsal ve zamansal düzlemlerde yakalayarak çözmüşlerdir.

Zhao ve arkadaşları [20], 2020 yılında MaskFlowNet adı verdikleri bir yaklaşım önermişlerdir. Yöntemleri akış etiketleri ile tıkanma (Occlusion) problemini çözmeyi hedeflemişlerdir. Böylelikle görüntülerde oluşan tıkanma alanları yok sayılarak daha başarılı optik akış değerleri elde edilmiştir. PWCNet yaklaşımını baz alarak geliştirmiş bir modeldir.

Teed ve arkadaşları [16], 2020 yılında SINTEL [21] veri seti kıyaslamalarına göre iyi sonuçlar elde etmiş RAFT adında optik akış yaklaşımı önermişlerdir. CNN ve RNN mimarilerinin

birleşimi optik akış tespit edilmektedir. Çalışmalarında RAFT-S ve RAFT adında iki farklı mimari geliştirmişlerdir. İki mimarinin ortak zemini olmasına rağmen RAFT-S, RAFT'a göre daha basit bir mimaridir. FlowNet2 yaklaşımına göre daha başarılı sonuçlar üretmiştir.

Ronneberger ve arkadaşları [22], genel bir CNN yöntemi kullanarak ham görüntülerden optik akışı tahmin etmenin uygulanabilirliğini gösterdikleri U-Net adında bir ağ önermişlerdir.

İnsansız hava araçları (İHA) çok çeşitli sivil ve askeri amaçlarda popülerlik kazanmaya başlamıştır. Dong Hye D. ve arkadaşları [1], yaptıkları çalışmada, kalabalık bir hava sahası ortamında çarpışma önleme sistemlerinin geliştirilmesinde başarı elde etmek adına derin öğrenme tabanlı bir yaklaşım sunmuştur. Yaptıkları çalışmada farklı bir İHA'ya monte edilmiş tek bir kameradan İHA'ları tespit etmek ve izlemek için yeni bir yaklaşım sunmaktadırlar. Kameradan hareket tespiti ve takibi için ilk olarak, arka plan hareketlerini bir perspektif dönüşüm modeli ile tahmin edilip ve ardından hareketli nesne adaylarını derin öğrenme aracılığıyla belirmemişlerdir.

Çalışmalar genel olarak sentetik veriler üzerinde eğitilmiştir. Sentetikte verileri ile gerçek dünya verilerini kıyaslamak için yeterince gerçek dünya verisi bulunmamaktadır. Ayrıca eğitim verilerini oluşturmak ve ağı eğitmek yüksek teknoloji istemektedir. Bundan dolayı araştırmacılar denetimsiz öğrenme yöntemleri üzerine çalışmalar yürütmektedir. Amaçları herhangi bir videodan optik akışı hesaplamaktır. Bu yaklaşımlar denetimsiz öğrenme yöntemlerinin temelidir. Etiketli veri olmadan ağı öğrenmesi ve sonuçlarını etiketli veriler üzerinde test edilmesidir. Bu popüler yaklaşımlar özet olarak aşağıdaki gibidir:

J.Janai ve arkadaşları [23], 2018 yılında art arda üç görüntü üzerinde akış tahmini bulma yaklaşımı sunmuşlardır. Üç resimden birini (ortadaki) görüntü referansı olarak alınarak önceki (geçmiş) ve sonraki (gelecek) görüntüler üzerinde akış hesabı yapılmaktadır. Üç görüntü alınmasının asıl amacı tıkanma problemi denilen optik akış tahmin sorununu ortadan kaldırmaktır. Üç görüntüden birçok olası durumu ele almışlardır. Pikselin durumunu bu olasılıklarla elde edilmiştir. PWC-Net mimarisini alıp üzerinde değişiklik yaparak başarılı sonuçlar elde etmişlerdir.

2019 yılında Liu ve arkadaşları [24] tarafından DDFlow adında bir model önerilmiştir. Aynı iki model oluşturulup öğretmen ve öğrenci modeli adını vermişlerdir. Bu öğretmen ve öğrenci modelleri birbirine yakın işlemler yapmışlardır. Öğretmenden gelen veriler öğrenciler için gerçek akış olarak değerlendirmiş ve optik akışı elde etmişlerdir.

DDFlow mimarisini oluşturan ekip tarafından yine Liu ve arkadaşları [25], SelfFlow adı verdikleri bir model geliştirmişlerdir. Yine DDFlow mimarisinde olduğu gibi öğretmen ve öğrenci yapılı bir model geliştirmişlerdir. Süper piksel kavramı üzerine durmuşlardır. Süper pikselleri hareket ettirerek öğrenci yapısı için yine gerçek akış olarak değerlendirip optik akış ile hareket tahmini yapmışlardır.

Maurer ve arkadaşları [26], çalışmalarında, her kare için hareket modelini ayrı ayrı tahmin eden evrişimli bir sinir ağına (CNN) dayalı çevrimiçi öğrenme yaklaşımı önermişlerdir. Tahminden

önce bir ağı eğiten diğer yaklaşımların aksine, ProFlow adı verdikleri modelde modeli denetimsiz olarak öğrenmesi sağlanmaktadır.

Memisevic ve arkadaşları [27], görüntü dönüşümlerini denetimsiz bir şekilde öğrenmek için geçitli sınırlı Boltzmann makinesini önermektedir. Yaklaşımlarında olasılık tabanlı bir model tasarlamışlardır.

Hareket tespitinin temellerinden birini oluşturan optik akış yönteminin derin öğrenme ile birleşmesi sonucu elde edilen başarı çalışmanın temelini oluşturmaktadır. Sadece optik akış tabanlı yapılan çalışmalara göre derin öğrenme tabanlı optik akış yaklaşımları, hareket tespiti için daha başarılı çıktılar üretmektedir.

Çalışmanın amacı, derin öğrenme tabanlı optik akış ile hareket tespiti yapılan ve başarılı sonuçlar elde edilen çalışmaların karşılaştırılması ve karşılaştırmalardan elde edilen çıktıların farklı video görsellerine uygulanarak performans kriterleri açısından değerlendirilmesi ile literatüre katkıda bulunmaktır. Çalışma sonucunda elde edilen çıktıların bu alanda daha sonra yapacak araştırmacılara katkıda bulunulacağı öngörülmektedir. Ayrıca derin öğrenme tabanlı optik akış ile hareket tespiti yapan çalışmalara, en güncel çalışmaları ilave edilerek mevcut literatür çalışmalarının bir araya getirilmesi ve yüksek başarı sergileyen çalışmalara dikkat çekilmesi amaçlanmaktadır.

Bu çalışmada, ilk olarak geleneksel yaklaşımlardan günümüz derin öğrenme tabanlı yaklaşımlara kadar hareket tespitinin temelini oluşturan optik akış hesaplama yöntemleri ile ilgili literatür taraması yapılmıştır. Geleneksel yaklaşımlardan olan LK, HS, Farneback yöntemleri farklı hareketli görüntülere uygulanarak elde edilen sonuçlar karşılaştırılmıştır. Ayrıca derin öğrenme tabanlı yaklaşımlarından FlowNet, FlowNet2 ve RAFT yöntemleri MPI Sintel [21], KITTI [33] veri setleri üzerinde eğitilerek Uç Nokta Hatası (EPE) ve Kayıp (loss) değerleri farklı hiper parametreler kullanılarak grafik, tablo ve görsel çıktılar vasıtasıyla karşılaştırılmıştır.

Genel başlıklarıyla;

Tezin ikinci bölümünde optik akış ve derin öğrenme ile hareket tespiti yapan çalışmalara ait literatür taraması yer almaktadır.

Üçüncü bölümünde yapay sinir ağları (YSA) ve derin öğrenme detayları anlatılmaktadır. Derin öğrenme tabanlı optik akış ile hareket tespiti yöntemlerinde kullanılan kavramlara yer verilmiştir. Derin öğrenme mimarileri ve ağ yapıları anlatılmaktadır.

Dördüncü bölümde optik akış, optik akış yön ve şiddet haritası, optik akış denklemi, seyrek ve yoğun optik akış, klasik optik akış yöntemleri olan HS metotları ve LK metotları detayları ile anlatılmaktadır. Klasik yöntemlere uyguladığımız çıktı sonuçlarına da yer verilmiştir.

Beşinci bölümde derin öğrenme tabanlı optik akış ile hareket tespiti yapan ve başarılı sonuçlar üreten yaklaşımların uygulaması ve çıktıları bulunmaktadır. Bu bölümde, FlowNet, FlowNet2 ve RAFT mimarileri eğitim çıktıları ve farklı görseller üzerinde uygulamaları yapılmış

ve performans deęerlerine yer verilmiřtir. Ayrıca performans parametreleri, geliştirme ortamı ve optik akışta kullanılan veri kümelerine yer verilmiřtir.

Altıncı bölümde optik akış hesaplamalarında kullanılan klasik yöntemler ve derin öğrenme tabanlı yöntemlerin bulguları tartışılmış ve çıktıları yorumlanmıştır.

Yedinci bölümde ise çalışmada elde edilen sonuçlara yer verilmiştir.



2. LİTERATÜR ÖZETİ

Hareket tespiti güvenlik ve askeri alanlar gibi birçok alanda uygulanmaktadır. Hareket tespiti sırasında oluşabilecek ışık değişimi ve yetersizliği, nesnenin doğrusal olamayan hareketleri, nesnenin hızlı yer değiştirmesi, nesne veya ortamın yapısal bozulması, nesnenin kamera açısının dışına çıkması gibi durumlara karşı hareket tespiti başarılı bir şekilde çalışması son derece önemlidir. Başarılı hareket tespit hesaplamalarından dolayı uzun süredir hareket tespiti üzerine çalışmalar yürütülmektedir [28].

Literatürde çeşitli hareket takip yöntemleri üzerine çalışmalar bulunmaktadır. Video verisi üzerinden art arda gelen iki görüntünün birbirinden farkının ortaya konulmasından en popüler konulardan olan derin öğrenme yöntemlerine varıncaya dek birçok yöntem bulunmaktadır. Geçici değişiklikler yöntemi, arka görüntü parça çıkarımı ve optik akış yöntemleri literatürde en sık kullanılan yöntemler olarak dikkat çekmektedir. Geçici değişiklikler yöntemi, iki görüntü karesi arasındaki birim zamandaki farkı alarak değişimi gözlemlemektedir. Arka görüntü parça çıkarma yönteminde arka plan çıkarma prensibi temel alınmaktadır. Optik akış yönteminde ise piksel tabanlı çıkarımlar yapılmaktadır. Bu yöntemlerin kullanım oranları ve başarıları farklılık göstermektedir [3].

Optik akış yöntemi hareket tespiti için kullanılan klasik yöntemlerin temelini oluşturmaktadır. Optik akış, genel olarak hareket yönü ve şiddetini tespit etmek için kullanılmaktadır. Ayrıca pikselin renk ve parlaklığına bağlı olarak hareketin hızı belirlenmektedir. Piksel tabanlı olmasından dolayı bu yöntemde diğer yöntemlere göre daha doğru ve tutarlı sonuçlar elde edilmektedir.

Derin öğrenme yaklaşımı kullanılmadan önce optik akışı tahmini için birçok çalışma yapılmıştır. Çalışmalarda amaçlanan tahmin kalitesini iyileştirmektir. Yapılan çalışmalardan bazıları aşağıda verilmiştir;

Simioncelli ve arkadaşları [29], optik akış hesaplaması için olasılık tabanlı bir yaklaşım sunulmuştur. Çalışmada basit bir Gauss gürültü modeli kullanılarak optik akışın hesaplanmasındaki belirsizlik probleminin üstesinden gelmişlerdir. Sonuç olarak hem sentetik hem de gerçek görüntüler üzerinde başarılı sonuçlar elde edilmiştir.

Roth ve arkadaşları [30], derinlik haritalarından oluşturulan dizileri kullanarak optik akışın uzamsal istatistiklerini incelemişlerdir. Çalışmada makine öğrenmesi yaklaşımları ile optik akışın hesaplanmasında yeni bir yaklaşım geliştirilmiştir.

Sun ve arkadaşları [31], optik akışın hesaplamasında, gerçek optik akış alanlarıyla görüntü dizilerini kullanarak hem parlaklık sabitliği hem de optik akışın uzamsal özelliklerinin istatistiksel yöntemlerle tespit etmişlerdir. Çalışmada optik akışın eksiksiz bir olasılık modelini elde etmişlerdir.

Revaud ve arkadaşları [11], EpicFlow adında yeni bir optik akış tahmini yaklaşımı sunmuşlardır. EpicFlow derin öğrenme tabanlı bir yaklaşım olmasa da sonrasında derin öğrenme tabanlı birçok optik akış tahmini uygulamasında kullanılmıştır.

Wulff ve arkadaşları [32], MPI Sintel [21] ile KITTI [33] veri setlerinde EpicFlow [11], DeepFlow [15] gibi yaklaşımlardan daha başarılı sonuçlar üreten PCA-Flow adında seyrek optik akış hesaplama yöntemi geliştirmişlerdir. Seyrek eşleşme kullanıldığı için yaklaşım diğer yaklaşımlara göre daha hızlıdır.

Xu ve arkadaşları [34], dört boyutlu maliyet hacmi üzerinde çalışan optik akış tahmini yaklaşımı sunmuşlardır. Yapılan çalışmada, en gelişmiş optik akış yöntemlerinden önemli ölçüde daha yüksek doğruluk elde ederken çoğundan daha hızlı tahmin elde ettiğini gözlemlemişlerdir. Yaklaşımları hem Sintel hem de KITTI kıyaslamalarında yayınlanan tüm klasik optik akış yöntemlerinden daha iyi performans göstermektedir. Çalışmalarına DCFlow adını vermişlerdir.

Derin öğrenme yaklaşımları, bilgisayarlı görü alanındaki problemleri etkili bir şekilde çözmektedir. Yaklaşımların problem çözmedeki elde ettiği etkili sonuçlar bu alana olan ilgiyi artırmıştır. Derin öğrenme, bilgisayarlı görü, işaret işleme, yüz tanıma, ses tanıma, sürücüsüz araç kullanma ve dil işleme gibi alanlarda etkin olarak kullanılmaktadır. Klasik yöntemlere derin öğrenme tabanlı yöntemler uygulanarak sonuçların performansı artırılabilir [35-37].

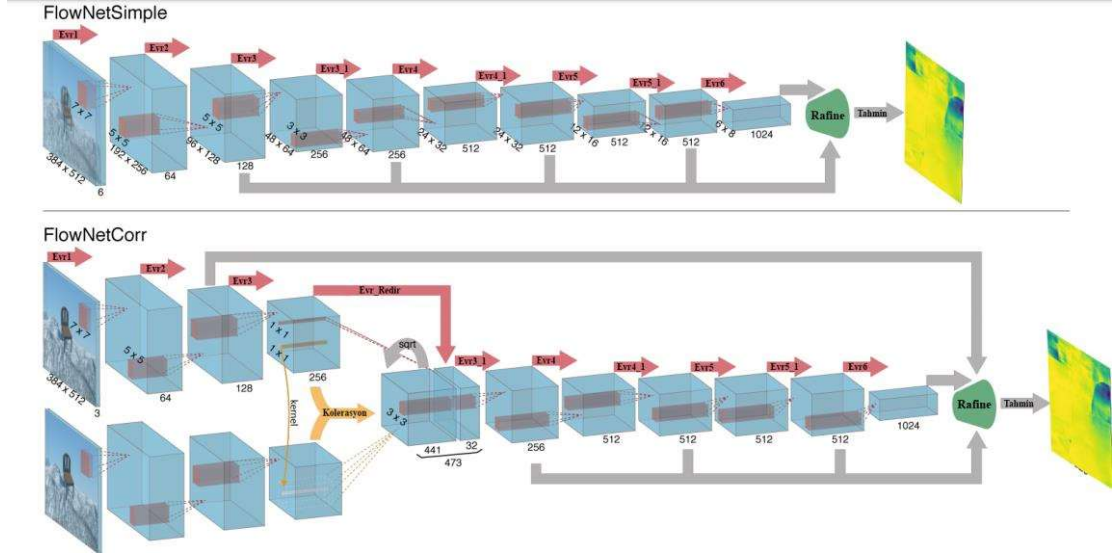
Hareket takip yöntemlerinin meydana gelen olumsuz durumlara karşı hızlı ve etkili sonuçlar göstermesi beklenmektedir. Son yıllarda optik akış ve derin öğrenme yöntemlerine dayalı çeşitli yöntemler geliştirilmiştir. Bu yöntemler sayesinde uygulanan derin öğrenme tabanlı yaklaşımlar ile optik akış tahmini birçok geleneksel akış yönteminden daha iyi performans göstermeye başlamıştır.[38]

Geleneksel yöntemlerle karşılaştırıldığında, derin öğrenme tabanlı yöntemlerin kalitesinde büyük bir iyileşme sağlanmıştır. FlowNet ve FlowNet2 optik akış tahmini için Evrişimli sinir ağını (CNN) kullanan öncü çalışmalardır. FlowNet2, performans ve tutarlılığı artırmak için FlowNetC ve FlowNetS gibi FlowNet'in varyantları olarak geliştirilmiştir. Literatür çalışmasında FlowNet ve FlowNet2 modelleri yakından incelenmektedir. Derin öğrenme tabanlı yapılan çalışmalardan bazıları aşağıda verilmiştir. Bu çalışmalar denetimli öğrenme alt yapısıyla çalışmaktadır.

FlowNet mimarisi, optik akışı tahmin etmek için ilk CNN yaklaşımı olarak 2015 yılında geliştirilmiştir. Dosovitskiy ve arkadaşları [12], tarafından yapılan çalışmada iki mimari önerisinde bulunulmuş ve bu mimariler birbirleri ile karşılaştırılmıştır. Yöntem, uçtan uca denetimli öğrenme temeliyle çalışmaktadır. CNN mimarilerini sınıflandırma, derinlik tahmini ve anlamsal bölümlendirme işlemlerini temel alınarak oluşturulmuştur. Bunun sonucunda gözetimli öğrenme ile optik akış tahmin problemini çözebilen uygun CNN'leri inşa edilmiştir. Optik akış tahmini için FlowNetS (FlowNetSimple) ve FlowNetCorr adında iki sinir ağı geliştirilmiştir. Ayrıca temel olarak FlowNetS ve FlowNetCorr yöntemleri, kodlayıcı ve kod çözücü parçalardan oluşmaktadır.

Kodlayıcıda iki ardışık görüntüden özellikler çıkarılırken; kod çözücüde, kodlayıcı özellik haritalarıyla son optik akış tahminini elde edilmektedir. Derin öğrenme tabanlı optik akış ile hareket tespiti yapan ilk yaklaşım olmasından dolayı çalışmada FlowNetS ve FlowNetCorr ağırları daha detaylı bir şekilde ele alınmaktadır. FlowNetS (FlowNetSimple olarak da bilinir) kodlayıcı (coder) giriş verileri, iki ardışık görüntünün birleşiminden oluşmaktadır. Bu iki görüntü 6-kanallı tensöre yerleştirilir, burada ilk üç kanal ilk görüntüye, kalan üç kanal ikinci görüntüye aittir. Kodlayıcı kısmı birkaç evrişim katmanından ve ardından aktivasyon fonksiyonlarından oluşmaktadır. Bu yöntem, ağın iki yığılmış görüntünün nasıl işleneceğine ve ardından sonuç iyileştirmesi için özellik haritasını kendi kendine karar vermesine olanak tanır.

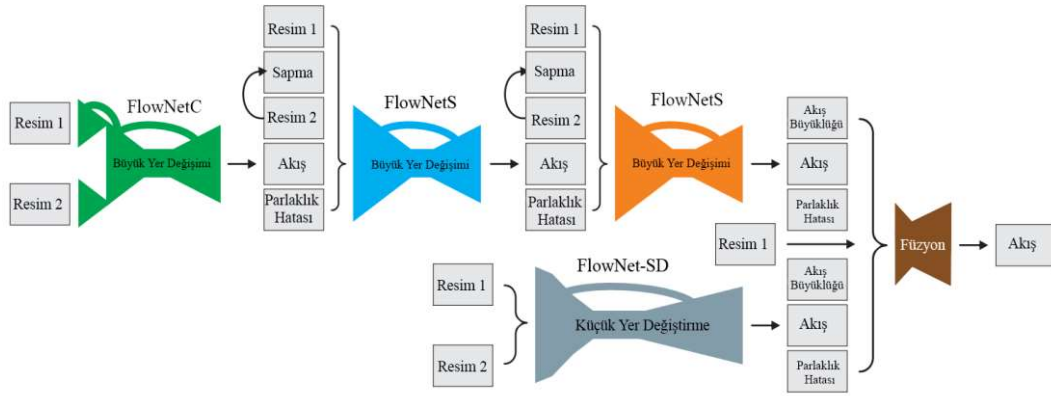
FlowNetCorr kodlayıcı; FlowNetCorr 'da giriş olarak her seferinde yalnızca bir görüntü almaktadır. Bundan dolayı girilen görüntüler kodlayıcıda depolanmaz. Bu aşada, birbirini izleyen iki görüntüden ayrı ayrı özellikler çıkarmak için CNN'in ilk aşamasında paylaşılan ağırlıklar kullanılır. Bir sonraki adım, sırasıyla birinci ve ikinci karelerden hesaplanan iki özellik haritasını birleştirmektir. Bunu sağlamak için, FlowNetS ve FlowNetCorr arasındaki temel fark olan Korelasyon katmanı adı verilen yeni bir teknik kullanılır. Korelasyon katmanı, iki özellik haritası arasında çarpma ve yama karşılaştırmaları yapmak için kullanılan bir katmandır. Korelasyon hesaplaması evrişim işlemine eşittir, ancak burada eğitilebilir çekirdek yoktur. Bir özellik haritası diğeriyle eşleştirilir. Yapılan iki özellik haritasının eşleştirme işleminden sonra, korelasyon sonucu, yüksek seviyeli özellikleri çıkarmak için geri kalan evrişim katmanlarına ileri besleme yapar. FlowNetS ve FlowNetCorr'un model yapıları Şekil 2.1'de verilmiştir. FlowNetS ve FlowNetCorr kod çözücü(decoder); Kodlayıcıdan alınan çıktı özelliği haritaları, giriş görüntü çözünürlüğünden 64 kat daha küçüktür. Bu nedenle sonucun çözünürlüğünün yükseltilmesi gerekmektedir. Bunun için doğrusal olmayan yukarı örnekleme yöntemi kullanılmaktadır. Eğitilebilir evrişim katmanları kod çözücü Optik Akış çıktısını büyütme için kod çözme aşamasında kullanılır. Gerçek veri setleri bir CNN'yi eğitmek için yeterince büyük olmadığından, sentetik "Uçan Sandalyeler" veri seti kullanılmışlardır. Sintel ve KITTI veri setleri kullanılarak başarımlar elde edilmiştir. Yapılan çalışmada evrişimli sinir ağ yöntemlerinin tasarımındaki son gelişmelere dayanarak, iki giriş görüntüsünün optik akışını doğrudan tahmin etmek için bir ağ eğitmenin mümkün olduğu iddia edilmektedir. Uçan Sandalyelerin veri setinde CNN'ler DeepFlow ve EpicFlow gibi derin öğrenme tabanlı olmayan optik akış ile hareket tespiti yapan en gelişmiş yöntemlerden bile daha iyi performans sergilenmiştir.



Şekil 2.1. FlowNetS ve FlowNetC model yapıları [12]

FlowNet yapısında her ağ, birinci görüntü ile ikinci görüntü arasındaki akış ayarlamasına katkıda bulunarak önceki akış alanını iyileştirir. Ancak sonuç olarak model birçok uygulamaya göre yavaş bir çalışma süresine sahiptir.

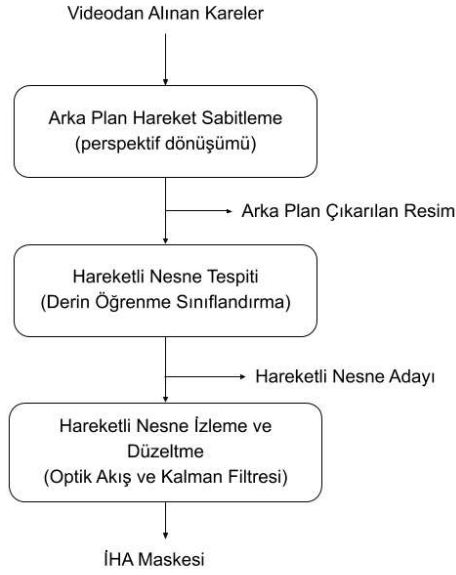
FlowNet2, FlowNet geliştiricileri tarafından önerilmiş bir yöntemdir. Ilg ve arkadaşları [13], FlowNet yaklaşımının optik akış tahmini için kullanılabileceğine ancak optik akış tahminin kalitesi hala geleneksel yöntemler daha başarılı sonuçlar verdiğini ifade etmişlerdir. Özellikle küçük yer değiştirmeler ve gerçek dünya verilerinde, FlowNet klasik yöntemlerle rekabet edemeyeceklerini ifade etmişlerdir. Yaptıkları çalışmada optik akışın uçtan uca (end-to-end) denetimli öğrenme modeli üzerinden ilerlenmemiştir. Çalışmalarında eğitim verisi üzerine odaklanmışlardır. Kullanılan eğitim verilerine bağlı olarak optik akış tahminin başarı oranının değiştiği gözlemişlerdir. Verilerin sadece türünün değil aynı zamanda eğitime alınma sıralarının da önemli olduğunu savunmuşlardır. FlowNet yönteminin küçük görüntülerde doğru çalışmadığı konusunda yapmış oldukları eleştiriye istinaden küçük hareketler konusunda başarılı sonuçlar üreten bir alt ağ modeli geliştirmişlerdir. FlowNet2 modeli Şekil 2.2’de verilmiştir. FlowNet2, orijinal FlowNet’ten yavaş olmasına rağmen tahmin hatasını %50’den fazla azaltmışlardır. Ancak FlowNet2 640 MB bellek alanı gerektirir ve mobil ve gömülü cihazlar için pek uygun olmadığı düşünülmektedir. Şekil 2.2’de görüldüğü gibi büyük yer değiştirmeli optik akışı hesaplamak için birden çok FlowNetS’i birleştirmişlerdir. Modelde bulunan parlaklık hatası, önceden tahmin edilen akışla eğilmiş ilk görüntü ile ikinci görüntü arasındaki farktır. Küçük yer değiştirmelerle en iyi şekilde başa çıkmak için FlowNetS mimarisine başlangıçta daha küçük adımlar ve yukarı dönüşümler arasında evrim eklemişlerdir. Son olarak, nihai tahmini sağlamak için küçük bir füzyon ağı uygulayıp tahmini elde etmişlerdir.



Şekil 2.2. FlowNet2 model yapısı [13]

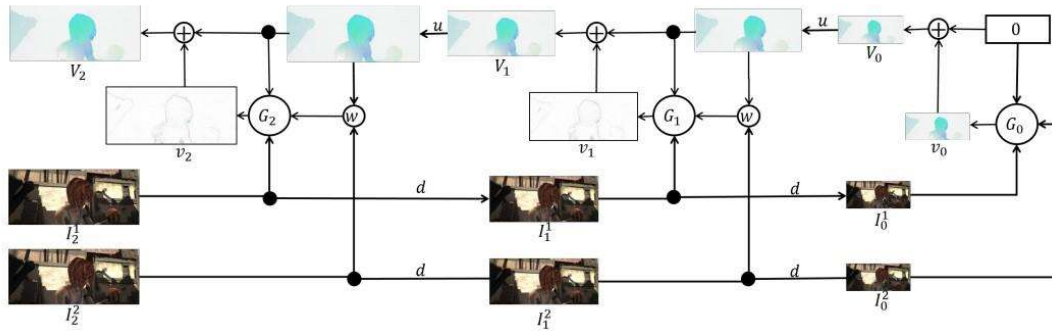
Hui ve arkadaşları [39] yeni bir CNN tabanlı optik akış ile hareket tespiti modeli önermişlerdir. Varyasyonlu yöntemlerden veri uygunluğu ve düzenliliği benimseyerek klasik optik akış tahmini sorununu ele almak için basit ve etkili bir CNN geliştirmişlerdir. Derin öğrenme ve pramit formülasyonunu birleştirerek yeni bir optik akış yöntemi sunmuşlardır. Pramidin her seviyesinde, yalnızca bir derin ağı eğitmek yerine ayrı ayrı ağ eğiterek optik akışı hesaplamışlardır. Yaptıkları çalışmada model boyutunu ve parametreleri küçültürerek FlowNet2'den daha az model boyutuna erişip FlowNet2'den yaklaşık 3.1 daha hızlı optik akış hesaplamayı başarmışlardır. Çalışmalarında LiteFlowNet'in ağ mimarisini ve eğitim protokollerini optimize ederek daha iyi bir akış doğruluğuna ve daha hızlı bir çalışma süresine sahip LiteFlowNet2'yi geliştirmişlerdir.

Ye ve arkadaşları [1], İnsansız hava araçları için yeni bir yaklaşım önermişlerdir. Yaptıkları çalışmada, ilk olarak perspektif dönüşüm modeli ile iki ardışık kare arasındaki arka plan hareketini tahmin edilmesi işlevi yapmaktadır. Daha sonra ortaya çıkan arka plandan çıkarılan görüntüden, ayırt edici yamalara derin öğrenme sınıflandırıcısı uygulayarak hareketli nesneleri tespit etmişlerdir. Daha sonrasında arka planı çıkarılmış yapıda optik akış ile hareket takibi yapılmaktadır. Yapılan çalışmanın modeli Şekil 2.3'te verilmiştir.



Şekil 2.3. İHA için optik akış ile hareket tespiti diyagramı

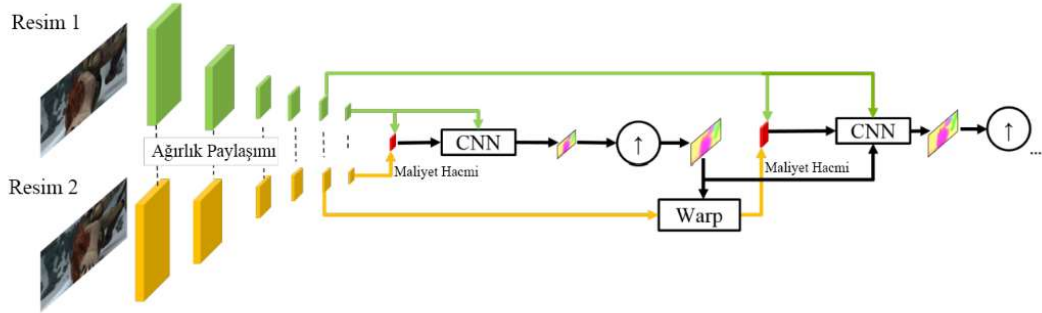
Ranjan ve arkadaşları [11], klasik bir uzaysal piramit formülasyonunu derin öğrenme ile birleştirerek SpyNet adını verdikleri optik akış modeli geliştirmişlerdir. Mimarileri uçtan uca denetimli öğrenme yapısıyla çalışmaktadır. Uzamsal piramidin her seviyesinde, bir sinir ağı eğitmişlerdir. Kullandıkları bu yöntem, büyük hareketler için başarılı sonuçlar elde etmiştir. SpyNet, klasik yöntemleri CNN'lerle birleştirmenin olası bir sonuç üretebileceğini göstermiştir. FlowNet ile karşılaştırıldığında, SPyNet çok daha basittir ve model parametreleri açısından % 96 daha küçüktür. Ayrıca, kıyaslamalarda SPyNet, FlowNet'ten daha doğru akış sonuçları üretmiştir. Modelin üç seviyeli piramit ağı yapısı Şekil 2.4'te verilmektedir.



Şekil 2.4. Üç seviyeli piramit ağı yapısı [11]

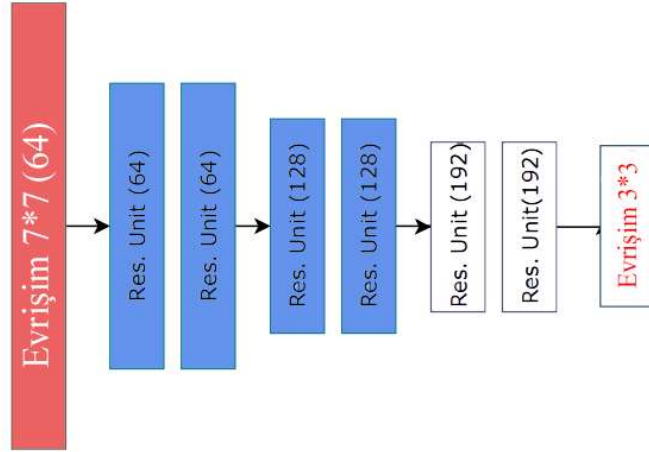
Sun ve arkadaşları [14], PWC-Net adında karmaşık ama etkili bir CNN modeli tasarlamışlardır. Optik akış tahmini için bir CNN modelinin hem doğruluğunu artırmak hem de boyutunu küçültmek amacıyla çalışmalarını geliştirmişlerdir. Mimarileri uçtan uca denetimli

öğrenme yapısıyla çalışmaktadır. Optik akış tahmini için modeller ve eğitim verileri üzerinde durmuşlardır. Çalışmada ilk olarak piramit şeklinde işleme, eğirme ve maliyet hacmi işleme gibi basit ve iyi belirlenmiş ilkelere göre yeni bir yaklaşım tasarlamışlardır. FlowNetS ve FlowNet2 ile olan performans açığının, klasik yöntemlerin kullanılmamasından kaynaklandığını düşünmüşlerdir. FlowNet2'nin bir alt ağı olan FlowNetC'yi yeniden eğitmek için aynı eğitim prosedürünü kullanmışlardır. Alan bilgisini derin öğrenmeyle birleştirmenin, optik akış tahmini hem boyutsal hem de performans olarak iyi sonuçlara ulaşabileceğini gözlemlemişlerdir. Optik akış için mevcut CNN modellerine göre model boyutunda ve doğruluğunda önemli iyileştirmeler elde edilmiştir. Yeniden eğitilen FlowNetC, Sintel finalde önceden eğitilmiş olandan %56 daha doğru ve hatta FlowNet2 modelinden yüzde beş daha doğru sonuçlar üretmiştir. PWC-Net'in boyutu FlowNet-2 göre 17 kat daha küçüktür. Çıkarımda 2 kat daha hızlı ve Sintel finalde son FlowNet2 modeline göre %11 daha doğru sonuçlar elde ettiklerini gözlemişlerdir. PWC-Net, son teknoloji sonuçlara ulaşmak için görüntü piramidi, eğrilme ve maliyet hacmi gibi çeşitli klasik optik akış tahmin tekniklerini uçtan uca eğitilebilir derin sinir ağlarında birleştirir. PWC-Net modelinin mimarisi Şekil 2.5'te verilmektedir.



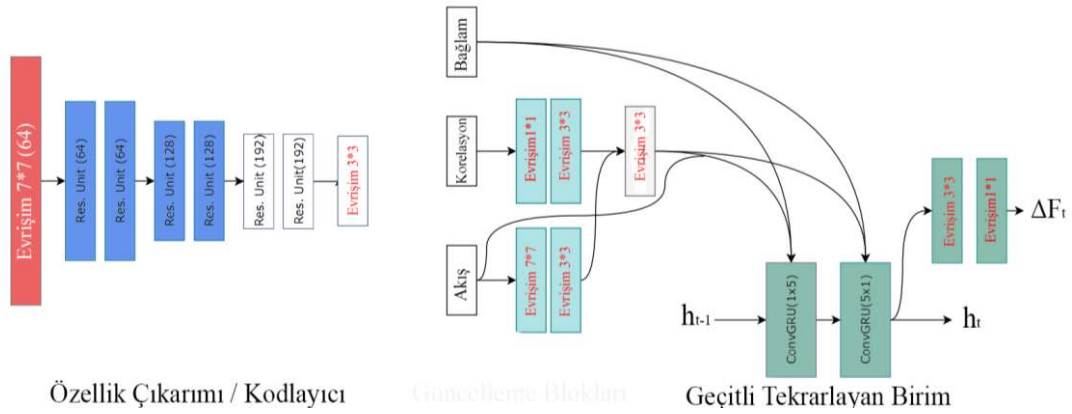
Şekil 2.5. PWCNet model yapısı [14]

Teed ve arkadaşları [16], 2020 yılında SINTEL kıyaslamalarına göre iyi sonuçlar elde etmiş RAFT adında CNN ve RNN mimarilerinin birleşimi ile model geliştirmişlerdir. RAFT yapısı üç aşamada gerçekleşmektedir. İlk olarak özellik çıkarımı aşaması gelmektedir. Özellik çıkarma aşamasında, modele iki ardışık görüntü verilir. Bu iki görüntüden özellik çıkarmak için, paylaşılan ağırlıklara sahip iki CNN kullanmışlardır. Yaklaşımları FlowNetCorr mimarisine benzer şekilde iki görüntünün ayrı ayrı özelliklerini bulmaktadır. CNN'in mimarisi, altı katmandan oluşur ve çözünürlük her iki katmanda yarı yarıya azalır ve artan sayıda kanal bulunur. Şekil 2.6'da görüldüğü gibi RAFT mimarisinin özellik çıkarma aşamaları klasik bir sinir ağı yapısındadır.



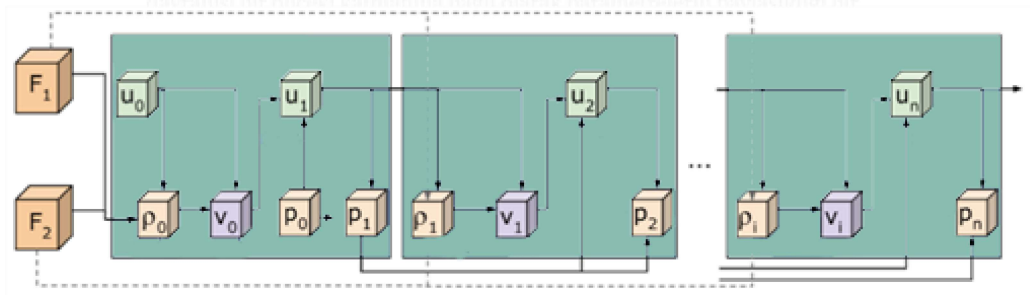
Şekil 2.6. RAFT özellik çıkarma ağı [16]

İkinci aşama olarak görsel benzerlik adımı gelmektedir. Görsel benzerlik, tüm özellik haritası çiftlerinin iç çarpımı olarak hesaplanır. Bu aşamada küçük ve büyük piksel yer değiştirmeleri hakkında önemli bilgiler veren Korelasyon hacimleri adlı bir 4 boyutlu tensör elde edilmiş olur. Bu yaklaşım, FlowNetCorr'daki Korelasyon katmanı ile karıştırılmamalıdır. RAFT'ta sabit boyutlu bir pencere olmadan iki özellik haritasının korelasyonunu hesaplanmaktadır. Korelasyon piramidi, ani hareketleri daha belirgin hale getirmek için çok ölçekli bir görüntü benzerlik özellikleri oluşturmak için kullanılır. Böylece piramit hem küçük hem de büyük yer değiştirmeler hakkında bilgi üretir. Üçüncü aşamada ise yinelemeli güncelleme adımı gelmektedir. Yinelemeli güncelleme, daha önce hesaplana tüm verileri birleştiren Geçitli Tekrarlayan Birim (GRU) hücreleri dizisidir. Bir güncelleme yinelemesi, her yeni adımda tahmini daha doğru hale getirmek için yeni optik akış güncellemesi üretir. RAFT mimarisi Şekil 2.7'de verilmektedir. Çalışmalarında RAFT-S ve RAFT adında iki farklı mimari geliştirmişlerdir. İki mimarinin ortak zemini olmasına rağmen RAFT-S, RAFT'a göre daha basit bir mimaridir.



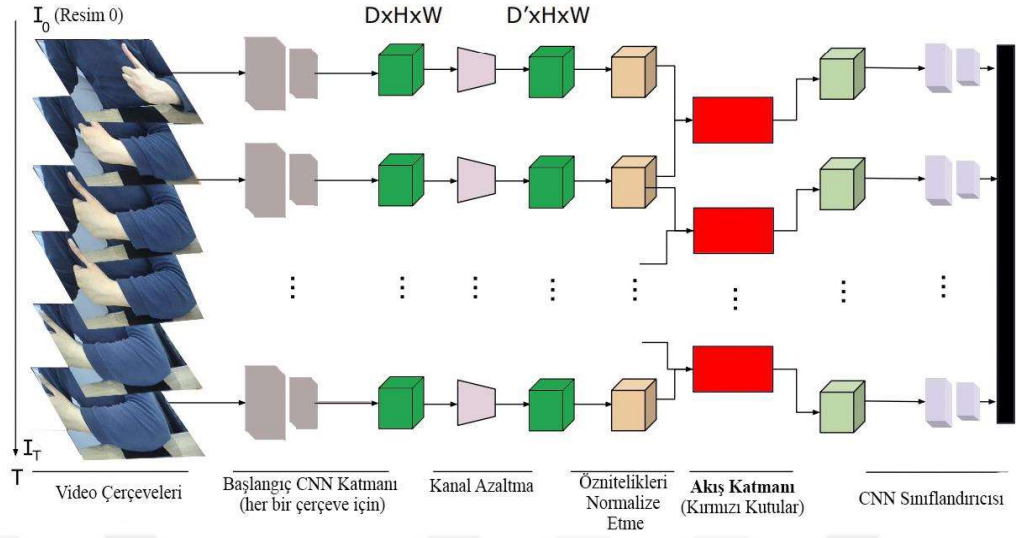
Şekil 2.7. RAFT'ın genel model yapısı [16]

Piergiiovann ve arkadaşları [40], optik akışı temel alan gösterim akışı (Representation flow) adı verilen bir CNN katmanı oluşturmuşlardır. Gösterim akış katmanı adı verdikleri katmanı modelin içerisine yerleştirerek video-CNN adında mimari geliştirmişlerdir. Optik akışı, hesaplama zorunda kalmadan hareket tespiti için ağa hareket gösterimlerini öğretmeyi başarmışlardır. Çalışmada bu katman, üç boyutlu RGB yerine daha düşük boyutlu CNN özellik haritasına uygulanmaktadır. Katman uçtan uca bir şekilde eğitilecek CNN modelinin içerisine yerleştirilir. Gösterim akış katmanı Şekil 2.8'de verilmiştir. Akış katmanı birden fazla yinelemeye sahiptir. Her katman bir önceki katmana bağlı bir şekilde çalışır. Bağlantılar ile parametrelerin paylaşımı sağlanır.



Şekil 2.8. Gösterim akış katman yapısı [40]

Geliştirdikleri katmanı Video-CNN mimari adı verdikleri bir mimarinin içerisine yerleştirerek optik akışı başarılı bir şekilde elde etmişlerdir. Şekil 2.9'da verilmiştir.



Şekil 2.9. Video-CNN mimari yapısı [40]

Bailer ve arkadaşları [41], FlowFieldsCNN adı verdikleri optik akış tahmini için CNN tabanlı bir parça eşleştirme yaklaşımı sunmuşlardır. Özellik haritalarının düşük geçişli filtrelemesinin CNN'ler tarafından oluşturulan özelliklerin sağlamlığını artırabileceğinden düşünerek, farklı görüntü ölçekleri ile CNN tabanlı özellikleri hesaplayarak yeni bir yöntem önermişlerdir. Çalışmalarında optik akış için özel olarak tasarlanmış yeni çok ölçekli özellik oluşturma yaklaşımı elde etmişlerdir. FlowFieldsCNN, seyrek eşleştirme için CNN özelliklerini öğrenir ve eşleşmeleri EpicFlow ile uygulatır. Yaklaşımları mevcut yaklaşımlardan daha iyi performans gösterdiğini belirtmişlerdir.

Optik akış tahmini için gerçek dünya verileri ile ağı öğrenmek daha doğru olduğunu düşünen araştırmacılar denetimsiz öğrenme yaklaşımına yönelmişlerdir. Optik akışı öğrenmek, mat olmayan nesneler, tıkanmalar, büyük hareketler ve aydınlatma değişikliklerinden kaynaklanan belirsizlikler nedeniyle karmaşıktır. Etiketli gerçek dünya eğitim verilerinin eksikliğiyle başa çıkmak için, bazı araştırmacılar denetimsiz öğrenmeye dayalı optik akış yöntemleri önermiştir. Etiketli eğitim verileri olan veri kümeleri üzerindeki denetimli yaklaşımlardan daha düşük başarılarla sahip olsa da etiketlenmemiş veriler üzerinde CNN modellerini (önceden) eğitmek için mevcut denetimsiz yöntemler geliştirilmeye devam edilmektedir [42]. Denetimsiz öğrenme tabanlı yapılan çalışmalardan bazıları aşağıda verilmiştir.

Meister ve arkadaşları [43], enerji tabanlı optik akış yöntemlerinden esinlenerek optik akışın denetimsiz öğrenilmesini sağlayan UnFlow adında yeni bir yaklaşım önermişlerdir. Optik akış için denetimsiz öğrenmenin etkinliğini gösteren uçtan uca denetimsiz bir yaklaşım sunmaktadırlar. Değiştirilen iki giriş görüntüsüyle ikinci bir geçiş gerçekleştirerek çift yönlü optik

akışı hesaplamaktadırlar. Yapılan çalışmada KITTI karşılaştırmalarında denetimsiz model daha önceki derin ağlara göre büyük başarılar elde edildiği gözlemlenmiştir.

Guan ve arkadaşları [44], PCLNet adlı denetimsiz bir optik akış tahmin yöntemini önermişlerdir. Önerilen yöntem, daha ayırt edici bir benzerlik haritası sağlayan bir polarizasyon şeması yapısıyla çalışmaktadır. Çalışmaları herhangi bir videodan çok çerçeveli optik akışların esnek bir şekilde tahmin edilmesine olanak tanıyan, bitişik çerçeve ile yeniden yapılandırma kısıtlamasıyla piramit Evrişim LSTM kullanır. Özellikler, tahmini akış verildiğinde eşleşen çiftler için yüksek benzerlik ve belirsiz çiftler için düşük olacak şekilde güncellenir. KITTI, Sintel ve Uçan Sandalyeler [12] veri setlerinde yöntemleri ile etkili sonuçlar elde etmişlerdir. Yapmış oldukları kapsamlı deneyler sonucunda yöntemlerinin yalnızca optik akışı etkili ve doğru bir şekilde tahmin etmekle kalmayıp, aynı zamanda eylem tanıma konusunda karşılaştırılabilir performans elde ettiğini göstermiştir.

Janai ve arkadaşları [23], çoklu görüntüler üzerinde denetimsiz öğrenme ile optik akış tahmini yapan bir yöntem geliştirmişlerdir. Birden fazla çerçevenin dikkate alınması ile bazı belirsizliklerin, özellikle de tıkanmalardan kaynaklanan optik akış tahmin sorunlarını çözülmesine yardımcı olacağını savunmuşlardır. Bu nedenle, akış alanlarını ve tıkanmaları birlikte tahmin etmek için evrişimli bir sinir ağını eğitmek için çok çerçeveli bir formülasyon önermişlerdir.

Fatma Güney ve arkadaşları [38], optik akış tahminin derin öğrenme yardımı ile tespitinden esinlenerek ayrık optimizasyon kullanarak optik akışı çözmek için yeni bir yöntem sunmuşlardır. Hızlı GPU matris çarpımını kullanarak, referans görüntüdeki her pikseli hedef görüntüdeki her pikselle karşılaştırarak özellik eşleştirme gerçekleştirmişlerdir.

Memisevic ve arkadaşları [45], görüntü dönüşümlerini denetimsiz bir şekilde öğrenmek için geçitli sınırlı Boltzmann makinesini kullanmışlardır. Yaklaşımlarında olasılık tabanlı bir model tasarlamaktadırlar. Model ayrıca daha önce işlenmemiş görüntüler üzerinde yeni dönüşümler tahmin edebilir ve denetimsiz bir şekilde optik akış tahmini yapabilmektedir.

Stefano Alletto ve arkadaşları [46], hareket tahmini için denetimsiz öğrenme tabanlı optik akış hesaplama yöntemi geliştirmişlerdir. Optik akışın birbirini izleyen iki video çerçevesi arasında geometrik bir eğme olarak kullanılabilceğini ve bu dönüşümü iki aşamada tahmin etmek için derin bir mimari tasarlayabileceğini göstermişlerdir. Çalışmada ilk olarak yoğun piksel tabanlı akış, geometrik uzaysal sınırlamalarla hesaplanır. Akış alanını hesaplamak için uzaysal transformatör katmanını kullanmışlardır. İkinci aşama da ise ikinci bir derin ağı besleyen yapı geliştirmişlerdir. TransFlow adını verdikleri model, diğer denetimsiz algoritmalara göre daha iyi performans göstermekte ve işlenmemiş verilerde hatada üç kat azalma göstermiştir. Ayrıca denetimli yöntemlere göre daha iyi tahmin elde edildiğini göstermişlerdir.

Long ve arkadaşları [47], çerçevelerin enterpolasyonu yöntemiyle görüntü eşleştirme yaparak denetimsiz öğrenmeye tabanlı bir model oluşturmuşlardır. Optik akış tahmini için CNN

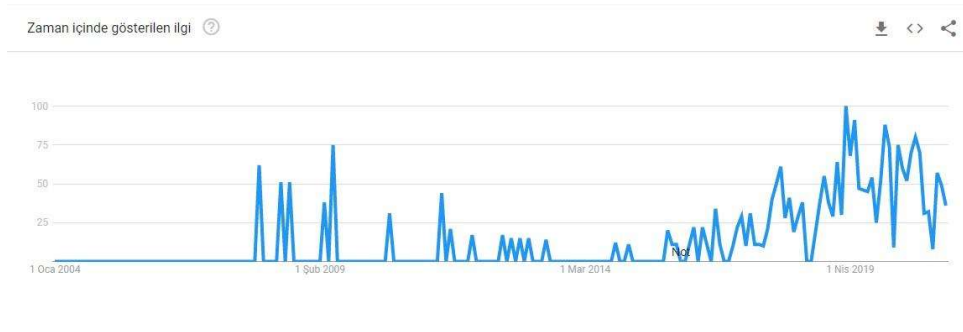
modellerini geliřtirmişlerdir. Çalışmalarında ilk olarak çerçeve enterpolasyonu için bir CNN eğitilmişlerdir. Ardından öğrenilen CNN’i ters çevirerek tekrardan verileri eşleştirip uygun optik akış değeri elde etmişlerdir. Çalışmada, çerçeve enterpolasyonu için CNN’nin gerçek dünya video dizilerinde doğal olarak bulunan görüntülerden yararlanılarak denetimsiz bir şekilde eğitilen yöntem geliřtirmişlerdir. Modelleri görüntü eşleřtirmeyi sadece videoları izleyerek öğrenmektedir. Yaklaşımları, geleneksel yöntemlere göre iyi performansa sahiptir.

Maurer ve arkadaşları [26], çalışmalarında her kare için hareket modelini ayrı ayrı tahmin eden CNN dayalı denetimsiz öğrenme yaklaşımı önermişlerdir. Tahminden önce bir ağı eğiten diğeryaklaşımların aksine, ProFlow adı verdikleri modelde modeli denetimsiz olarak, yani tahmin sırasında öğrenmesi sağlanmaktadır. Yaklaşımlarında her sekans için sadece bir model değil, her sekansın her çerçevesi için bir model öğrenilir. Çalışma prensibi olarak daha önce toplanan eğitim verileri kümesine dayalı olarak tüm görev için ağı bir kez eğitmek yerine, ağı her bir dizinin çerçevesi için çalışma zamanında ayrı ayrı eğitmişlerdir. Optik akış tahminlerine dayanan denetimsiz bir yaklaşım sunmuşlardır.

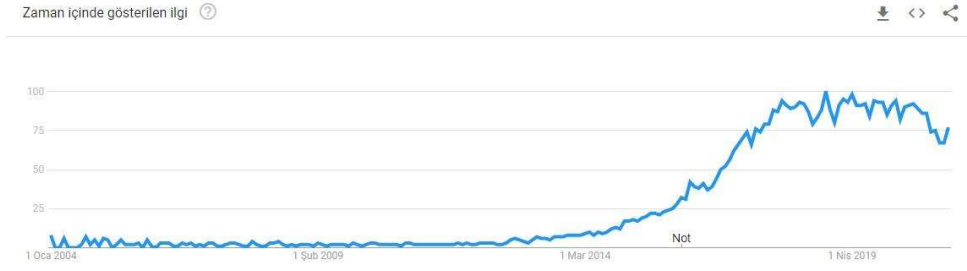
Gelişmiş derin modeller, genellikle önemli bir hesaplama karmaşıklığı ve zaman alan eğitim süreçleriyle doğru akış tahminine ulaşır. Kong L. ve arkadaşları [48], 2020 yılında gerçek zamanlı optik akış tahmini için basit ama etkili bir model sunmuşlardır. Evrişim işlemi için tam bağlantılı bir yapı önerilmektedirler. Hesaplama kaynaklarından tasarruf ederken çok ölçekli bilgileri verimli bir şekilde birleştirebilen optik akış tahmini için piramit ağının bir iyileřtirmesi olarak karmaşık ama verimli bir U-ağ modeli oluşturmuşlardır. FDFlowNet olarak isimlendirdikleri modelde PWC-Net’ten yaklaşık 2 kat daha hızlıyken KITTI ve Sintel kıyaslamalarında daha iyi veya benzer doğruluk elde etmişlerdir. Model LiteFlowNet’ten de yaklaşık 3,2 kat daha hızlı çalışmaktadır.

3. DERİN ÖĞRENME

Derin öğrenme, insan beyninin yapısından ve işlevinden ilham alan yapay sinir ağlarını kullanan bir makine öğrenimi alt alanıdır [49]. Öğrenme kavramının matematiksel modele dönüştürülmesi ve bilgisayar ortamına aktarımının yapılması sonucunda yapay zeka kavramı ortaya çıkmıştır. Yapay zeka, derin öğrenme, yapay sinir ağları gibi konular 1950’li yıllarında dönemin bilgi kuramı uzmanlarından A.M. Turing bu kavramların öncülüğünü oluşturmuştur. Daha sonrasında derin öğrenmenin gelişimi sırasıyla; 1955 J. McCharthy “Yapay Zeka” terimini ilk defa kullanmıştır. 1957 yılında perseptron(algılayıcı) yapısı ilk kez Frank Rosenblatt tarafından tanımlanmıştır. 1986 yılında geriye yayılım uygulaması H. Hilton tarafından geliştirildi. 1997 yılında IBM firması Deep Blue ile alakalı çalışmalar yapmışlardır. 1998 yılında Yann LeCup ve ark, CNN yapısı ile ilk defa 0-9 arası rakamları sayısallaştırmayı başarmışlardır. 2009 yılında etiketlenmiş veri seti olan IMAGENET ücretsiz bir şekilde herkesin erişimine açılmıştır. 2011 yılında “siri” ile yapay zeka iyice ün kazanmıştır. 2012 yılında derin öğrenme ALEXNET yapısı ile ön plana çıkmıştır. 2014 yılında GOOGLNET imageNet yarışmasında birinci olması ile beraber herkesin ilgisini çekmeyi başardı. 2014 yılında GAN yapısı Yoshua Bengio ve ark, tarafından insan yüzü transferini başararak derin öğrenmeyi çok popüler hale getirmişlerdir. 2017 yılında ALFHAGO yazılım Go oyununun dünya şampiyonunu yenerek önemli bir üne kavuşmuştur. 2019 yılında derin sinir ağları ile birçok başarıya imza atmıştır [50]. Teknolojinin gelişimi ve bilgiye erişimin kolaylaşmasından dolayı derin öğrenmenin Türkiye’de ve dünyada popülerlik kazandığı görülmektedir. Derin öğrenme anahtar sözcüğünün Türkiye’de ve Dünya’da aranması Şekil 3.1 ve Şekil 3.2’de verilmiştir.



Şekil 3.1. Derin öğrenme Google trend veri grafiği (Türkiye)



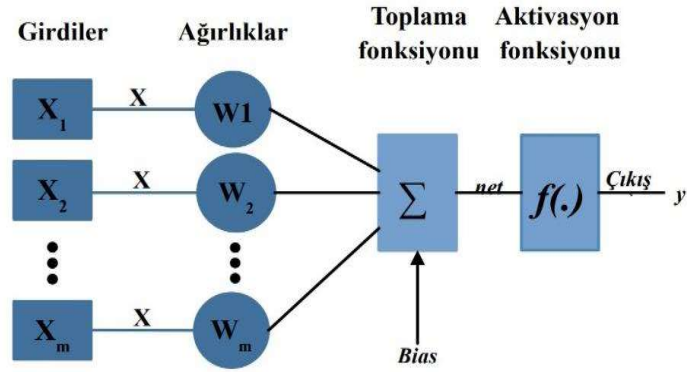
Şekil 3.2. Derin öğrenme Google trend veri grafiği (Dünya)

Şekil 3.1 ve Şekil 3.2’de verilen sayılar, arama ilgisini belirli bir bölge ve zaman için grafikteki en yüksek noktayla görece olarak gösterir. 100 değeri, terimin en yüksek popülerliğe sahip olmasıdır. 50 değeri, terimin bunun yarısı kadar popüler olduğu anlamına gelir. 0 değeri ise bu terim için yeterince veri olmadığı anlamına gelir.

3.1. Yapay Sinir Ağları

Yapay sinir ağları (YSA), canlıların öğrenme yapısının matematiksel olarak modellenmesi sonucu ortaya çıkan bir makine öğrenmesi tekniğidir. YSA’nın nöron yapısı Şekil 3.3’te verilmiştir. Bir işin, adım adım yapılması ile istenilen sonucu elde edilmesi işleme algoritma denilmektedir. Kullanıcıya göre farklılık gösteren yöntemler sabit bir algoritmanın oluşmasının önüne geçebilmektedir. Böyle durumlarda veriden yararlanarak istenilen işin çözülmesi için makinenin kendiliğinden öğrenmesini sağlanabilmektedir. Yapay sinir ağlarının, hesaplamalı finans, görüntü işleme, hesaplamalı biyoloji, enerji üretimi, otomotiv, havacılık, doğal dil işleme gibi birçok kullanım alanı bulunmaktadır [51-54].

Yapay sinir ağlarının çalışma yapısı, girdi değerleri ve ağırlıkların çarpılmasından sonra tüm ağırlık ve giriş değerlerinin toplanıp bias değeri eklenerek ve ardından aktivasyon fonksiyonu uygulanarak çıkış elde edilmektedir. Bu çıkış değeri bir sonraki sinir hücresine girebilir veya çıkış olarak kullanılabilir. Bir nöronun temel iç yapısının matematiksel karşılığı denklem 3.1 de verilmiştir.

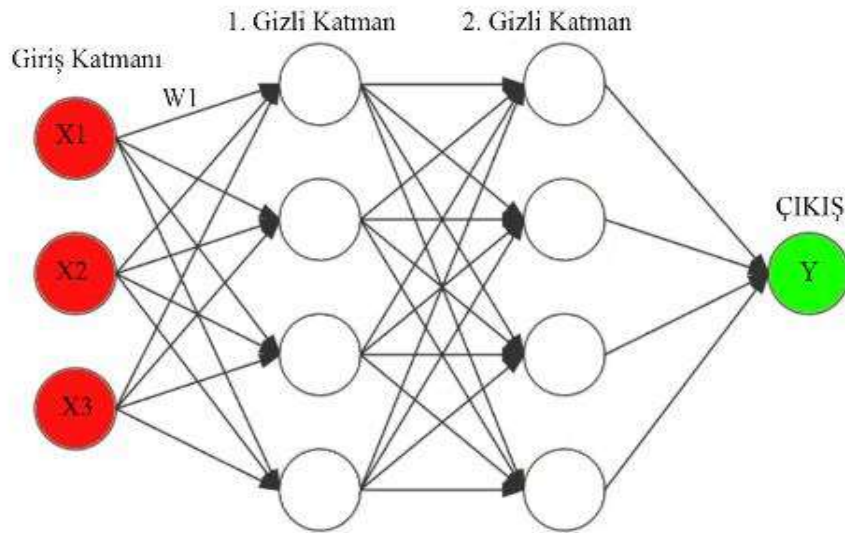


Şekil 3.3. YSA nöron yapısı

$$Y(\text{Çıkış}) = X(\text{Giriş}) * W(\text{Ağırlık}) + b(\text{bias}) \quad (3.1)$$

Her bir nöron ayrı ayrı hesaplanmaktadır. Nöronlar birbirleri ile seri ve paralel olarak bağlanmaktadır. Şekil 3.4'te Yapay sinir ağının genel yapısı verilmiştir. Her bir düğüm nöronu temsil etmektedir. Gizli katmanda bulunan her nörona, düğüm ve düğümlerin tümüne de katman adı verilmektedir. YSA'lar üç temel yapısı bulunmaktadır. Bunlar:

- Giriş katmanı,
- Ara – Gizli katman,
- Çıkış katmanı.



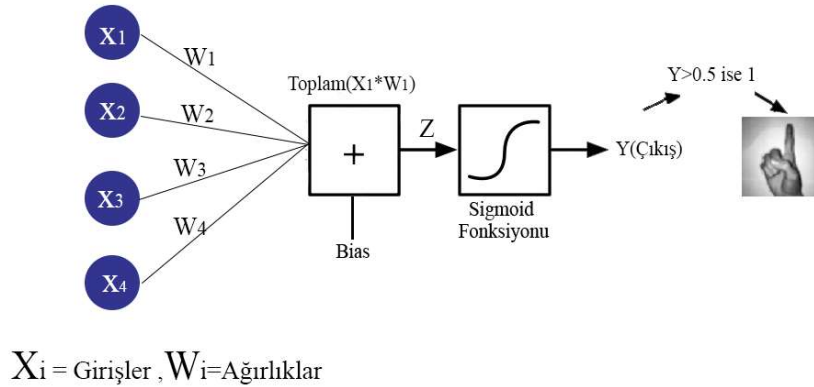
Şekil 3.4. Yapay sinir ağının genel yapısı

- **Giriş Katmanı:** Eğitilmesi istenen verilerin dışardan giriş olarak verildiği katmandır. Giriş katmanı katman sayılarında yer almaz. Örneğin Şekil 3.4’te görüldüğü gibi 1. Gizli, 2. Gizli ve Çıkış katmanı olarak 3 katmanlı olarak isimlendirilir.
- **Ara – Gizli Katman:** Bu düzeyde, eğitim işlemlerinin olduğu kamandır. Modelin yapısına bağlı olarak sinir ağlarında birden fazla ara katman olabilir veya hiç olmayabilir. Birden fazla gizli (orta) katman içeren ağlarda, katman sayısı da farklı olabilir. Bu durum karmaşıklığı artırır ve daha karmaşık görünen sorunları çözmek için kullanılabilir.
- **Çıkış Katmanı:** Ara katmanlarda elde edilen verilerin birleşmesi ile çıktıları üretildiği katmandır [55].

YSA modeli tek katmandan oluşuyorsa buna tek katmanlı sinir ağı, çok katmandan oluşuyor ise de çok katmanlı sinir ağı denilmektedir.

3.1.1. Tek Katmanlı Algılayıcılar

En basit YSA modelidir. Girdiler direk çıkışa ya da diğer katmana eklenebilir. İki girdili bir çıktılı yapıdadır. Girişler ağırlıklarla çarpılır ve bir aktivasyon fonksiyondan geçirilir ve ardından sınıflandırma işlemlerinde kullanılır. İki sınıfı ayırmak için tek katmanlı sinir ağı yeterlidir. Ayrıca birden fazla sınıf bulunuyorsa birden fazla tek katmanlı sinir ağına birleşmesi sonuca çözüme ulaşılabilir. Lojistik regresyonlar ve algılayıcılar [56] tek katmanlı sinir yapılarına örnektir. İkili sınıflandırma (0 ve 1 çıktıları) işlemine öncülük eden yapı lojistik regresyondur. Lojistik regresyon aslında en küçük YSA modelidir. Lojistik regresyonu anlamak YSA ve derin öğrenmeyi anlamaya yardımcı olacağından biraz açmakta fayda olacağını düşünülmektedir. Şekil 3.5’te lojistik regresyon yapısı verilmiştir.



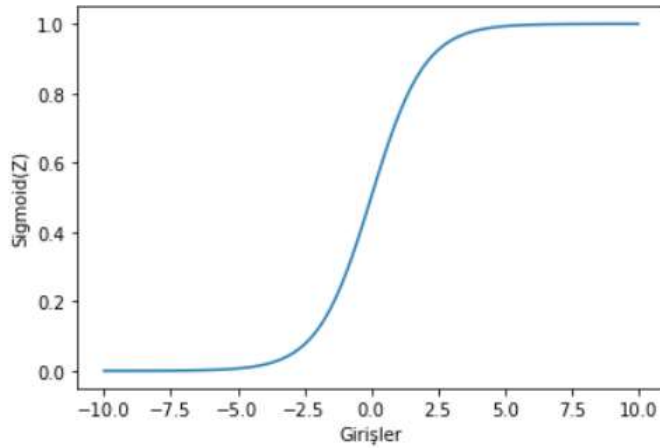
Şekil 3.5. Lojistik regresyon yapısı

Şekil 3.5’te görüldüğü gibi X değerleri, piksel değerini ifade etmektedir. W, ağırlıkları ifade edilmektedir. Ağırlıklar eğitilecek alanları simgelemektedir. Bias, nöronun tepki verme eşini belirleyen değerdir. Ayrıca bias aktivasyon fonksiyonunu sağa veya sola ötelenmesini (shift) sağlamaktadır. Z ise çıkıştır. Çıkışın denklemi aşağıdaki gibidir.

$$Z = X * W + b(bias) \quad (3.2)$$

Elde edilen çıkış değeri bir aktivasyon fonksiyonu ile işleme girmektedir. Şekil 13’te kullanılan fonksiyon sigmoid fonksiyonudur. Bu fonksiyonunun kullanım sebebi çıkışı 0 ve 1 arasına sıkıştırmasıdır. Sigmoid fonksiyonun grafiği Şekil 3.6’da verilmiştir. Böylece hiçbir zaman çok büyük bir değerle işlem yapmaya gerek kalmamaktadır. Sigmoid fonksiyonu denklem (3.3) de verildiği gibidir.

$$sigmoid = \frac{z1}{1+e^{-Z(çıkış)}} \quad (3.3)$$



Şekil 3.6. Sigmoid fonksiyonun grafiği

Loss fonksiyonu ileri yayılım sırasında gerçek ile tahmin arasındaki oluşan hata değeridir. Çıkış değeri y tahmin sonucu da Y olarak simgelenirse loss formülünün denklemi (3.4) aşağıdaki gibidir.

$$Loss(Kayıp) = -(1 - y)log(1 - Y) + y * log(Y) \quad (3.4)$$

Ayrıca önemli olan bir diğer fonksiyon da toplam kayıp (loss) değerleri olan maliyet(cost) fonksiyonudur. Maliyet fonksiyonu hata maliyeti anlamına gelmektedir. Kayıp fonksiyonlarının toplamını oluşturur. Maliyet değeri yüksek ise sonuç başarısız anlamına gelmektedir. Amaçlanan durum ağırlık ve bias değerlerini azaltarak maliyet değerini azaltmaktır. Maliyet formülünün denklemi (3.5) aşağıdaki gibidir.

$$Cost (Maliyet) = \sum lost \quad (3.5)$$

Loss değerlerini düzeltmek adına ileri yayılımın tersi yönünde fonksiyonların türevleri alınarak ağırlık ve bias değeri düzenlenir. Düzeltme işlemin her adımına geri yayılım algoritması denilmektedir. Bu işlemlerin tekrarına iterasyon (döngü) ve ağırlığın değişme miktarına da öğrenme katsayısı denilmektedir. Ağırlığın geri yayılımla değişmesi için ağırlığa uygulanacak değişim formülü denklem 3.6'daki gibidir. α , öğrenme katsayısıdır. Öğrenme katsayısının yüksek seçilmesi yanlış sonuçlar elde edebilir. Çok küçük seçilmesi ise fazladan iterasyon yani döngü yapabilmektedir [57].

$$W(yeni) := W - \alpha \frac{dj(W,b)}{d(w,b)} \quad (3.6)$$

3.1.2. Çok Katmanlı Algılayıcılar

Çok katmanlı algılayıcılar sınıflandırma ve bağlamlarda kullanılan bir yapay sinir ağı modelidir. Kullanım amacı insansı çözümler üretmektir. Algılayıcılar ve lojistik regresyonlar doğrusal olmayan problemlerde etkili sonuçlar üretememektedirler. Gerçek hayatta sorunların birçoğu lineer olmayan sorunlardır ve bunların çözümü çok katmanlı yapay sinir ağları ile çözülebilmektedir. Tek katmanlı yapay sinir ağlarından farklı olarak birden fazla gizli katman bulunmaktadır. Çok katmanlı yapay sinir ağı modeli Şekil 3.4'te verilmiştir. Derin ileri beslemeli ağlar ya da ileri beslemeli sinir ağları olarak da bilinen çok katmanlı algılayıcılarda, giriş ve çıkış katmanları arasına gizli katman olarak adlandırılan yeni bir katman eklenmiştir [50].

3.1.3. Eğitici (Gözetimli Supervised) ve Eğitici (Gözetimsiz Unsupervised) Öğrenme

Eğitici öğrenme, etiketli verilere öğrenmeyi sağlayan bir öğrenme modelidir. Modelde, giriş ve çıkış değişkeninin olduğu ve girdiden çıktıya eşleştirme kullanılarak öğrenmeyi sağlayan bir modeldir. Sistem kendine geri dönen hataya göre eğitilmiş olmaktadır.

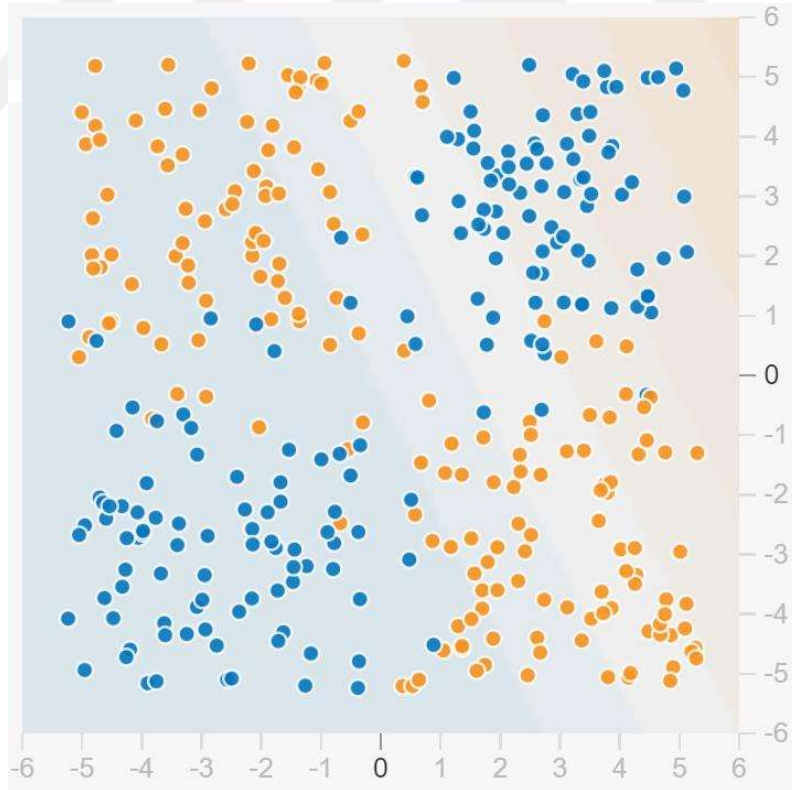
Eğitici öğrenme, etiketli veriler olmadan sezgisel olarak öğrenmeyi sağlayan bir öğrenme modelidir. Veri kümesi verildiğinde özellikleri otomatik olarak sınıflandırma yapmayı sağlar. Eğitici öğrenme, düzensiz verileri düzenli yapıya dönüştürür [58].

3.1.4. İleriye ve Geriye Yayılım

Öğrenmenin sağlanması için kayıp ve maliyet fonksiyonlarının en düşük seviyeye gelmesi gerekmektedir. Bundan dolayı her düğüm için ağırlık ve giriş değerlerinin çarpımına bias eklenip daha sonrasında aktivasyon fonksiyonuna tabi tutuluyor. Varsayılan ağırlık değerleri ve bias değerlerine göre elde edilen çıkış tahmini etiket verisi ile farklı gelmesi durumunda ağırlıkların ve bias değerlerinin değiştirilmesi için geri yönde türev yardımıyla değerler elde edilir. Elde edilen bu

değerler tekrardan YSA tabi tutularak yeni çıkış tahmini elde edilir yine bu tahmin ile etiket arasındaki fark, karekök farkları, mutlak değer farkları gibi matematiksel işlemler ile etiket ve tahmin sonucuna tekrardan bakılır. Loss değerinin düşmesi ve sabite yaklaşması durumunda öğrenme işlemine ulaşılmış olunur. İleri besleme, katmanlar ileri yönde ilerler etiket ve tahmin değerlerini kontrol edinceye kadar yapılan adımdır. Geri beslemeli kavramı ise türevlenerek yeni bias ve ağırlık değerlerini bulmayı sağlamaktadır. Modelin öğrenme yapısı bu şekilde sağladığı için ileri ve geri yayılım bu noktada çok önemlidir.

İterasyon, katman sayısı, öğrenme katsayısı, aktivasyon fonksiyonu gibi parametrelere hiper parametre olarak adlandırılır. Hiper parametreler, yapay sinir ağ modelinin doğru ve performanslı sonuçlar üretmesi adına önemlidir. Bir sınıflandırma uygulaması üzerinde çıktıları incelendiğinde daha iyi anlaşılacağı düşünülmektedir [59]. Yapay sinir ağı uygulanacak veriler ve yapıları Şekil 3.7’de verilmiştir. Şekil 3.7’de verilen renkler, veriyi ve ağırlıkları ifade etmektedir. Python dili ve tensorflow kütüphanesi kullanılarak verilerin sınıflandırma işlemi yapılmaktadır. Kullanılan hiper parametre değerleri ve başarımları Tablo 3.1’de verilmiştir. Çalışmada iterasyon ve öğrenme katsayısı sabit tutularak hiper parametreler değiştirilmiştir.



Şekil 3.7. YSA uygulanacak veriler ve yapıları (Renkler, veriyi ve ağırlıkları ifade etmektedir).

Yapılan çalışmada, aktivasyon fonksiyonu, öğrenme katsayısı, katman sayısı, düğüm sayısı ve eğitim oranı modelin öğrenmesinde önemli etkiler gösterdiği gözlemlenmiştir.

Tablo 3.1. YSA Hiper parametrelerin, değışime göre performansı (iterasyon 300)

Ara- Gizli Katman Sayısı	Düğüm – Node Sayısı	Aktivasyon Fonk	Test-Eğitim Oranı(%)	Test Loss sonucu	Eğitim Loss sonucu	Sınıflandırma Başarısı
1	2	ReLU	70			Başarısız
1	4	ReLU	70			Başarısız
2	4 - 2	ReLU	70			Başarısız
2	4 - 2	Sigmoid	70	0.017	0,011	Başarılı
2	4 - 2	Tanh	70	0,026	0,005	Başarılı
2	4 - 3	ReLU	70			Başarısız
2	4 - 3	Tanh	70	0.026	0.004	Başarılı
2	4 - 3	Sigmoid	70	0.056	0.006	Başarılı
2	4 - 4	Sigmoid	70	0.032	0.003	Başarılı

3.2. Derin Öğrenme Model Yapısı

YSA'nın katmanlarındaki düğüm ve gizli katmanların artmasıyla derinleşen yapay sinir ağları derin öğrenmenin kapılarını açmıştır. Derin öğrenme makine öğrenmesinin alt sınıfını oluşturmaktadır. Derin öğrenme, nesne tanıma, nesne sınıflandırma, mail spam işlemlerinde, kimlik tespitinde, tavsiye ve öneri sunma, dolandırıcılık işlemlerinde ve otonom araçlarında büyük veri analizinde, konuşma tanımlama, veri sınıflandırma gibi çeşitli alanlarda kullanılmaktadır [50-54].

Makine öğrenmesinde öznitelik çıkarmak ve seçmek zorlu bir işlemdir. Model eğitimlerinde bazı özelliklerin önemsiz olmasından dolayı modelden çıkarma işlemi yapılmaktadır. Derin öğrenme, özneteliği bulmayı kendisi sağladığından dolayı öznitelik seçme ve bulma ortadan kalkmıştır.

Model tasarlarırken eğitim ve test verilerine ayrılarak model gelişiminde bazı problemler yaşanmaktadır. Bu sorunları gidermek için hiper parametre seçimi önemlidir. Hiper parametre, veriden öğrenen makine öğrenmesi ya da derin öğrenme modelleri tasarlarırken ne olması gerektiği çeşitli koşullar göz önüne alınarak modeli tasarlayan kişiye bırakılmıştır. Hiper parametreler, probleme, veri setine ve donanıma bağlı olarak değışkenlik gösteren parametrelerdir [60].

3.2.1. Model Gelişiminde Karşılaşılan Problemler

Model gelişiminde bazı problemler bulunmaktadır. Bunlar;

- Dengesiz veri kümesi problemi veri setinde birçok veri olabilir ama büyük kısım aynı tipte olması durumunda karşılaşılan problemidir.

- Test ve eğitim veri seti yapısından kaynaklanan problemler.
- Doğruluk değerlendirmesi yapılması
- Bias ve Varyans ilişkisinin iyi kurulması
- Aşırı öğrenme (ezberleme)

3.2.2. Model Gelişiminde İyileştirme Fikirleri

Model gelişiminde modelin daha iyi performans sağlaması için bazı iyileştirme fikirleri bulunmaktadır. Bunlar;

- Model tasarlanırken olabildiğince fazla veri gerekmektedir.
- Model fazla olmasının yanında çeşitliğe sahip olması gerekmektedir.
- Eğitim süresini uzun tutmak gerekebilir. Ayrıca farklı en uygun şekle sokma yöntemleri de kullanılmalıdır.
- Katman veya node eklenmesi yapılabilir.
- Unutma yani seyretme işlemleri uygulanabilir.
- Model mimarisi, gizli katman sayısı, aktivasyon fonksiyonu farklı seçilebilir.

3.2.3. Regularizasyon

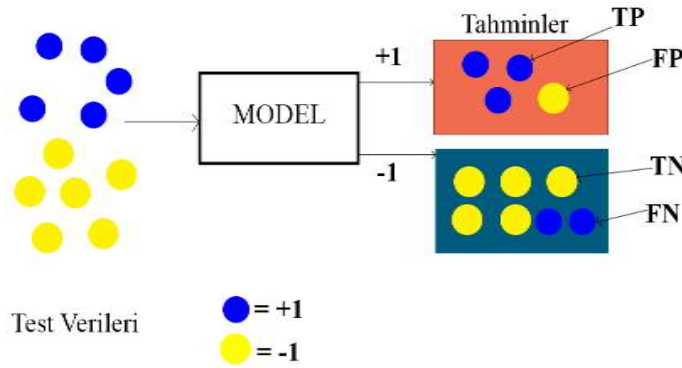
Modelden bağımsız olarak başarıyı artırmanın bazı yolları bulunmaktadır. Buna düzeltme (Regularization) denilmektedir. Düzenli hale getirme, ezberlemeyi önlemede ve performans artırma açısından önemli bir bileşendir. Bu yöntemler modeli uygun değerlerle çalışmasını sağlar. Düzeltme yollarından bazıları;

- Veri artırma (Data Augmentation): Veri eklenmesi ile model daha iyi eğitilir. Ancak verinin yön, parlaklık vb. özelliklerinin değiştirilmesi gibi yöntemlerde uygulanabilir.
- Sentetik veri üretme (Synthetic Data Generation): Sentetik olarak yüz veya olmayan bir veri üretme yapılarak veri artırımı sağlanabilmektedir. Derin öğrenmenin yeni teknolojileri ile başarılmaktadır.
- Gürültü ekleme (Adding Noise): Görüntüye gürültü ekleme işlemi yapılarak veri artırımı sağlanabilmektedir.
- Seyretme (Dropout): Düğüm ve katman azaltarak insan yapısına benzeyen unutma yapısıdır.
- Seyrek – Yoğun –Seyrek Eğitim (Dense-Sparse-Dense Training): Öğrenme yöntemidir. İnsanların özellikle bebeklerin öğrenme modeline benzer yapıda çalışan bir yöntemdir.

- Erken Durdurma (Early Stopping): Eğitim ve test kayıp değerleri arasındaki fark artmaya başladığında ezberleme durması adına yapılan işlemdir. Bu işleme erken durdurma denilmektedir.
- Transfer öğrenme (Transfer Learning): Daha önce eğitilmiş verileri modelde kullanmak için kullanılan bir yöntemdir [61].

3.2.4. Optimizasyon

Problemi incelemek ya da çözmek işlemlerine, en uygun şekle sokma yani optimizasyon olarak ifade edilir. Model değerlendirme önemli bir yere sahiptir. Model değerlendirme işleminde kullanılan bazı metrikler bulunmaktadır. Bu metrikler özellikle insan hayatını doğrudan etkileyen problemlerde önemlidir. Hassasiyet (Precision), Kesinlik (Recall) ve Doğruluk (Accuracy) metrikleri bunlardan bazılarıdır. Şekil 3.8’de görüldüğü gibi bir model olduğunu düşünülürse +1’ler mavi -1’ler sarı renkle gösterilen yapılar verilerdir. Bu veriler bir modelden geçirilip -1’ler bir sınıfa diğerlerini de ikinci sınıfa atan bir yapıdır.



Şekil 3.8. Model değerlendirme metrikleri

TP gerçek pozitif değerini, TN gerçek negatif değerini, FP beklenmedik sonuç değerleri ve FN ise kaybedilen sonuç değerlerini gösterildiği varsayılırsa oluşturulan modelin doğruluk değerleri aşağıdaki denklemlerle ifade edilir.

$$Hassasiyet = \frac{TP}{TP+FP} \quad (3.7)$$

$$Kesinlik = \frac{TP}{TP+FN} \quad (3.8)$$

$$Doğruluk = \frac{TP+TN}{TP+FP+TN+FN} \quad (3.9)$$

Bir başka ölçüt, çalışma süresi ve doğruluk değeri ilişkisidir. Modelin çalışma süresi önemli ise doğruluk ve çalışma süresi seçeneklerinden çalışma süresini temel alarak seçim yapılması gerekmektedir.

Bir başka ölçüt, tüm verilerin bir arada verilmesidir. Bu modelin işlem yükü açısından oldukça karmaşıktır. Bundan dolayı veriyi küçük paketlere ayırıp model geliştirme sürecine katkıda bulunur.

Bir başka ölçüt, aktivasyon fonksiyonunun seçimidir. Tablo 3.2’de aktivasyon fonksiyonların davranışı görünmektedir. En çok kullanılan aktivasyon fonksiyonu sigmoid, RELu, Swish, ELU, SoftPlus, Tanh fonksiyonlarıdır [50].

Bir başka ölçüt, eğitim tur sayısının (Epoch) seçimidir. Model eğitilirken verilerin tümü eğitime dahil edilmez. Parçalara bölünerek eğitilir. Öncelikle ilk parça eğitilir ve modelin performansı kontrol edilir, performans değerine göre geriye yayılım ile ağırlıklar değiştirilir. Bu işlem tekrarlanarak model için en iyi ağırlık değerleri hesaplanır. Bu adımların hepsine eğitim tur sayısı denilmektedir.

Bir başka ölçüt, başlangıç ağırlıklarının seçimidir. Ağırlık eğitilen yani öğrenen değerdir. Model çalışmasında varsayılan değer olarak ya sıfır ya da rasgele küçük değerler seçilir.

Bir başka ölçüt, bias seçimi ve varyans ilişkisidir bu ölçüm performansı etkileyen önemli bir ölçüttür [50,55,62].

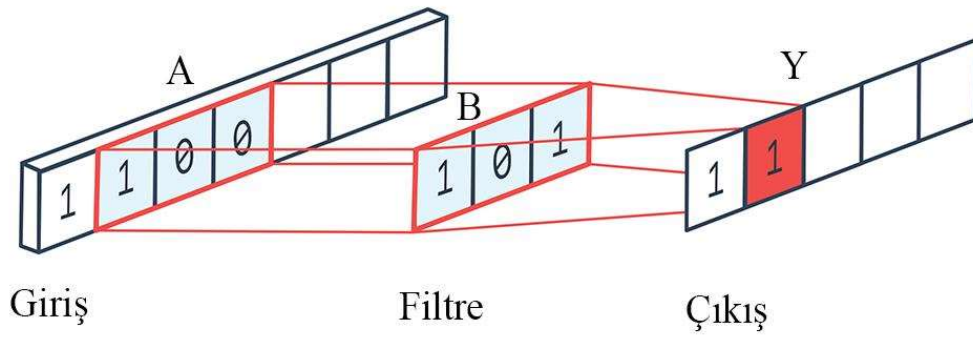
3.3. Evrişimli Sinir Ağları (Convolutional Neural Networks)

Evrişimli sinir ağları (CNN) bilgisayarlı görünün birçok alanında tercih edilen bir yöntem haline gelmiştir. CNN genel olarak sınıflandırma uygulamalarında kullanılmaktadır. Evrişimli sinir ağları yapısı gereği giriş olarak görüntü veya videoda herhangi bir andaki görüntüleri alır. Daha sonra görüntüye kenar bulma, piksel ekleme (padding), adım kaydırma (stride) ve havuzlama (pooling) gibi evrişim işlemlerin uygulayıp elde edilen görüntüye YSA uygulanma işlemi evrişimli sinir ağları olarak adlandırılır. CNN hareket tespitinde, nesne tespitinde, kişi takip edilmesinde ve sınıflandırma işlemlerinde sıklıkla kullanılır. CNN kameradan alınan verileri işleyip, veriler hakkında bilgi edinmemizi sağlamaktadır.

CNN’ler 80’lerin sonlarında ortaya çıkmış bir teknoloji, ancak derin CNN’lerin düzgün bir şekilde eğitmek için ayrılma, aktivasyon fonksiyonları, toplu normalleştirme ve veri artırma gibi ayrıntıları bulmak onlarca yıl almıştır. Son zamanlarda CNN, uçtan uca eğitilebilir ve güçlü özellik çıkarma yeteneği olduğu için bilgisayar görüşünün birçok alanına hakim olmayı başarmıştır. Gelişmiş paralel hesaplama ekipmanları ile CNN’ler gerçek zamanlı olarak çıkarılabilir hale gelmesi CNN olan talebi artırmıştır.

3.3.1. Evrişimli Sinir Ağı Katmanları

Evrişim (Konvolüsyon) Katmanı: Evrişim, öznitelik haritası oluşturma olarak tanımlanabilir. CNN'nin evrişim katmanı CNN'nin ana yapı taşıdır. Görüntünün özelliklerini algılamaktan sorumludur. Evrişim, hesaplamayla ilgili ağır yüklerin çoğunu yapan bir evrişimli ağı temel yapı taşıdır. Evrişim işlemi aslında işaret işleme gibi mühendislik derslerinde işlenen bir konudur. Görsel işlendiği için evrişim iki boyutlu matris olarak tanımlanmaktadır. Evrişim işleminin temel amacı evrişim matrisini bazı filtreler ile matrisi tarayarak özellikleri ortaya çıkarmaktır. Şekil 3.9'da evrişim işlem adımları verilmektedir.



Şekil 3.9. Evrişim katmanı işlemleri

Şekil 3.9'da gösterilen A matrisi giriş matrisi, B ise bir kenar bulma filtresi ve Y çıkışı değeridir. B matrisini simetriği alınarak A matrisi üzerinde hareket ettirerek yani kaydırarak Y değeri elde edilmektedir. Kaydırma işlemine de adım kaydırma (Stride) denilmektedir. Matematiksel olarak konvolüsyon işlemi aşağıdaki denklemde tanımlanmıştır.

$$Y(i, j) = \sum_j \sum_i A[i, j] \cdot B[0 - i, 0 - j] \quad (3.10)$$

Görüntüdeki özellikleri çıkarmak için Şekil 3.9'da gösterildiği gibi görüntüye bazı filtreler uygular kenar bulma filtresi (B) uygulanmıştır. Giriş resmi işlemine uygulanan sonucunda elde edilen yeni resme özellik çıkarma işlemi gerçekleştirilmiş olur. Birden fazla filtre uygulanarak farklı özellikleri çıkarma işlemi yapılmaktadır. Uygulanan filtre ile giriş görüntüsüne kaydırıp işlem yapılmasına konvölüsyon denilmektedir. Konvölüsyon uygulanması sonucu elde edilen çıkış küçülmüş görüntüye dönüşür. Ayrıca aktivasyon (RELu) uygulanarak bias eklenir ve böylece lineer olmayan bir yapıya dönüştürülmüş olur. Filtre boyutundan görüntülerde küçülme meydana gelmektedir. Görüntü boyutunu eski boyutuna getirmek için piksel ekleme işlemi yapılır. Bu işleme piksel ekleme veya doldurma (Padding) denilmektedir. Piksel ekleme de boş alanlara sıfır ile doldurma veya görüntüye uygun piksellerle ekleme yapılabilir. Aşağıdaki denklemde filtre

uygulandıktan sonra elde edilen matris boyutunu veren formüldür yani filtre uygulanan görüntünün yeni boyutunu vermektedir. Çok kanallı görüntülere tensör denilmektedir.

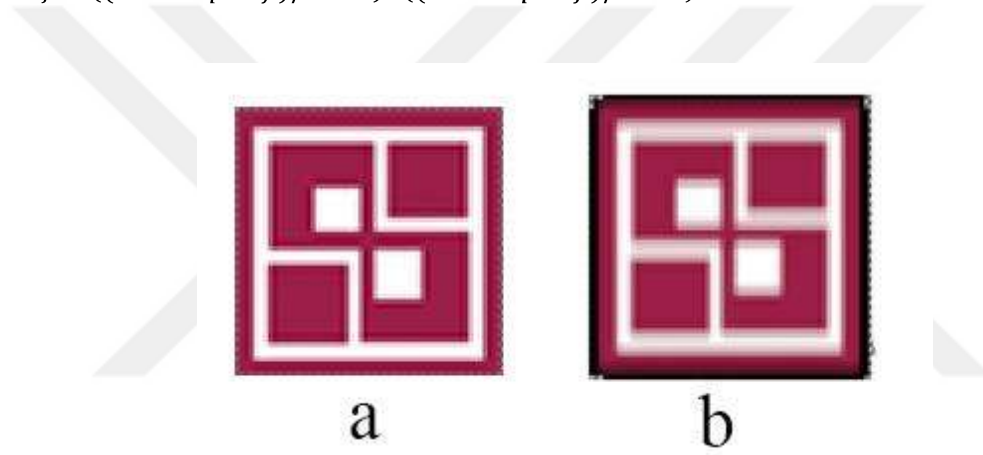
$$giriş = (n \times n), filtre = (f \times f), çıkış = (n - f + 1) \times (n - f + 1) \quad (3.11)$$

Aşağıdaki denklemde piksel ekleme varsa çıkışın boyutunu veren formüldür (p = padding).

$$g = (n \times n), f = (f \times f), p = \frac{(f-1)}{2}, \zeta = (n + 2 * p - f + 1) \times (n + 2 * p - f + 1) \quad (3.12)$$

Aşağıdaki denklemde ise birden fazla filtre varsa elde edilen çıkış görüntüsünün denklemini vermektedir (s = adım sayısı).

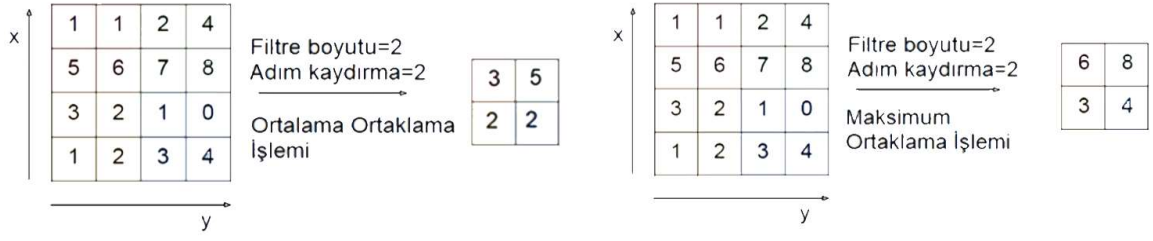
$$\zeta_{ıkış} = ((n + 2 * p - f)/2 + 1) \times ((n + 2 * p - f)/2 + 1) \quad (3.13)$$



Şekil 3.10. Konvolüsyon uygulanan görüntü

Görüntü değiştirme editörü olan GIMP ile kenar bulma filtresi uygulanarak konvölüsyon işlemi yapılmış görsel Şekil 3.10'da verilmiştir. Şekil 3.10'da verildiği gibi sol (a) görsel kenar bulma uygulanmamış görsel sağ (b) görsel ise kenar bulma uygulanmış çıktıdır.

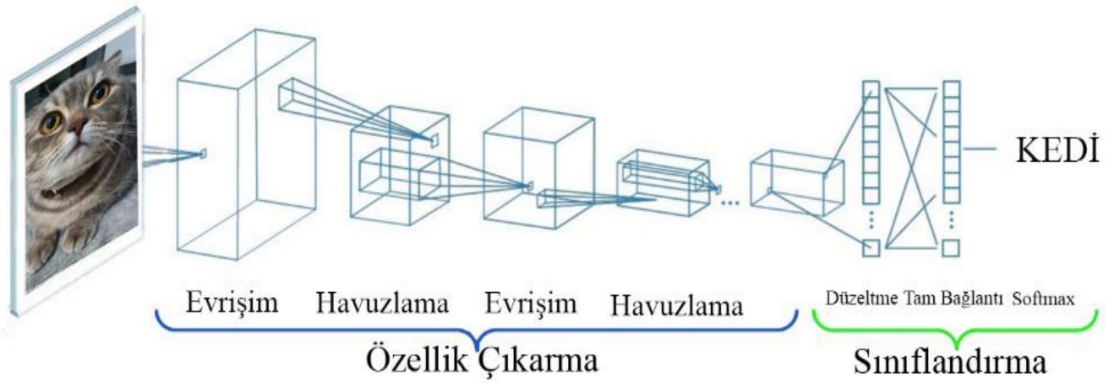
Pooling (Havuzlama-Ortaklama) Katmanı: Gösterimin boyutunu, ağ parametrelerini ve hesaplama sayısını azaltmak için kullanılan bir katmandır. Sinir ağının sonuç vermesi için bilgileri barındıran ve küçük çıktı üreten katmandır. En çok kullanılan iki yöntemden biri Ortalama havuzlama diğeri ise en büyük havuzlamadır (max pooling). Ortalama havuzlama işleminde 2x2 bir filtre uygulanmış görüntüye filtre işleminden sonra adım kaydırma 2 olarak seçilerek değerlerin ortalaması alınarak yeni elde edilen küçük resmin değerlerini oluşturur. Havuzlama uygulama yapısı Şekil 3.11'de verilmiştir.



Şekil 3.11. Ortalama ve en büyük havuzlama yapısı

Evrişim ve Pooling katmanından gelen matrislerin tek boyutlu diziye çevrilmiş işlemine Flattening (Düzleme) denir. YSA'nın giriş değerleridir.

Full Connection (Tam bağlantı) Katmanı: Flatting (düzeltme) verilerini giriş olarak YSA işlemine girdi olarak verilmekte ve bu girişlere YSA uygulanmasına tam bağlantı denilmektedir. CNN katmanlarının tamamını içeren örnek bir model Şekil 3.12'de verilmiştir [60].



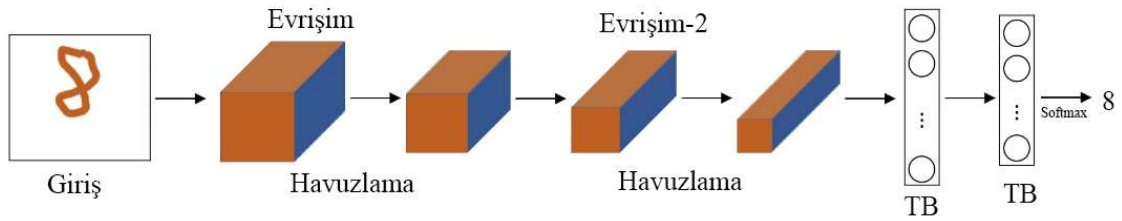
Şekil 3.12. CNN katmanlarının tamamını içeren örnek bir model [63]

3.3.2. Evrişimli Sinir Ağı Mimarisi

Modeller, Evrişimli sinir ağının performanslı açısından son derece önemlidir. LeNet-5 [64] ve AlexNet'in [65] başarılı sonuçlar elde etmesi ve ImageNet [66] yarışmasındaki başarıları ile diğer modellere göre avantajlı olduğunu göstermişlerdir. Bundan dolayı model geliştirmeye yönelim artmış ve birçok yeni derin evrişimli sinir ağı (CNN) modeli geliştirilmiştir. Yönelme ile CNN'ler, son yıllarda bilgisayarlı görü alanını ve özellikle üst düzey görme problemlerini önemli ölçüde ilerletmiştir. Çalışmada, kronolojik olarak günümüze gelmiş CNN mimariler yer akmaktadır.

LeNet-5:

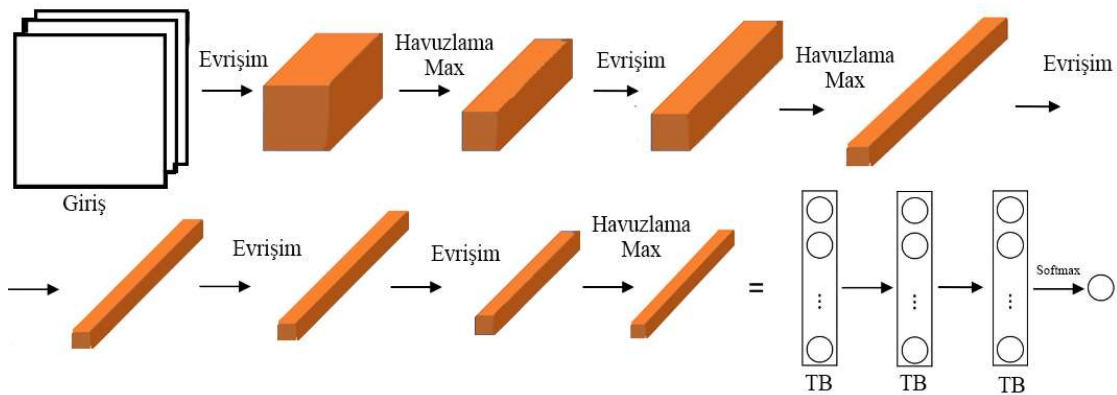
LeCun ve arkadaşları [64], el yazılarını kameradan okumak için LeNet-5 adında bir CNN modeli tasarlanmıştır. 1998 yılında yayınlanan ve başarılı sonuçlar veren ilk CNN modelidir. MNIST verileri üzerinde çalışmalar yapmışlardır. Bu modelde havuzlama için ortalama havuzlama yöntemi kullanılmış ve aktivasyon fonksiyonu (hiperbolik Tan ile sigmoid) kullanılmaktadır. LeNet-5 yedi katmanlı bir mimaridir. LeNet-5 mimari yapısı Şekil 3.13'te verilmiştir. Katmanlar, 3 evrişimli katmandan, 2 alt örnekleme katmanından ve 2 tam bağlantı katmanından oluşmaktadır.



Şekil 3.13. LeNet-5 mimari yapısı [64]

AlexNet:

LeNet-5 ten sonra 2012 yılında CNN'nın tekrar popüler hale gelmesini sağlayan çalışmadır. Donanım ve internetin yaygınlaşması kullanılabilecek verilerin sebep olmuştur ve CNN'nın gelişmesini sağlamıştır. Krizhevsky ve arkadaşları [65], tarafından geliştirilmiştir. Birbirini takip eden evrişim ve havuzlama katmanları bulunmasından dolayı yapı olarak LeNet-5 modeline benzemektedir. LeNet-5'ten farklı olarak paralel işlem yapılmıştır. Yine aktivasyon fonksiyonu (ReLU) kullanılmıştır. Havuzlama katmanlarında da en büyük havuzlama kullanılmaktadır. AlexNet mimari yapısı Şekil 3.14'te verilmiştir.



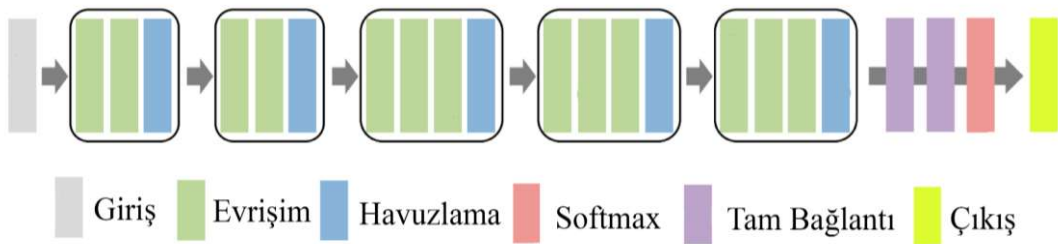
Şekil 3.14. AlexNet mimari yapısı [65]

GoogLeNet:

Google'daki araştırmacılar, 2014 yılında sınıflandırma ve tespit problemleri için 2014 ImageNet yarışmasında ilk sırada yer alan Inception ağını tanıtmışlardır. Model, içinde farklı yapıdaki birçok evrişim işlemi yapan ve sonuçları bir araya getiren temel bir birimden oluşmaktadır. Hesaplama performansı artırmak için, (1x1) evrişimler kullanmışlardır. Çıktının düzgün bir şekilde birleştirilebilmesi için boyutları korumak için maksimum havuzlama kullanılmıştır [67].

VGG-16:

VGG-16, 2015 yılında yayınlanan derin bir CNN ağıdır. Görsel Geometri Grubu (VGG) olarak tanımlanmaktadır. Evrişim katmanı ikili ve üçlü kullanılmaktadır. ImageNet veri seti üzerinde işlem yapılmıştır. Yine aktivasyon fonksiyonu (ReLU) kullanılmıştır. Havuzlama katmanlarında da en büyük havuzlama kullanılmaktadır. 3×3 boyutunda çok küçük evrişim filtrelerine sahip mimari kullanılır, bu da son teknoloji konfigürasyonlara kıyasla önemli gelişme gösterir [68]. VGG-16 mimari yapısı Şekil 3.15'te verilmiştir.



Şekil 3.15. VGG-16 mimari yapısı [68]

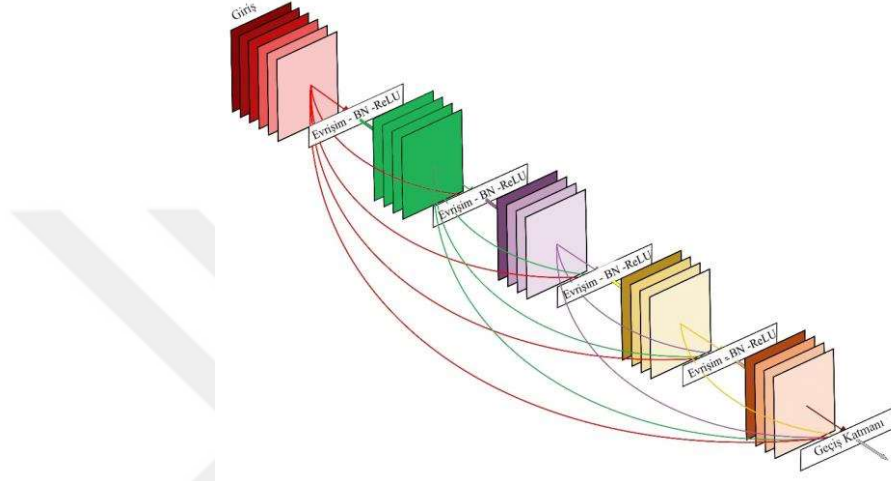
ResNet:

Bir katman atlayarak model eğitimi yapılan bir ağ mimaridir. He ve arkadaşları [69] tarafından, 2015 yılında geliştirilen bir modeldir. Ağ mimarisine ResNet adını vermişlerdir. Eğitim maliyetini azaltmak için birden çok artık blok dedikleri katmanları kullanmışlardır. ResNet, hesaplama açısından zorluk olan ve performansı etkileyen parametre hesaplama sayısını azaltmasından dolayı tercih edilen bir model olmuştur.

DenseNet:

Ağda daha önceki özellik haritalarına başvurmak faydalı olabilir düşüncesi ile Huang ve arkadaşları [70], 2016 yılında bu modeli geliştirmişlerdir. Modelde her katmanın özellik haritası, yoğun bir blok içindeki her ardışık katmanın girdisine birleştirilir. Bu, ağdaki daha sonraki katmanların önceki katmanlardan gelen özelliklerden doğrudan yararlanmasına izin vererek

özelliğin ağ içinde yeniden kullanımını teşvik eder. Gao H. Ve ark, farklı katmanlar tarafından öğrenilen özellik haritalarının birleştirilmesi, sonraki katmanların girdisindeki çeşitliliği artırır ve verimliliği artırır sonucuna ulaşmışlardır. Ağ önceki herhangi bir özellik haritasını doğrudan kullanabildiğinden, yazarlar çok küçük çıktı kanalı derinlikleriyle (yani katman başına 12 filtre) çalışabileceklerini ve gerekli toplam parametre sayısını büyük ölçüde azaltabileceklerini keşfettiler. ResNet modelleriyle karşılaştırıldığında, DenseNets'in daha az karmaşıklıkla daha iyi performans elde ettiği bildirilmektedir. Şekil 3.16'da DenseNet mimari yapısı verilmiştir.



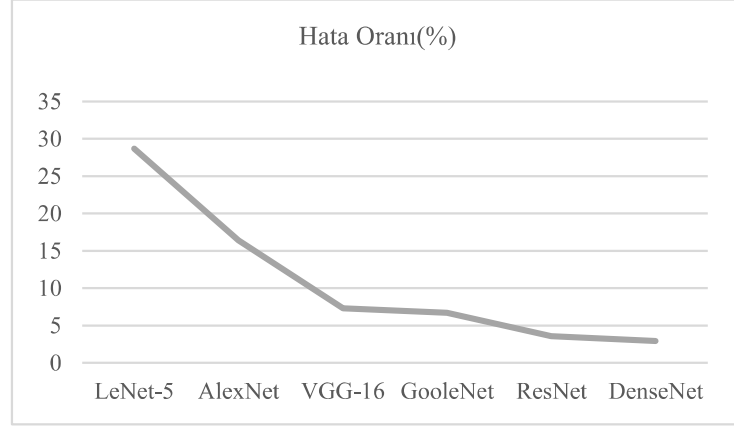
Şekil 3.16. DenseNet modeli mimarisi [70]

Mimarilerin ImageNet verilerine göre derinlikleri ve hata oranlarını gösteren değerler Tablo 3.2'de verilmiştir. Grafik verileri de Şekil 3.17'te verilmiştir.

Tablo 3.2. ImageNet verilerine göre derin öğrenme mimarilerin başarımları oranları [66]

Mimari Adı	Oluşturulma Tarihi	Katman Sayısı	Hata Oranı (%)
LeNet-5	1998	5	28,7
AlexNet	2012	8	16,4
VGG-16	2014	16	7,3
GooleNet	2014	22	6,7
ResNet	2015	56	3,57
DenseNet	2016	150	2,95

Tablo 3.2'de gösterildiği gibi katman sayısındaki artışa (derinlik) bağlı olarak hata oranlarında düşüş yaşandığı gözlemlenmiştir.

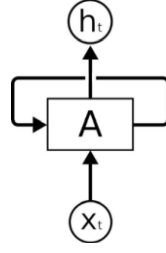


Şekil 3.17. Mimarilerin ImageNet verilerine göre derinlikleri ve hata oranları

Şekil 3.17’de de gösterildiği üzere katman sayısının artmasına bağlı olarak hata oranlarının düştüğü gözlemlenmiştir.

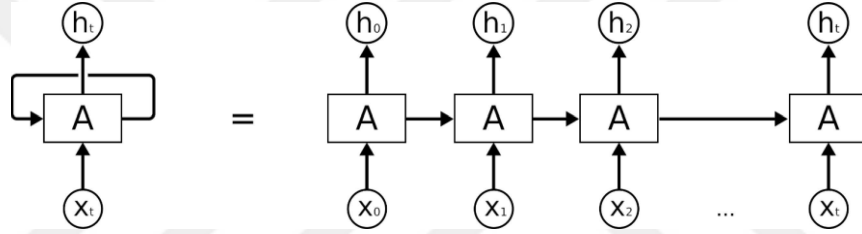
3.4. Özyinelemeli Sinir Ağları (Recurrent Neural Networks)

Özyinelemeli Sinir Ağları (RNN), 1980’li yıllardan bugüne üzerinde çalışmalar yapılan derin öğrenme konusudur. Zaman serisi şeklinde ifadeliden problemlere cevap olması adına üretilmiş bir çözümdür. Özellikle ses, video ve metin ile ilgili olan problemlerde kullanılan bir sinir ağıdır. Müzik üretme, sestten metne metinden sese dönüştürme işlemlerinde, medikal alanlarda, ilaç ve kimyasal kodlama ve üretimde, çeviri programlarda, doğal dil işlemede, video işleme, duygu analizi kullanım alanlarından bazılarıdır. Bu problemlerin çözümünde RNN kullanılır. Denetimli öğrenme modelleri ile öğrenme işlemi yapmaktadır. Özyinelemeli sinir ağların popüler olmasının temel nedeni, giriş olarak vektör alması ve çıkış olarak da dizi üretilip üzerinde işlem yapmamıza izin vermeleridir. Bilginin kalıcı olmasına izin veren, içlerinde döngüler bulunan ağlardır. Özyinelemeli sinir ağların girişler birbirinden bağımsızdır ve sıralamanın önemi yoktur ayrıca girişin çıktı verdikten sonra YSA girdiyle işlem yapmaz. Kelime veya video verilerini ağa bir seferde giriş olarak vermek gerekmektedir. Diğer sinir ağı türlerinde olduğu gibi verileri parçalara ayrılmaz çünkü veriler birbirinden bağımsız düşünülmemektedir [71]. Şekil 3.18’de verildiği gibi, diyagramda bir sinir ağı parçası olan A , bazı X_t girişine bakarak, bir h_t değeri elde etmektedir.



Şekil 3.18. Özyinelemeli ağ parça yapısı [71]

Döngü, bilgilerin ağın bir adımından diğerine aktarılmasına izin verir. Özyinelemeli sinir ağı her biri bir sonrakine bilgi ileten aynı ağın birden fazla kopyası olarak düşünülebilir. Özyinelemeli sinir ağının veri paylaşım yapısı Şekil 3.19’da verilmiştir. Aktivasyon fonksiyonu Tanh, ReLu ve Sigmoid fonksiyonları kullanılmaktadır.



Şekil 3.19. RNN mimari yapısı [71]

3.5. Kapsül Ağları (Capsule Networks)

2017 yılında evrişimli sinir ağlarının açıklarından dolayı Hinton ve arkadaşları [72], tarafından geliştirilen bir modeldir. Evrişimli sinir ağları görüntü işleme sorunlarını çözmeye önemli başarılarla imza atmıştır. Ancak evrişimli sinir ağlarında havuzlama işleminden dolayı veri kayıplarına neden olmaktadır. Bundan dolayı kapsül ağları önermişlerdir. Kapsül ağların çıkışları vektörel olarak hesaplanmaktadır. Çıkışlardan vektör elde edilmek için girdisinin ve çıktısının vektör olduğu ezme (squashing) aktivasyon fonksiyonu kullanılır. Kapsül ağlarında evrişim katmanının çıkışı kapsüle giriş olarak uygulanmaktadır. Havuzlama işleminden daha başarılı sonuçlar elde edilmiştir. Bir kapsül aktivite vektörü, bir nesne veya bir nesne parçası gibi belirli bir varlık türünün somutlaştırma parametrelerini temsil eden bir nöron grubudur. Bir seviyedeki aktif kapsüller, daha yüksek seviyeli kapsüllerin örnekleme parametreleri için dönüşüm matrisleri aracılığıyla tahminlerde bulunur. Birden fazla tahmin uyduğunda, daha yüksek seviyeli bir kapsül aktif hale gelir. Kapsül katmanlarının çıkışlarında, nesneyi temsil eden özelliğin olasılığı, konumu, yönelimi gibi bilgileri belirlenir [72].

4. OPTİK AKIŞ

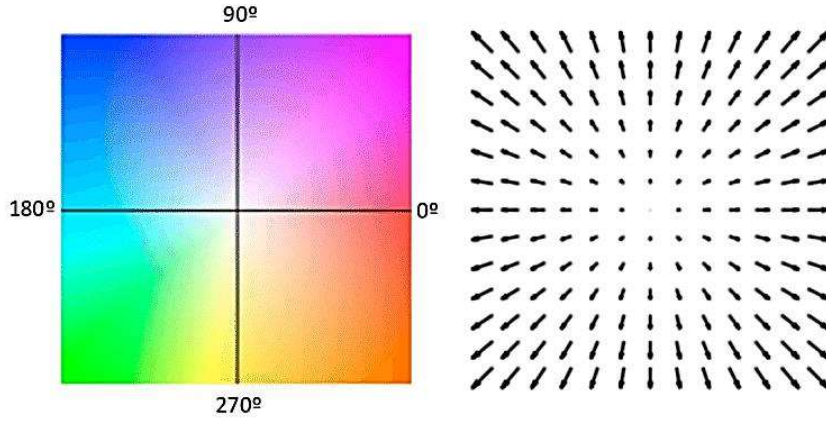
Belirli bir görüntü dizisinden hareketin tahmin edilmesi bilgisayarlı görüdeki temel sorunlardan biridir. Optik akış kavramı ilk olarak 1950'de James Gibson [73] tarafından önerilen bir yöntemdir. Optik akış, zaman ve görüntü düzlemindeki piksellerin hareketi olarak tanımlanır. Genellikle görüntü düzlemine yansıtılan gerçek hareketin iyi bir tahmini olarak tanımlanır. Optik akışı hesaplayan çoğu yöntem, bir pikselin renginin bir video karesinden diğerine küçük yer değiştirme altında değişmediğini varsayarak optik akış tahminini elde edilir. Optik akış hem hareket eden görüntünün bölgelerinin hem de hareket şiddetinin bir açıklamasını sağlar. Optik akış tahmini, hareket algılama [12], videonun anlaşılması [13] ve otonom araçlarda [1], robotik sistemler, medikal alanlar gibi birçok görme uygulamasında önemli bir rol oynayan bilgisayarla görmedeki temel bir konudur [18].

Optik akışın, art arda gelen görüntülerdeki aynı renge sahip piksellerin farklı karelerde aynı renge sahip olacağından nesnenin aynı olması ve hareketlerin küçük olma prensibi temelidir. Temel olarak optik akış görevi, art arda gelen iki görüntü arasındaki nesne yer değişiminin piksel tabanlı kaydırma vektörünün hesaplanmasını ifade eder. Optik akışın ana fikri, nesnenin hareketinden veya kamera hareketlerinden kaynaklanan yer değiştirme vektörünü tahmin etmektir [74].

Optik akış yoğun ve seyrek olmak üzere iki ana kategoriye ayrılır. Seyrek Optik akış, hız vektörlerini (hareket) izlemek için seyrek bir piksel seti (örneğin kenarlar ve köşeler gibi filtreler) seçmektedir. Çıkarılan özellikler, aynı noktaların izlenmesini sağlamak için optik akış işlevinde çerçeveden çerçeveye geçirilmektedir. Lucas Kanade (LK) yöntemi [9], Horn-Schunck (HS) yöntemi [8], Buxton Buxton [75] yöntemi ve daha fazlası olmak üzere optik akışın çeşitli algoritma ve uygulamaları bulunmaktadır. Yoğun optik akış ise her karenin her pikseli için optik akış vektörünü hesaplamaya çalışır. Farneback [7] metodu yoğun optik akış hesaplama yöntemlerindendir. Çalışmada optik akış denklemi ve klasik optik akış yöntemleri detayları ile incelenmektedir.

4.1. Optik Akış Yön ve Şiddet Haritası

Optik akışın yönü, renk ile şiddeti ise rengin tonuyla ifade edilmektedir. Optik akış renk ve şiddet haritası Şekil 4.2'de verilmiştir.

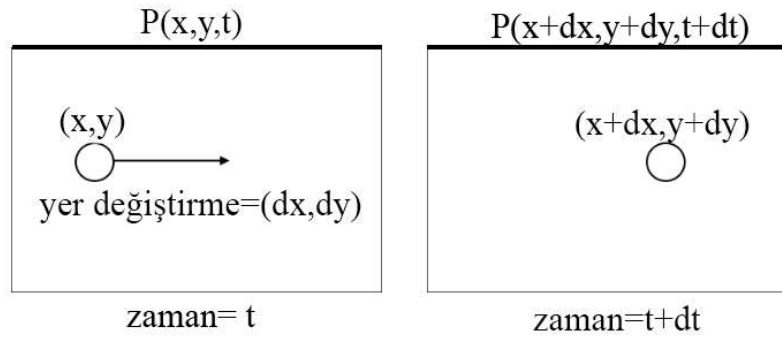


Şekil 4.1. Optik akış renk ve şiddet haritası [76]

Şekil 4.2’de verilen harita, optik akış hesaplamalarında elde edilen sonuçların değerlendirilmesinde kullanılan başarımların göstergesidir. Hareket sola doğru olduğunda hareketli nesnenin rengi Şekil 4.2’de verildiği gibi 180 derece yönündeki renklere dönüşür. Ayrıca hareketin hızı yüksek ise rengin daha koyu olması ile beklenir.

4.2. Optik Akış Denklemi

Optik akış denklemi, nesnenin parlaklığı önceki ve sonraki görüntüde sabit kaldığını ve iki görüntü arasında hareketin yavaş olduğu varsayımı ile oluşturulmaktadır. Optik akış denklemini oluşturmak için ardışık iki görüntü temel alındığında t zamanından bulunan (x,y) koordinattaki bir pikselin bir sonraki görüntüde yer değişimi yaptığını düşünüldüğünde aynı piksel $(x+dx, y+dy)$ konumuna gelmektedir. Bir sonraki görüntünün parlaklığının değişmeyeceği varsayımıyla altında denklem 4.1 elde edilir. Burada $P(x,y,t)$ parlaklık fonksiyonu (dx, dy, dt) ise küçük yer değişimi göstermektedir. $P(x+dx,y+dy,t+dt)$ ise $t+1$ zamanındaki görüntüdeki parlaklık fonksiyonudur. Optik akış hesaplama için örneklem modeli Şekil 4.1’de verilmiştir.



Şekil 4.2. Optik akış denklemi

$$P(x, y, t) = P(x + dx, y + dy, t + dt) \quad (4.1)$$

Türev, bir noktanın başka noktaya göre değişim miktarını vermektedir. Burada türev, değişimi ölçmek için kullanılır. Taylor serisi ise, fonksiyonun türevini hesaplamak kullanılır. Görüntüde belirli bir noktanın parlaklığı zamanla sabittir. Dolayısıyla aşağıdaki denklem elde edilir:

$$\frac{dP}{dt} = 0 \quad (4.2)$$

Parlaklığın uzay-zamansal düzlemde değişmediği kabulü ile değişim noktasındaki parlaklığın Taylor açılımı denklem 4. 3'te ifade edilmiştir.

$$P(x + dx, y + dy, t + dt) = P(x, y, t) + \frac{\partial P}{\partial x} dx + \frac{\partial P}{\partial y} dy + \frac{\partial P}{\partial t} dt + YDT \quad (4.3)$$

Denklem 4. 1'deki eşitliği denklem 4. 3 uygulanırsa denklem 4.4 elde edilir.

$$0 = \frac{\partial P}{\partial x} dx + \frac{\partial P}{\partial y} dy + \frac{\partial P}{\partial t} dt + YDT \quad (4.4)$$

Bununla birlikte, zaman değişimi çok küçük olduğu için YDT dikkate alınmaz ve denklem 4. 5 elde edilir.

$$0 = \frac{\partial P}{\partial x} dx + \frac{\partial P}{\partial y} dy + \frac{\partial P}{\partial t} dt \quad (4.5)$$

Elde edilen denkleme zamana göre zincir kuralı uygulanırsa denklem 4.6 elde edilir.

$$0 = \frac{\partial P}{\partial x} \frac{dx}{dt} + \frac{\partial P}{\partial y} \frac{dy}{dt} + \frac{\partial P}{\partial t} \frac{dt}{dt} \quad (4)$$

Hız denkleminde aşağıdaki eşitle dönüşüm sağlanırsa denklem 4.7 elde edilir.

$$V_x = \frac{dx}{dt}, V_y = \frac{dy}{dt}, 1 = \frac{dt}{dt} \quad (4.7)$$

Denkleme 4.7 denklem 4.6 ve 4.2'a uygulanırsa aşağıdaki denklem elde edilir.

$$0 = \frac{\partial P}{\partial x} V_x + \frac{\partial P}{\partial y} V_y + 0 \quad (4.8)$$

Denklem 4.8'in matris gösterimi aşağıdaki gibidir:

$$\frac{\partial P(x,y)}{\partial t} = - \left(\frac{dx}{dt} \frac{dy}{dt} \right) \begin{pmatrix} \frac{\partial P}{\partial x} \\ \frac{\partial P}{\partial y} \end{pmatrix} \quad (4.9)$$

Elde edilen denkleme iki boyutlu **hareket kısıtlama denklemi** denilmektedir [37]. Denklemde $u=dx/dt$, $v=dy/dt$ olarak ifade edilirse denklem aşağıdaki forma dönüşmektedir.

$$\frac{\partial P(x,y)}{\partial t} = -(u * v) \begin{pmatrix} \frac{\partial P}{\partial x} \\ \frac{\partial P}{\partial y} \end{pmatrix} \quad (4.10)$$

Veya

$$I_t = -(V_x V_y) \begin{pmatrix} I_x \\ I_y \end{pmatrix} \quad (4.11)$$

Denklem 4.11’da I_x ve I_y uzaysal şiddet gradyanı ve V_x ve V_y , (x, y) pikselindeki t anındaki görüntü hızı veya optik akışı denilmektedir. Gradyan vektörü tüm kısmi türevlerin bir araya konmuş halidir.

Hareket kısıtlama denkleminde elde edilen doğru, hızın tam yerini bulmaya yeterli değildir ve buna **belirsizlik problemi** denilmektedir. Hareket kısıtlama denklemi iki bilinmeyenli bir denklemdir. İki bilinmeyenli tek bir denklemin çözümü tek başına çözülmemektedir. Bundan dolayı bazı yaklaşımlar ile bunun çözümü aranmıştır. Bu yöntemler iki başlık altında toplanmıştır. Birincisi, seyrek optik akıştır. Seyrek optik akış, çerçeve içindeki bazı özelliklerin (örneğin bir nesnenin kenarlarını veya köşelerini gösteren birkaç piksel) akış vektörlerini verir. İkincisi ise optik akış yaklaşımı yoğun optik akıştır. Burada tüm görüntünün (tüm pikseller) akış vektörlerine bakılır. Yoğun optik akış, yavaş ve hesaplama açısından pahalı olmasına rağmen daha yüksek doğruluğa sahiptir. Seyrek optik akış ise hesaplama hızlı ama yoğun optik akış kadar hızlı değildir [76].

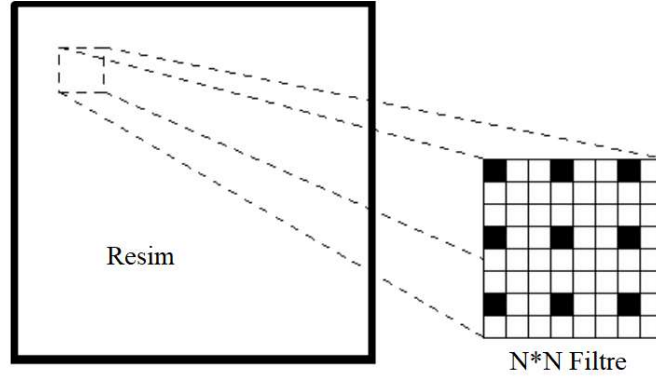
4.3. Seyrek Optik Akış

Seyrek optik akış, hız vektörlerini (hareket) izlemek için seyrek bir özellik kümesi (örneğin kenarlar ve köşeler gibi özellikler) seçerek optik akış hesaplaması elde etmektedir. Çıkarılan özellikler, aynı noktaların izlenmesini sağlamak için optik akış işlevinde çerçeveden çerçeveye aktarılır. LK yöntemi, HS yöntemi gibi çeşitli seyrek optik akış uygulamaları vardır. LK optik akış algoritması ve HS optik akış algoritması varsayımlar temelli çalışmaktadır.

4.3.1. Lucas-Kanade Optik Akış Yöntemi

Bilgisayarla görüde LK yöntemi, Bruce D. Lucas ve Takeo Kanade tarafından geliştirilmiş, optik akış tahmini için yaygın olarak kullanılan bir yöntemdir. LK yöntemi, birkaç yakın pikselden gelen bilgileri birleştirerek, optik akış hesaplaması yapmaktadır. LK yöntemi, seyrek bir özellik kümesi ile optik akışı hesaplamak için yaygın olarak kullanılır. Bu yöntemin ana fikri, yakındaki piksellerin aynı yer değiştirme yönüne sahip olduğu yerel bir hareket sabitliği varsayımına dayanmaktadır. Bu varsayım, iki değişkenli denklem için yani belirsizlik probleminin çözümünü elde etmeyi sağlar.

Yöntemde ilk olarak, Shi-Tomasi [77] tarafından tespit edilen özelliklerin etrafında küçük bir 3x3 filtre (neighborhood) uygulanır.



Şekil 4.1. Shi-Tomasi filtresi [77]

Shi-Tomasi önerdiği filtre Şekil 4.3'te verildiği gibi dokuz siyah noktanın hepsinin aynı hareket yönüne ve şiddetine sahip olduğunu varsayımı yapılmaktadır.

Bir denklem sistemi oluşturmak için sabit boyutlu bir pencere alınmaktadır. $P_i = (x_i, y_i)$, seçilen n elemanlı penceredeki piksel koordinatları olsun. Denklem sistemini aşağıdaki şekilde tanımlanır:

$$\begin{aligned} \frac{\partial P}{\partial x}(p_1) \frac{dx}{dt} + \frac{\partial P}{\partial y}(p_1) \frac{dy}{dt} &= -\frac{\partial P}{\partial t}(p_1) \\ \frac{\partial P}{\partial x}(p_2) \frac{dx}{dt} + \frac{\partial P}{\partial y}(p_2) \frac{dy}{dt} &= -\frac{\partial P}{\partial t}(p_2) \\ &\dots \\ \frac{\partial P}{\partial x}(p_n) \frac{dx}{dt} + \frac{\partial P}{\partial y}(p_n) \frac{dy}{dt} &= -\frac{\partial P}{\partial t}(p_n) \end{aligned} \quad (4.12)$$

Bu denklem sistemini matris formuna dönüştürürse aşağıdaki denklem elde edilir:

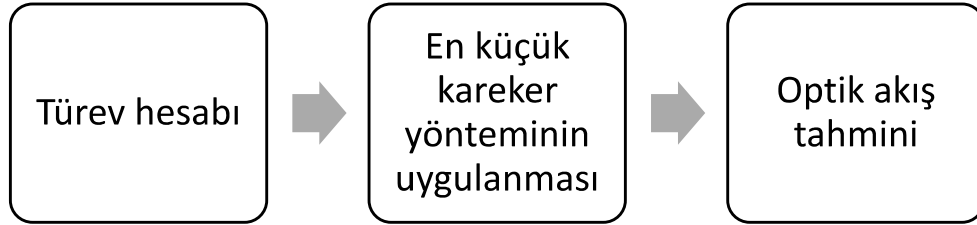
$$A = \begin{bmatrix} \frac{\partial Q}{\partial x}(p_1) & \frac{\partial Q}{\partial y}(p_1) \\ \frac{\partial Q}{\partial x}(p_2) & \frac{\partial Q}{\partial y}(p_2) \\ \dots & \dots \\ \frac{\partial Q}{\partial x}(p_n) & \frac{\partial Q}{\partial y}(p_n) \end{bmatrix}, v = \begin{bmatrix} dx \\ dy \end{bmatrix}, b = \begin{bmatrix} -\frac{\partial Q}{\partial t}(p_1) \\ -\frac{\partial Q}{\partial t}(p_2) \\ \dots \\ -\frac{\partial Q}{\partial t}(p_n) \end{bmatrix} \quad (4.13)$$

Sonuç olarak $Av=b$ elde edilir. En küçük kareler yöntemi uygulanırsa optik akış sonuca ulaşmış olur.

$$A^T A v = A^T b \Rightarrow (A^T A)^{-1} (A^T A) v = (A^T A)^{-1} (A^T b) \Rightarrow v = (A^T A)^{-1} (A^T b) \quad (4.14)$$

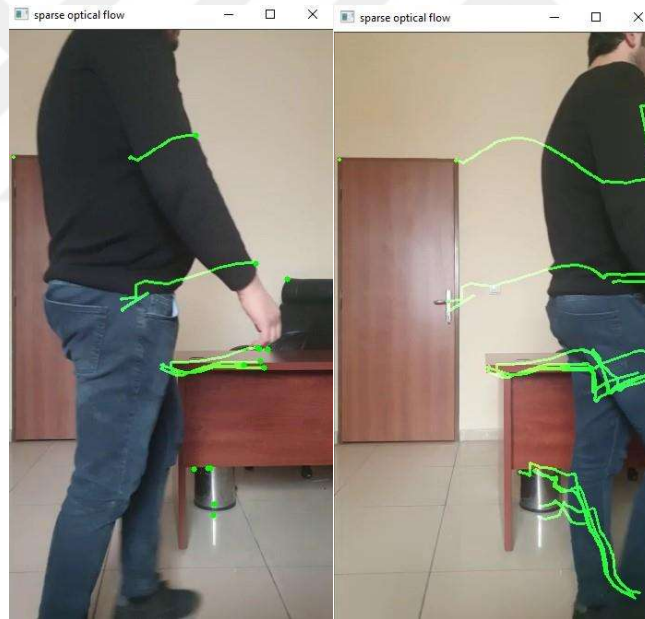
Optik Akış algoritmaları, algoritmanın sınırlamaları nedeniyle hızlı hareketlerde istenilen başarıyı sergilemez. Filtre görüntüdeki köşeleri bulmak ve ardından iki ardışık kare arasındaki

köşelerin hareket vektörünü hesaplamak için kullanılır [9] LK algoritmasının akış diyagramı Şekil 4.4 verilmiştir.



Şekil 4.2. LK algoritmasının akış diyagramı

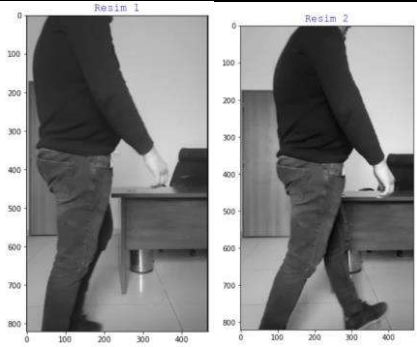
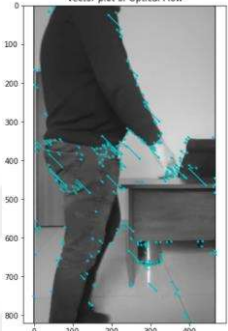
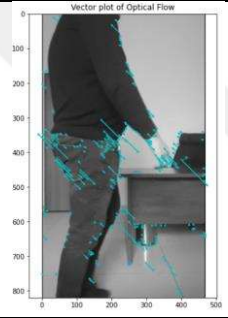
LK yöntemine uygulamasında görüntü çıktıları Şekil 4.5'te verilmiştir.



Şekil 4.3. LK uygulama çıktıları

Farklı filtre boyutlarına göre çıktı karşılaştırmaları Tablo 4.1'de verilmiştir.

Tablo 4.1. LK’de farklı filtre boyutlarına göre çıktı karşılaştırılması

Giriş Görüntüleri	
Çıkış 1 (Filtre Boyutu) = (3,3)	
Çıkış 2 (Filtre Boyutu) = (5,5)	

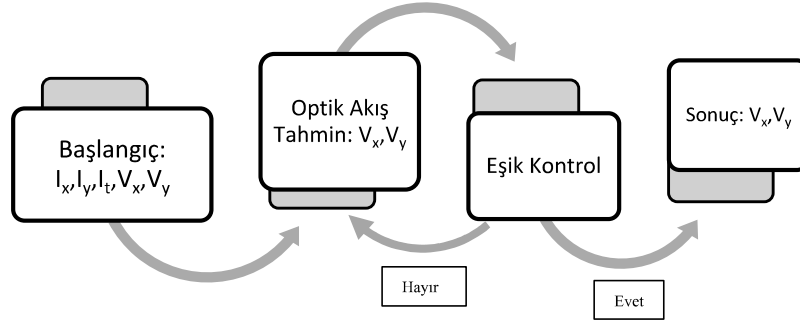
Tablo 4.1’de gösterildiği gibi filtre boyut sayısı artması ile tespit edilecek noktaların artışı gözlemlenmiştir. Ancak boyutun çok yüksek seçilmesi de akışın yanlış hesaplanmasına sebep olacaktır.

4.3.2. Horn ve Schunck Optik Akış Yöntemi

1981’de Horn ve Schunck (HS), iki boyutlu resmin gri seviyeye dönüştürerek optik akış kısıtlama denklemini elde etmişlerdir. Algoritma komşu piksellerin aynı akışa sahip olduğunu varsayımı prensibini temel alır. Bu varsayım, iki değişkenli denklem belirsizlik probleminin çözümünü elde etmeyi sağlar. HS optik akış algoritması, görüntüdeki hareketi tahmin etmek için yumuşatma kısıtlamaları tanımlar. Optik akış alanındaki her bir yerin hız değişimi, ani değişiklikler olmadan yani hareket boyunca akışta düzgünlük olduğu varsayımıyla hesaplanır.

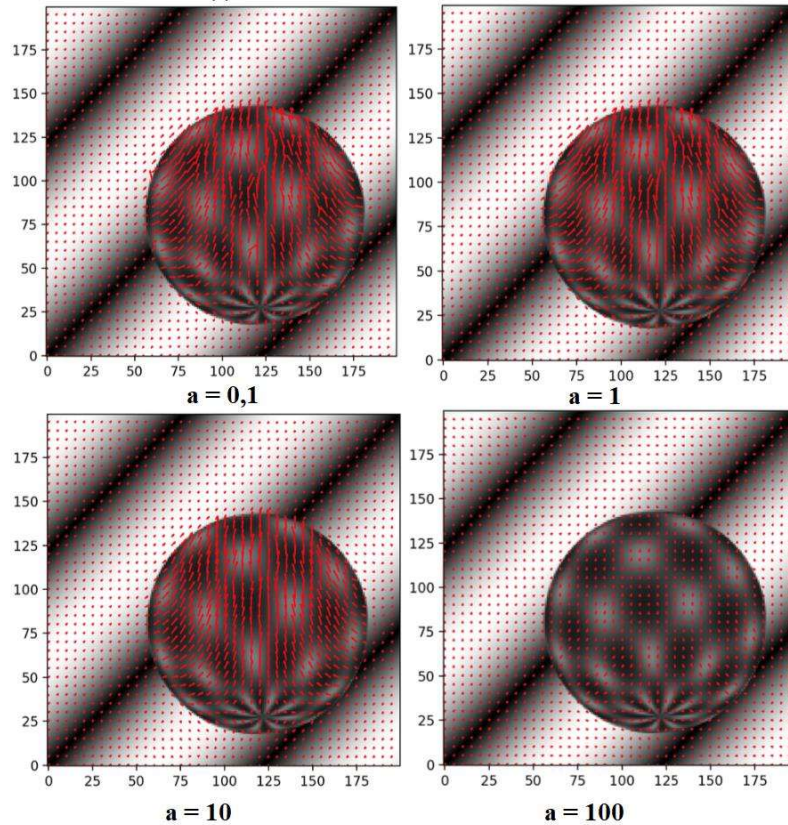
Yöntem, öncelikle başlangıç hız değişkenleri oluşturulup sıfır atanır ve türevler hesaplanır. Tekrarlama denklemi sonucunda elde edilen hız değerleri bir sonraki tekrarlama kullanılmak üzere girişe verilir. Sonuç olarak, eğer elde edilen hız değerleri farkı belirli bir eşiğin altında veya

daha önceden belirlenen maksimum tekrarlama sayısının üzerinde ise tekrarlama durur [8]. HS algoritmasının akış diyagramı Şekil 4.6’da verilmiştir.



Şekil 4.4. HS algoritmasının akış diyagramı

HS uygulama çıktıları Şekil 4.7’de verilmiştir. Şekil 4.7’de verilen çıktıya farklı yumuşatma katsayıları uygulanması sonucunda daha düzgün bir akış elde edildiği gözlemlenmiştir.



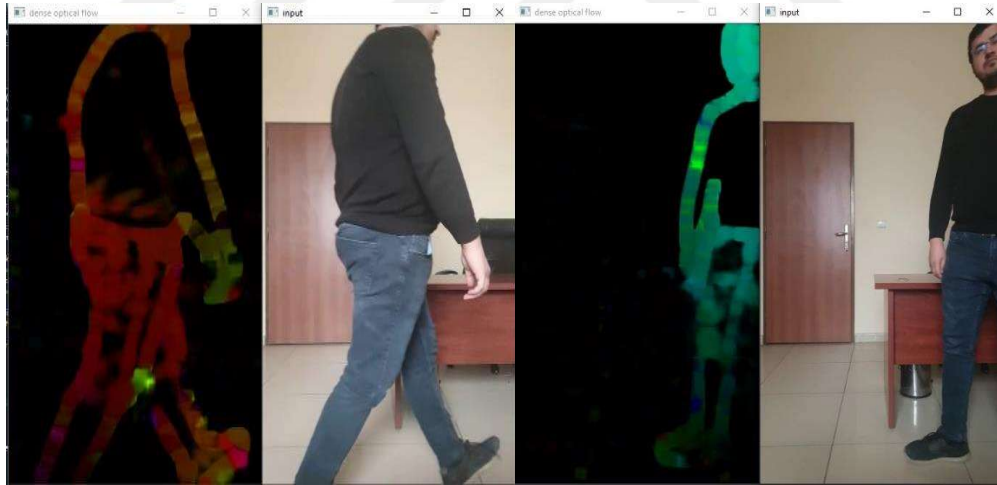
Şekil 4.5. HS uygulama çıktıları (a: yumuşatma katsayısı)

4.4. Yoğun Optik Akış

Yoğun optik akış, her çerçevenin her pikseli için optik akış vektörünü hesaplamaya çalışmaktadır. Bu tür bir hesaplama daha yavaş olabilse de daha doğru bir sonuç üretir ve hareket tespiti gibi uygulamalar için uygun ve daha doğru bir sonuç üretir. Yoğun optik akışın çeşitli uygulamaları vardır. En popüler uygulamalardan biri olan Farneback yöntemidir.

4.4.1. Farneback Optik Akış Yöntemi

Farneback ve arkadaşları [7] çalışmalarında, birbirini izleyen iki kareyi karşılaştırarak nesne hareketini tahmin etmek için etkili bir yöntem önermiştir. 2003 yılında önerilen yöntem, görüntüdeki her piksel için optik akışı hesaplamaktadır. Tüm noktalara bakılır ve iki kare arasındaki piksel yoğunluğu değişikliklerini tespit edilmektedir. Yöntem net bir görünürlük için HVS formatına dönüştürdükten sonra, vurgulanmış piksellere sahip bir görüntü ile sonuçlanır. Bu yöntemin ana fikri, pikselin bazı komşularına bir polinom ile yaklaşımdır. Buradaki amaç, polinom yaklaşımı kullanarak denklemdeki bir yer değiştirmeyi hesaplamaktır. Şekil 4.8’de uygulama çıktıları verilmiştir.



Şekil 4.6. Farneback uyguladığımız çıktıları

5. UYGULAMALAR

5.1. Optik Akış Veri Kümeleri

Optik akış hesaplaması için herkesin erişebileceği birçok veri kümesi bulunmaktadır. Yüksek başarımlar elde edilmiş en yaygın veri kümeleri Yosemite [79], Middleburn [78], MPI Sintel [21], KITTI [33] ve Flying Chairs (Uçan Sandalyeler) [12] veri kümeleridir. Optik akış veri kümelerinin görüntü ve gerçeklik bilgileri Tablo 5.1’de verilmiştir. Çalışmada kullanılan üç veri kümesi detayları ile anlatılmaktadır.

Tablo 5.1. Optik akış veri kümeleri ve içerikleri

Veri Kümesi	Görüntü Çiftleri Sayısı	Etiketli Görüntü Sayısı	Gerçeklik
Yosemite	12	12	Sentetik Veri
Middleburn	21	21	Gerçek Dünya Verisi
KITTI	194	194	Gerçek Dünya Verisi
Sintel	1041	1041	Sentetik Veri
Flying Chairs	22.872	22.872	Sentetik Veri

5.1.1. MPI Sintel Veri Kümesi

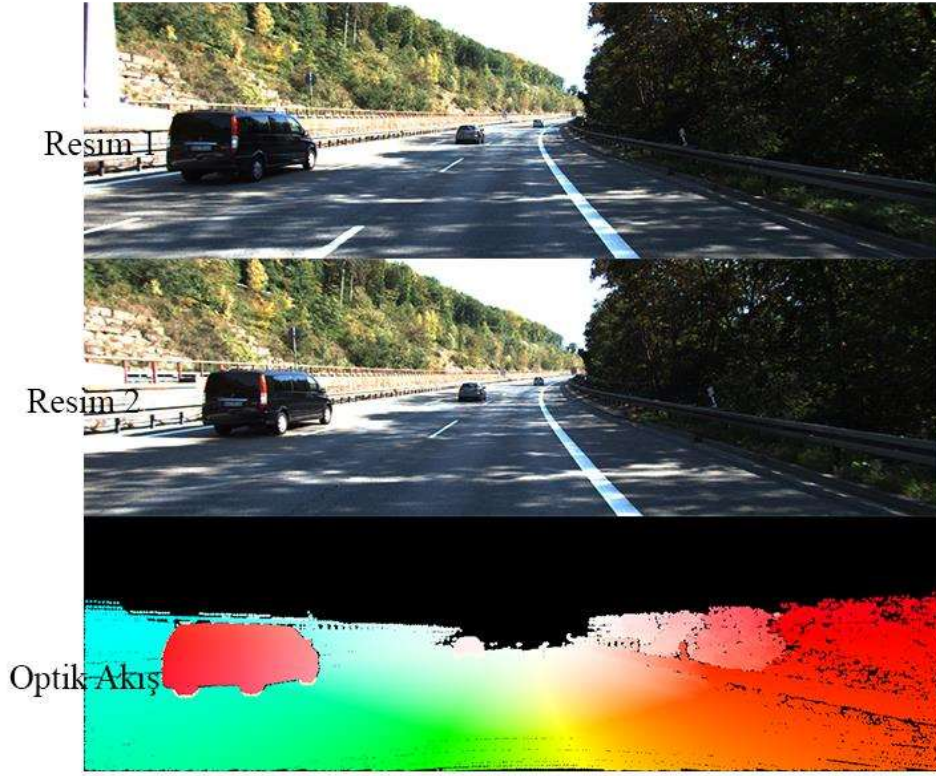
MPI Sintel veri kümesi, gerçekçi görüntü özelliklerine önem verilerek işlenmiş yapay sahnelerden temel gerçeği elde edilmesini sağlayan etiketli sentetik bir veri kümesidir. Sintel’in iki sürümü bulunmaktadır. Son sürüm hareket bulanıklığı ve sis gibi atmosferik efektler içerirken, ilk sürüm bu efektleri içermez. Sintel veri kümesi her sürüm için 1.041 görüntü çiftini içermektedir. Sintel veri kümesinin görüntü ve etiketli akış görüntüleri Şekil 5.1’de verilmiştir [21].



Şekil 5.1. Sintel veri kümesinin görüntü ve etiketli akış görüntüleri

5.1.2. KITTI Veri Kümesi

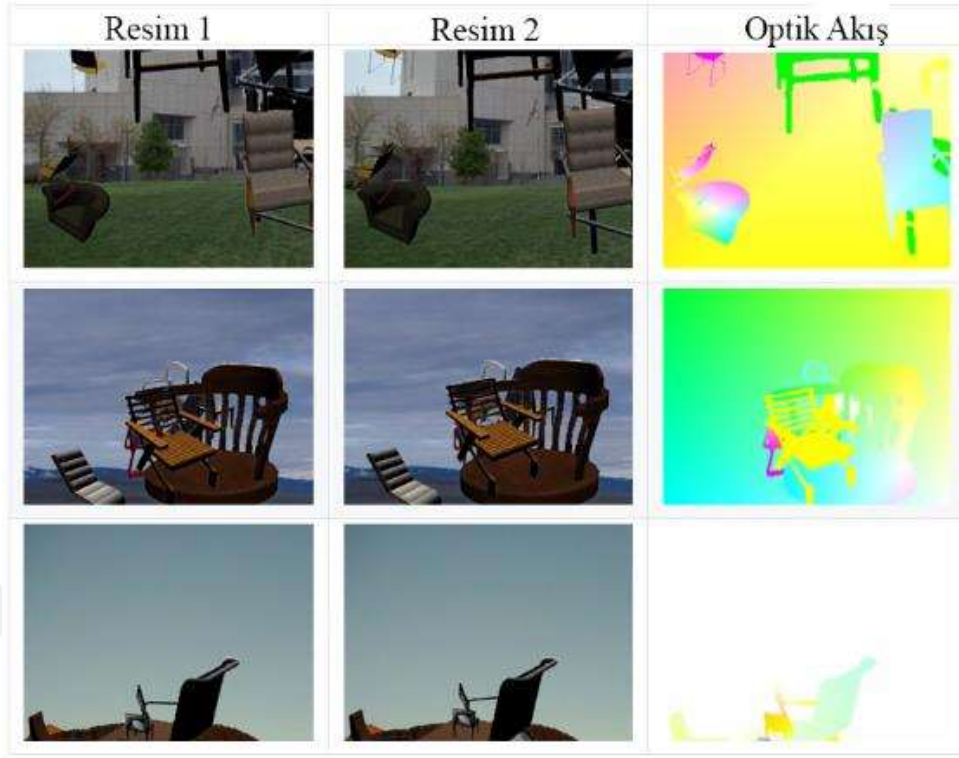
Sentetik olmayan bir yapıda gerçek dünya verileri ile oluşturulmuş verilerden oluşan veri kümesidir. Araç üzerinde bağlı kameralar ile sahne kayıtları alıp bunların bir araya getirilmesi sonucu elde edilmiş bir veri kümesidir. KITTI veri kümesi 2012 ve 2015 iki veri kümesinden oluşmaktadır. KITTI Flow 2012 veri kümesi, png(Portable Network Graphics) formatında kaydedilen 194 eğitim görüntü çifti ve 194 test görüntü çiftinden oluşur. KITTI Flow 2015 ise yine png formatında 200 eğitim görüntü çifti ve 200 test görüntü çiftinden oluşmaktadır. KITTI veri kümesine ait görüntüler ve veri kümesinde bulunan etiketli veriler Şekil 5.2’de verilmiştir [33].



Şekil 5.2. KITTI veri kümesine ait görüntüler ve veri kümesinde bulunan etiketli veriler

5.1.3. Flying Chairs (Uçan Sandalyeler) Veri Kümesi

Uçan sandalyeler, optik akış hesaplamalarında kullanılmak üzere üretilmiş sentetik veri setidir. 22872 görüntü çifti ve bu görüntü çiftlerine karşılık gelen akış alanlarından oluşur. Dosovitskiy ve arkadaşları [12], yeterli eğitim verisi sağlamak için, yaygın bir kullanıma sahip çevrimiçi fotoğraf yönetimi ve paylaşım uygulaması olan Flickr'den toplanan görüntülere, 3 boyutlu sandalye modellerinin görüntülerine dönüşümler uygulayarak "Uçan Sandalyeler" adını verdikleri basit bir sentetik veri kümesi oluşturmuşlardır. Flickr2'den 1024×768 çözünürlüğe sahip 964 görüntüyü "şehir" (321), "manzara" (129) ve "dağ" (514) kategorilerinden almışlardır. Görüntüleri 4 parçaya ayırıp elde edilen 512×384 görüntü kırıpmalarını arka plan olarak kullanmışlardır. Arka plana birden çok sandalyenin görüntülerini ekleyerek veri setini oluşturmuşlardır. Uçan sandalye veri kümesine ait görüntüler ve veri kümesinde bulunan etiketli veriler Şekil 5.3' verilmiştir.



Şekil 5.3. Uçan Sandalye veri kümesine ait görüntüler ve veri kümesinde bulunan etiketli veriler

5.2. Geliştirme Ortamı ve Optik Akış Değerlendirme Parametreleri

5.2.1. Colaboratory (Google Colab)

Colaboratory (Colab), tarayıcı tabanlı Python uygulaması yazma ve çalıştırma hizmeti veren Google destekli bir yazılımdır. Colab, hiç yapılandırma gerektirmeden GPU ve TPU üzerinden uygulamayı çalıştırmaya olanak veren ve bunlara ücretsiz erişim sağlayan bir geliştirme ortamıdır. Colab ile popüler Python kütüphanelerinin tüm avantajlarından yararlanarak veri analiz edip görselleştirebilmeyi sağlamaktadır.

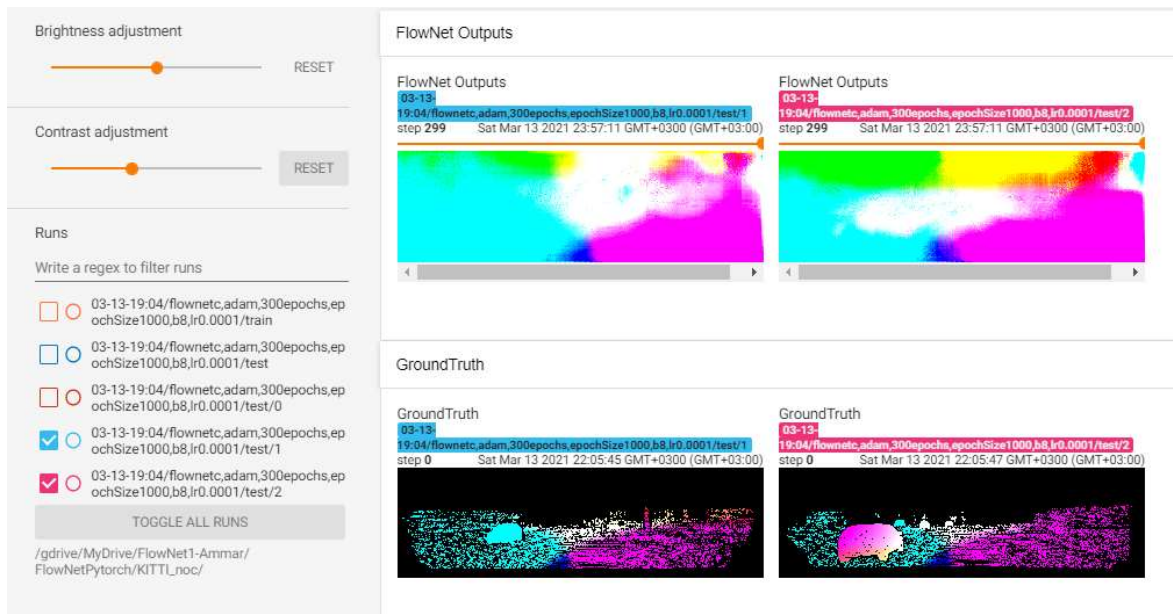
Colab az sayıda kod ile veri kümesini içe aktarabilir, sınıflandırıcıyı eğitebilir ve model değerlendirilebilmektedir. Colab not defterleri Google'ın bulut sunucularında kodu çalıştırmasından dolayı istenildiği gibi çalışma ortamı paylaşılabilir. Makinenin gücünden bağımsız olarak, GPU'lar ve TPU'lar dahil Google donanımının ayrıcalıklarından yararlanabilmektedir. Uygulama geliştirmek için kurulumlara gerek kalmadan sadece tarayıcı üzerinden çalıştırma yapılabilir. Aynı zamanda birçok Python kütüphaneyi içinde barındırdığından dolayı kullanımı oldukça basittir. Tesla K80 GPU kullanılmaktadır. 4992 Cuda core bulunmakta ve dual GPU tasarımına sahiptir.

5.2.2. TensorBoard Gösterim Aracı

Derin öğrenmede bir uygulama geliştirmek için genellikle ölçebilmek gerekmektedir. Ölçek modelin performans ve hatalarını görünmesine yardımcı olmaktadır.

TensorBoard, derin öğrenme iş akışı sırasında ihtiyaç duyulan ölçümleri ve görselleştirmeleri sağlayan google destekli bir araçtır.

Kayıp ve doğruluk gibi deney ölçümlerinin izlenmesini, model grafiğinin görselleştirilmesini ve çok daha fazlasını sağlar. FlowNet modelinin FlowNetC ağı için ölçümlerin sonuçlarının gösterimi Şekil 5.4'te verilmiştir.



Şekil 5.4. Tensorboard gösterim aracı çıktı ekranı

5.2.3. Son Nokta Hatası (End-point Error)

Optik akış ile hareket tespit modellerinde temel alınan başarımlar göstergelerden bir tanesidir. Son nokta hatası (EPE), tahmini ve gerçek akışı arasındaki ortalama Öklid mesafesi olarak tanımlanır.

Bir akış tahmininin bir temel gerçek akışına ne kadar yakın olduğunu belirlemek için bir ölçüt olarak kullanılan uç nokta hatasını (EPE) simgeler. V_{est} tahmin edilen optik akış ve V_{gt} de etiketli optik akış olsun bu durumda denklem 5.1'de gösterildiği gibi EPE hesaplanır.

$$||V_{est} - V_{gt}|| \quad (5.1)$$

5.2.4. Kayıp (Loss-L1)

Optik akış ile hareket tespit modellerinde temel alınan başarımlar göstergelerden bir tanesidir. Kayıp, gerçek akış ile tahmin arasındaki L1 mesafesi olarak tanımlanır. Tüm tahmin hücre çıktıları optik akış tahminlerinin bir dizisini oluşturur. Bu dizi aşağıdaki gibi ifade edilir.

$$f_1, \dots, f_N \tag{5.2}$$

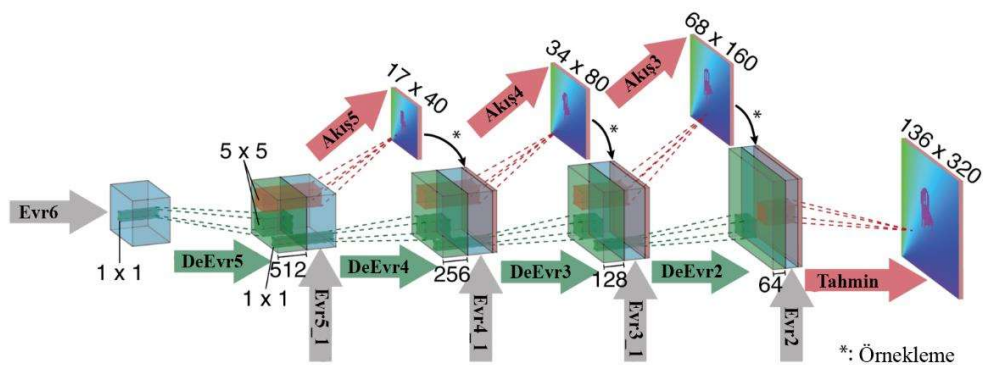
Toplam kayıp, gerçek optik akış ile tahmin arasındaki her tekrarlayan blok çıktısındaki kayıpların toplamıdır denklemler 5.3'te gösterildiği gibidir.

$$L = \sum_{i=1}^N x^{i-N} \|gt - f_i\|_1, x = 0.8 \quad (5.3)$$

5.3. İlk Derin Öğrenme Tabanlı Optik Akış Mimarisi (FlowNet)

5.3.1. Mimari Yapısı

FlowNet mimarisi, optik akışı tahmin etmek için ilk CNN yaklaşımı olarak 2015 yılında tanıtıldı. Optik Akış tahmini için FlowNetS (FlowNetSimple) ve FlowNetC (Corr) adında iki sinir ağı geliştirildi. FlowNetSimple ve FlowNetCorr adı verdikleri iki mimarinin her ikisi de uçtan uca öğrenme temellidir. FlowNetSimple, Şekil 3.1'de gösterildiği gibi, sıralı olarak iki bitişik giriş görüntüsünü bir araya toplanıp ve bunları ağ üzerinden beslemektedir. FlowNetSimple ile karşılaştırıldığında, FlowNetCorr ise önce iki görüntünün akışını ayrı ayrı üretir ve ardından bunları korelasyon katmanında birleştirir. İki mimarinin de çözünürlükleri yükseltmek için iyileştirmelere sahiptir. Korelasyon katmanı, iki özellik haritası arasında karşılaştırma gerçekleştirmek oluşturulmuştur. İki ağ modelinde de bilgi önce uzamsal olarak ağın daralan bir bölümünde sıkıştırılır. Daha sonra genişleyen bir bölümde rafine edilir [12]. Rafine işleminin detayı gösterimi Şekil 5.5'te verilmiştir.

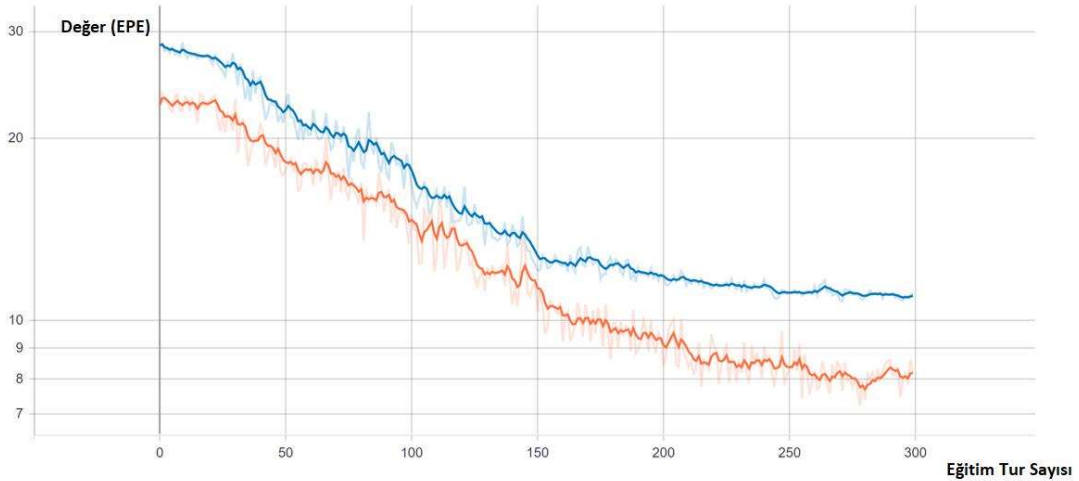


Şekil 5.5. Rafine işleminin detayı gösterimi [12]

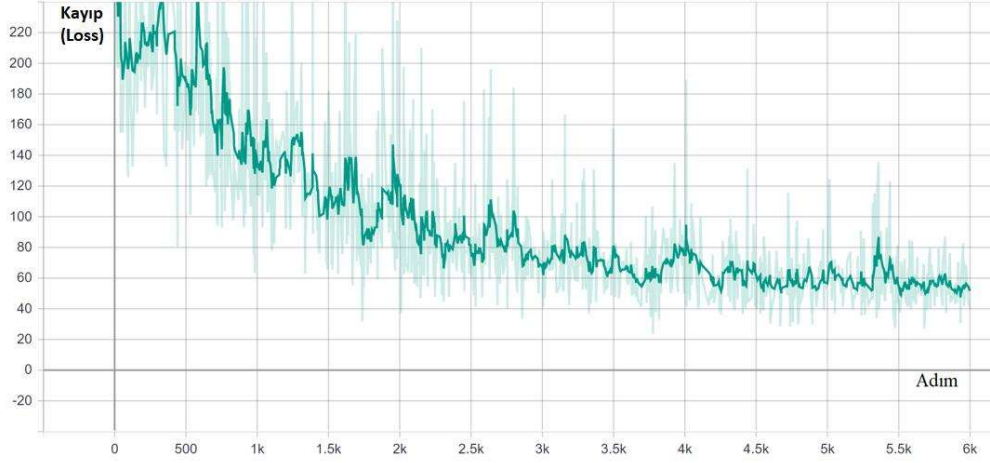
5.3.2. Modellerin Eğitilmesi ve Sonuçları

Model eğitimi Google Colab ortamında, çıktılar ise Tensorboard ile görselleştirilmiştir. Model geliştirilmesi Python dili ile yapılmış olup model kurulumu eğitim işlemleri için Python'un pytorch kütüphanesi kullanılmıştır. FlowNetC ve FlowNetS ağlarını eğitmek ve analiz etmek için KITTI veri setlerini kullanılmıştır. Sintel ve Uçan sandalyeler veri setinin eğitilmiş ağırlıkları ile farklı görsellere uygulayıp karşılaştırmalarda bulunulmuştur. Eğitim verilerini hem FlowNetC hem de FlowNetS'e uygulayıp görsel karşılaştırmalarda bulunulmuştur. FlowNetC ve FlowNetS'in ortalama son nokta hatası (EPE) ve eğitim kayıp (loss) değerleri elde edilmektedir. Ağ eğitimi için farklı hiper parametreler uygulanıp çıktılar karşılaştırılmıştır. Hiper parametre olarak; Optimizasyon algoritması "Adam", Epoch (döngü) sayısı olarak KITTI eğitim setine 1000 ve 2700 olarak iki farklı şekilde parametreler uygulanmıştır. Her iki eğitimde de Batch size 8, öğrenme hızı (learning rate) 0.0001, Eğitim Turu olarak 300 değerleri verilmiştir.

İlk olarak KITTI veri setine döngü sayısı 1000 verilerek ağ eğitilmiştir. FlowNetS ağının optimizasyon algoritması olarak "Adam", KITTI veri setinin eğitilmesi sonucu elde edilen eğitim ve testin EPE değerleri Şekil 5.6'te verilmiştir. Grafikte turuncu eğitim, mavi ise test EPE değerlerini göstermektedir. FlowNetS ağının eğitim kayıp değerlerinin grafiği Şekil 5.7'te verilmiştir. Eğitim süresi olarak Google Colab GPU ile 1 saat 37 dakika sürmüştür.

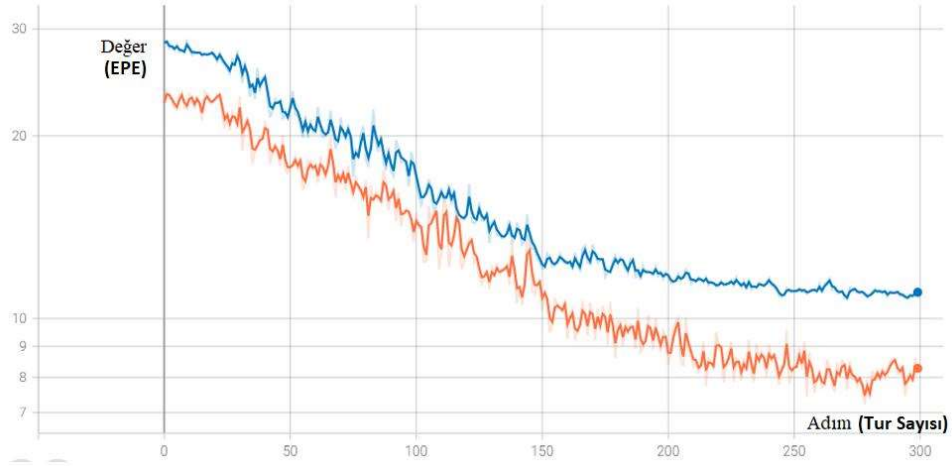


Şekil 5.6. FlowNetS ağının KITTI veri setine göre Epe değerleri (Epoch Size 1000)

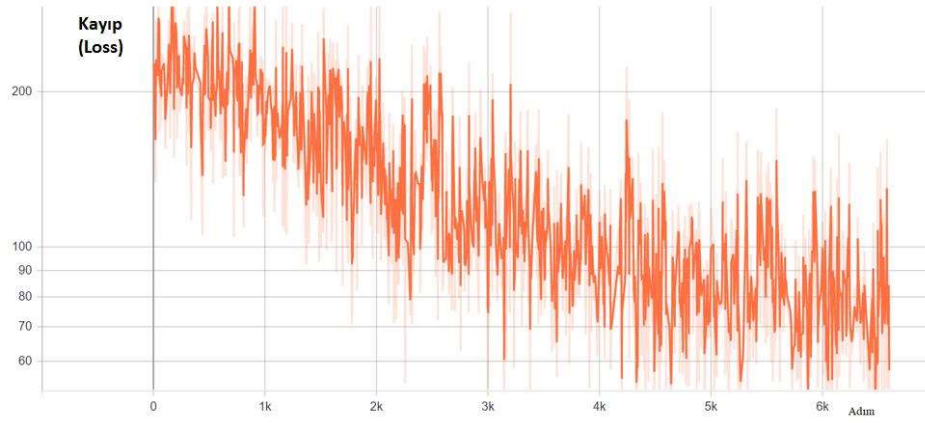


Şekil 5.7. FlowNetS ağının KITTI veri setine göre kayıp (loss) değerleri (Epoch Size 1000)

FlowNetC ağının optimizasyon algoritması olarak “Adam”, Epoch (döngü) sayısı olarak 1000, veri seti olarak KITTI veri setinin eğitilmesi sonucu elde edilen eğitim ve testin EPE değerleri Şekil 5.8’te verilmiştir. Grafikte turuncu eğitim, mavi ise test EPE değerlerini göstermektedir. FlowNetC ağının eğitim kayıp değerlerinin grafiği Şekil 5.9’te verilmiştir. Eğitim süresi olarak Google Colab GPU ile 1 saat 51 dakika sürmüştür.

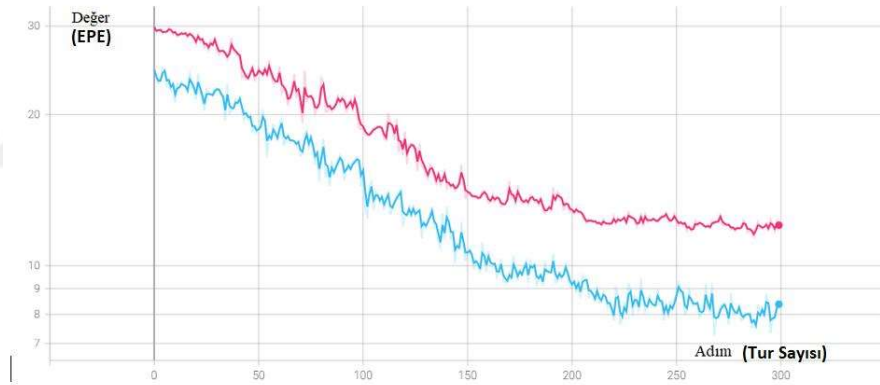


Şekil 5.8. FlowNetC ağının KITTI veri setine göre Epe değerleri (Epoch Size 1000)



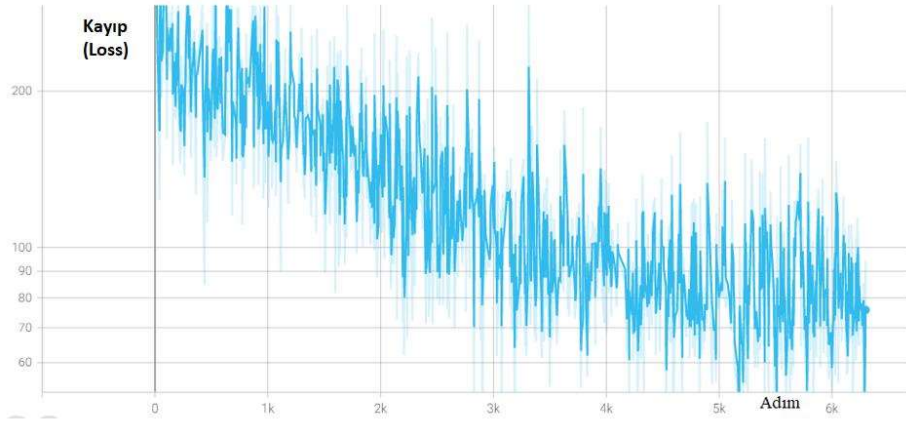
Şekil 5.9. FlowNetC ağının KITTI veri setine göre kayıp (loss) değerleri (Epoch Size 1000)

İkinci eğitim için Epoch (döngü) sayısı olarak 2700 alınarak elde edilen eğitim ve testin EPE değerleri FlowNetC için Şekil 5.10’da verilmiştir. Mavi eğitim verilerini kırmızı ise test verilerini simgelemektedir.



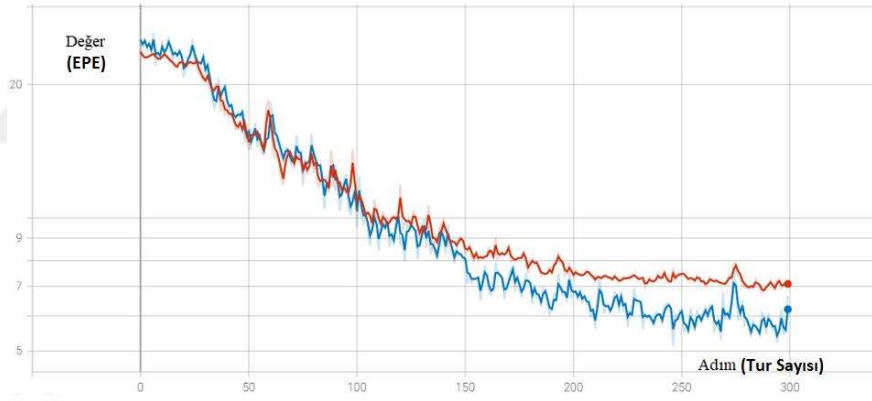
Şekil 5.10. FlowNetC ağının KITTI veri setine göre Epe değerleri (Epoch Size 2700)

Epoch (döngü) sayısı olarak 2700 alınarak FlowNetC modelinin kayıp değeri Şekil 5.11’te verilmiştir.

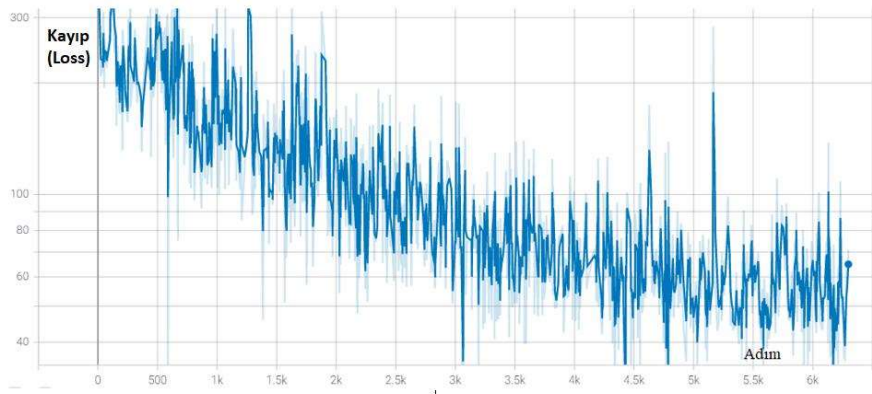


Şekil 5.11. FlowNetC ağının KITTI veri setine göre kayıp (loss) değerleri (Epoch Size 2700)

Epoch (döngü) sayısı olarak 2700 alınarak elde edilen eğitim ve testin EPE değerleri FlowNetS için Şekil 5.12’da verilmiştir. Mavi eğitim verilerini Kırmızı ise test verilerini simgelemektedir. Şekil 5.13’te kayıp değeri verilmiştir.



Şekil 5.12. FlowNetS ağının KITTI veri setine göre Epe değerleri (Epoch Size 2700)



Şekil 5.13. FlowNetS ağının KITTI veri setine göre kayıp (loss) değerleri (Epoch Size 2700)

FlowNetC ve FlowNetS ağlarının 1000 ve 2700 epoch değerlerine göre colab ortamındaki “karşılaştırması Tablo 5.2’de verilmiştir.

Tablo 5.2. FlowNetS ve FlowNetC KITTI veri seti için EPE karşılaştırılması

Model Adı	Ortalama EPE Değeri	Epoch Boyutu (Size)	Süre(s/d/s)
FlowNetC	8,248	1000	1s 51d 37s
FlowNetS	5,062	1000	1s 37d 26s

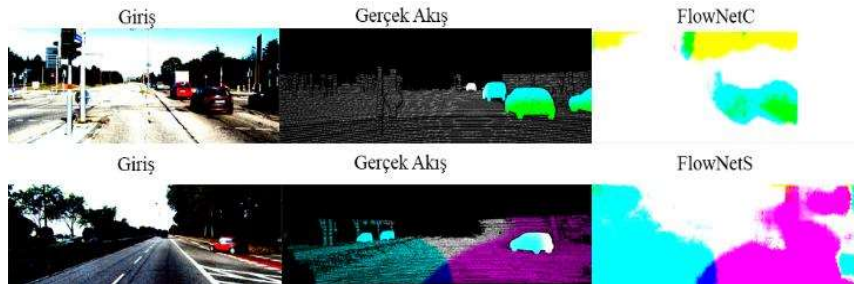
Tablo 5.3’te FlowNetC ve FlowNetS modellerinin 2700 epoch değerlerine göre KITTI ve Uçan sandalye veri setlerine göre EPE değerlerinin karşılaştırması gösterilmektedir.

Tablo 5.3. FlowNetS ve FlowNetC KITTI ve Uçan Sandalye veri seti için karşılaştırılması

Model Adı	Veri Seti	Öğrenme Oranı	Epoch Boyutu	Batch Boyutu	EPE Değeri
FlowNetS	Uçan Sandalyeler	1e-4	2700	8	1,951
FlowNetC	Uçan Sandalyeler	1e-4	2700	8	1,766
FlowNetS	KITTI	1e-4	2700	8	6,677
FlowNetC	KITTI	1e-4	2700	8	8,495

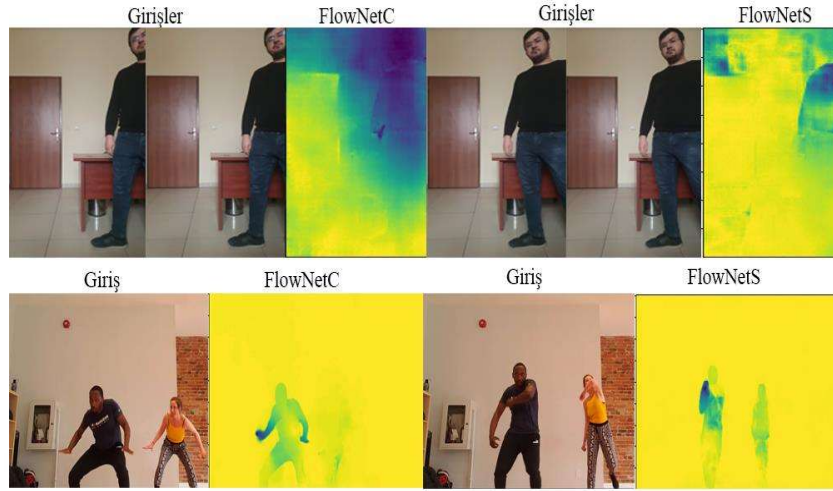
5.3.3. Görsel Çıktılar

KITTI veri seti eğitimine göre eğitilmiş FlowNetC ve FlowNetS modellerinin optik akış tahmini örnekleri Şekil 5.14’te verilmiştir.



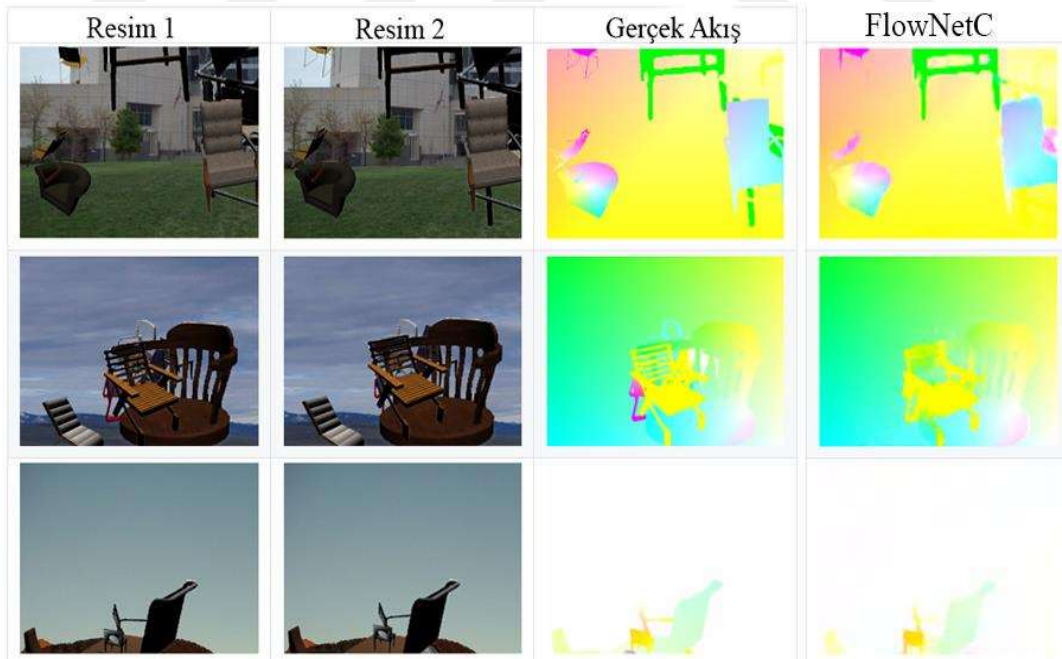
Şekil 5.14. KITTI veri seti eğitim verilerine göre FlowNet çıktıları

Uçan Sandalyeler veri seti eğitime göre optik akış tahmini örnekleri Şekil 5.15'te verilmiştir.



Şekil 5.15. Uçan Sandalyeler veri kümesine göre FlowNet uygulama çıktıları

Uçan Sandalyeler veri seti eğitime göre giriş olarak uçan sandalye verileri işlenmesi ile optik akış tahmini örnekleri Şekil 5.16'da verilmiştir.



Şekil 5.16. Uçan sandalye verileri ile optik akış tahmini ve çıktıları

5.4. Derin Öğrenme Tabanlı Optik Akış Mimarisi (FlowNet2)

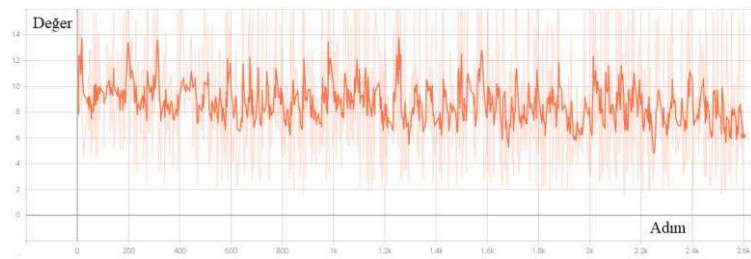
5.4.1. Mimari Yapısı

FlowNet2, Şekil 2.2’de verildiği gibi optik akışı hesaplamak için FlowNetC ve birden çok FlowNetS’i birleştirmesi ile oluşturulmuştur. Küçük yer değiştirmelerle en iyi şekilde başa çıkmak için FlowNetS mimarisine başlangıçta daha küçük adımlar ve yukarı dönüşümler arasında evrişimler eklemişlerdir. FlowNet2, FlowNet geliştiricileri tarafından önerilmiş bir yöntemdir. Yaptıkları çalışmada optik akışın uçtan uca (end-to-end) denetimli öğrenilmesi kavramını ilerletmiş ve derin öğrenme tabanlı optik akış ile hareket tespiti yöntemin daha iyi çalışmasını sağlamışlardır. Çalışmalarında FlowNet yönteminin küçük görüntülerde doğru çalışmadığı konusunda yapmış oldukları eleştiriye istinaden küçük hareketler konusunda uzmanlaşmış bir alt ağ modeli geliştirmişlerdir [13].

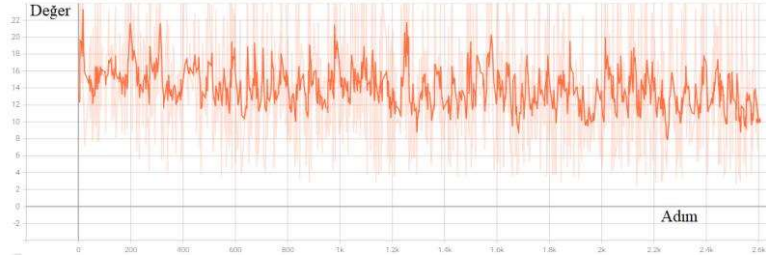
5.4.2. Model Eğitimi ve Sonuçlar

Model eğitimi, Google Colab ortamında çıktılar ise tensorboard ortamında görselleştirilmiştir. Model geliştirilmesi Python dili ile yapılmış olup model kurulumu eğitim işlemleri için Python’un pytorch kütüphanesi kullanılmıştır. Model eğitimi Sintel veri seti üzerinde eğitilmiş ve çıktılar alınmıştır.

Modelde hiper parametre olarak; optimizasyon algoritması Adam, epoch(döngü) sayısı olarak Sintel eğitim setinde 10 olarak alınmış, batch size 8, öğrenme hızı (Learning Rate) 0.0001 olarak seçilmiştir. Şekil 5.17’te FlowNet2 ağının eğitim ortalama kayıp değerlerinin grafiği gösterilmektedir. Şekil 5.18’te FlowNet2 ağının Sintel veri seti için FlowNet2 EPE değerlerinin grafiği gösterilmektedir. Eğitim süresi olarak Google Colab GPU ile 49 dakika sürmüştür.



Şekil 5.17. Sintel veri seti eğitim verilerine FlowNet2 ortalama kayıp (L1)



Şekil 5.18. Sintel veri seti eğitim verilerine FlowNet2 EPE

Eğitim sonucunda FlowNet2 modelinin Sintel veri setinde istenilen başarıyı sergilemediği gözlenmiştir. Bunun sebebi hiper parametre seçimleri olduğu düşünülmektedir.

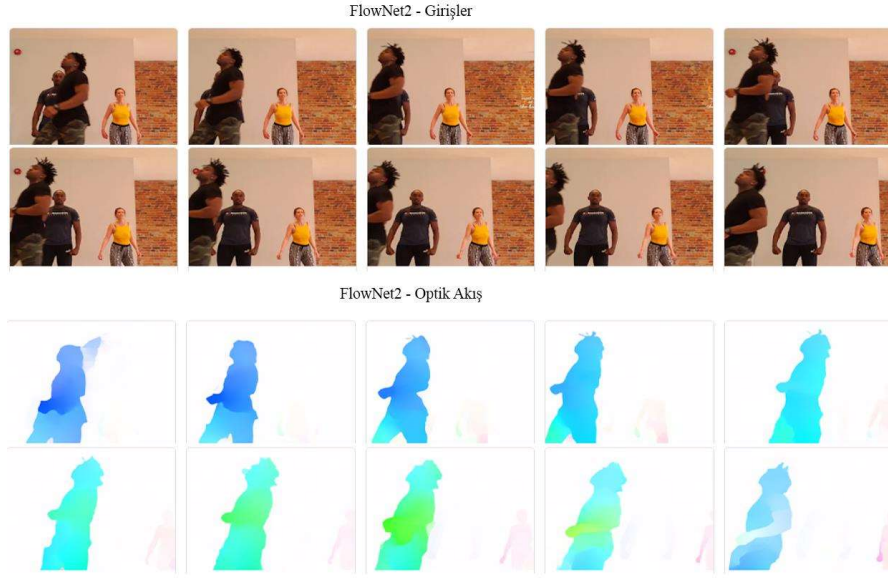
5.4.3. Görsel Çıktılar

Sintel veri seti için eğitilmiş verilere göre elde edilen ağırlıklar ile optik akış tahmini örnekleri, Şekil 5.19 verilmiştir.

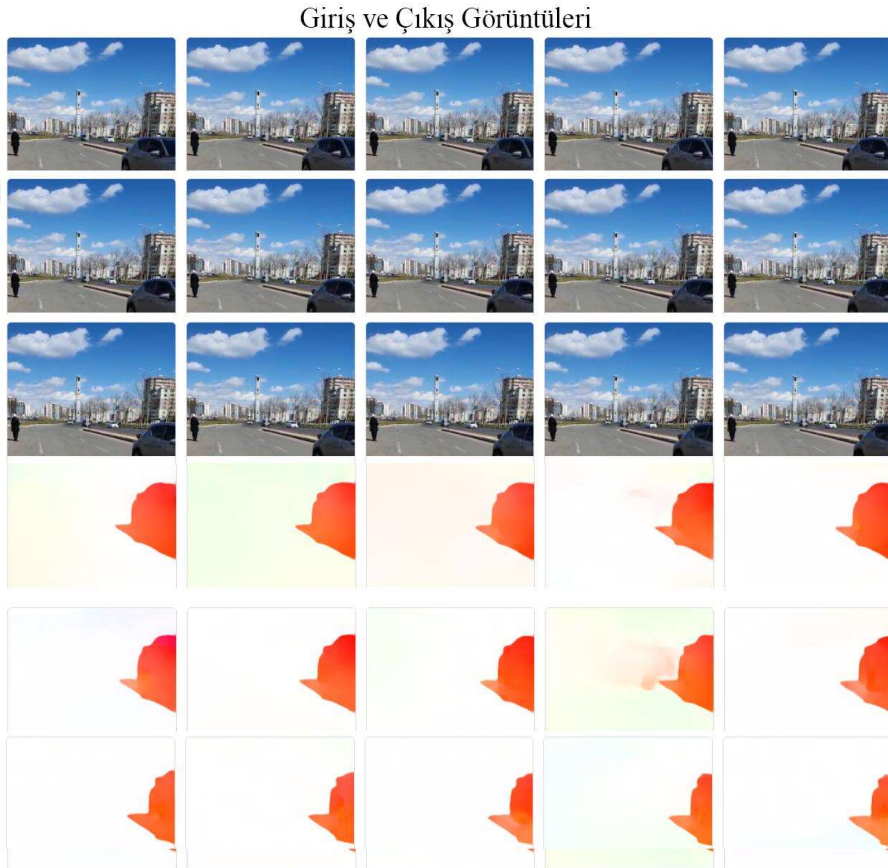


Şekil 5.19. FlowNet2, Sintel veri setine göre çıktılar

Uçan Sandalyeler veri seti eğitimine göre optik akış tahmini örnekleri Şekil 5.20 ve Şekil 5.21’te verilmiştir.

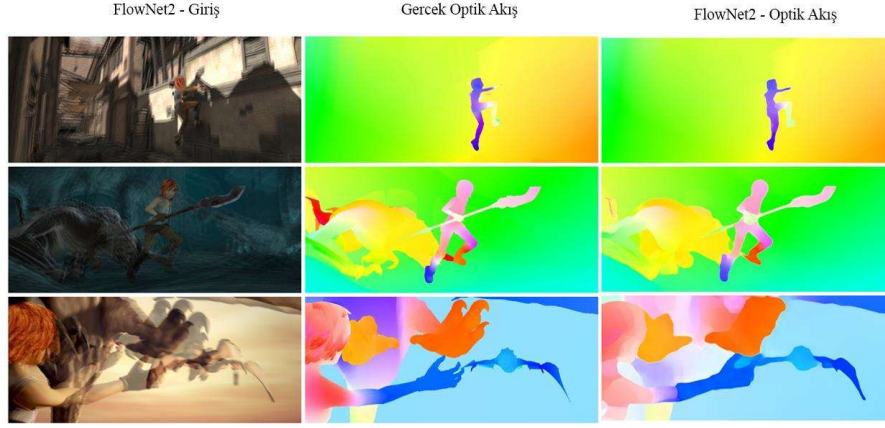


Şekil 5.20. Uçan Sandalyeler veri seti eğitime göre optik akış tahmini



Şekil 5.21. Uçan Sandalyeler veri seti eğitime göre optik akış tahmini

Sintel veri seti eğitimine göre Sintel verileri ile gerçek akış ve optik akış tahmini örnekleri Şekil 5.22’de verilmiştir.



Şekil 5.22. Uçan sandalye veri setine göre eğitilmiş ağırlıklara Sintel veri seti uygulama çıktıları

5.5. CNN ve RNN Tabanlı Optik Akış Mimarisi -RAFT Mimarisi

5.5.1. Mimari Yapısı

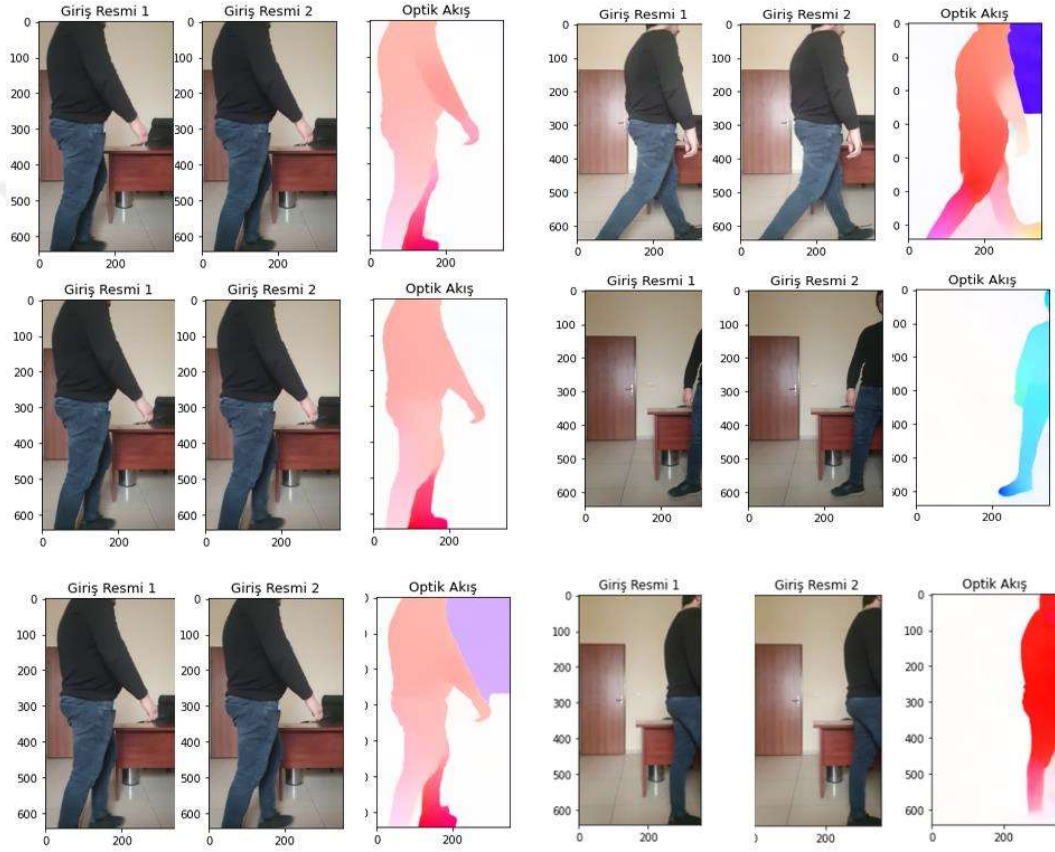
Teed ve arkadaşları [16], 2020 yılında SINTEL kıyaslamalarına göre iyi sonuçlar elde etmiş RAFT adında CNN ve RNN mimarilerinin birleşimi sonucunda yeni bir model geliştirmişlerdir. RAFT yapısı üç aşamada gerçekleşmektedir. İlk olarak özellik çıkarımı aşaması gelmektedir. Özellik çıkarma aşamasında modele iki ardışık görüntü alınır. Bu iki görüntüden özellikleri çıkarmak için, paylaşılan ağırlıklara sahip iki CNN kullanılmışlardır. İkinci aşama olarak görsel benzerlik adımı gelmektedir. Üçüncü aşamada ise yinelemeli güncelleme adımı gelmektedir. Yinelemeli güncelleme, daha önce hesaplanan tüm verileri birleştiren Geçitli Tekrarlayan Birim (GRU) hücreleri dizisidir. RAFT mimari yapısı Şekil 2.7’de verilmiştir.

5.5.2. Görsel Çıktılar

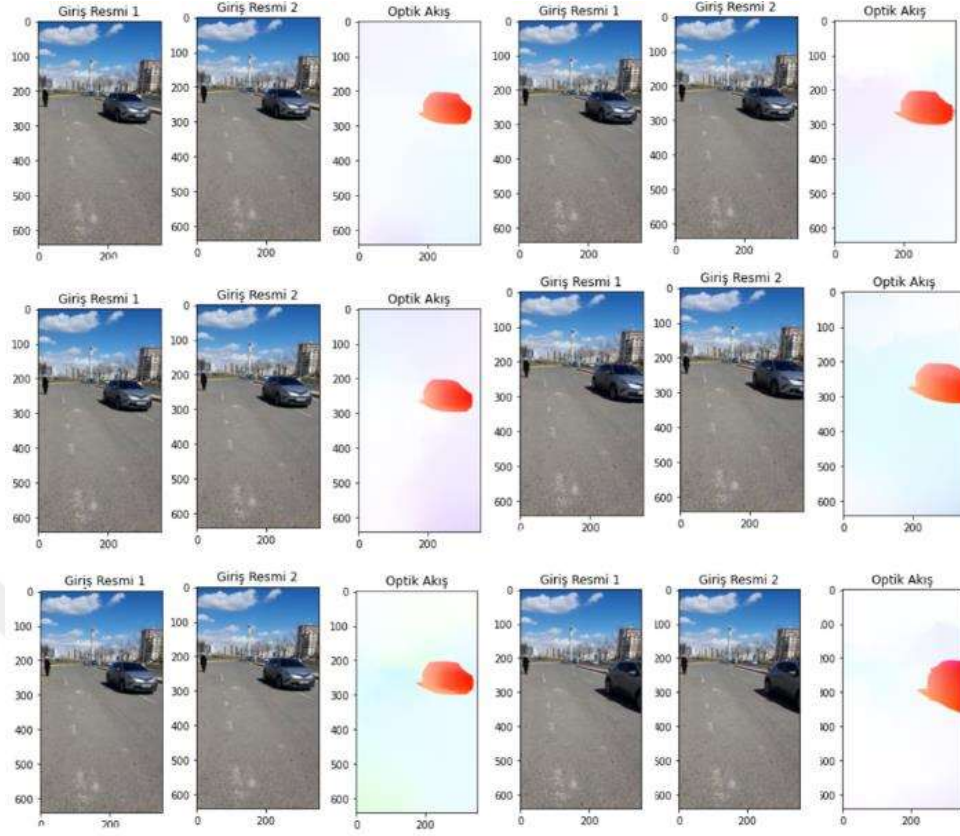
Sintel veri seti göre eğitilmiş ağırlıklar ile optik akış tahmini örnekleri Şekil 5.24’te ve Şekil 5.25’te gösterildiği gibidir. Ayrıca KITTI veri setinde bulunan görsellerin çıktıları Şekil 5.23’te gösterildiği gibidir.



Şekil 5.23. KITTI veri seti verilerine RAFT çıktıları



Şekil 5.24. Sintel veri seti eğitim verilerine RAFT çıktıları



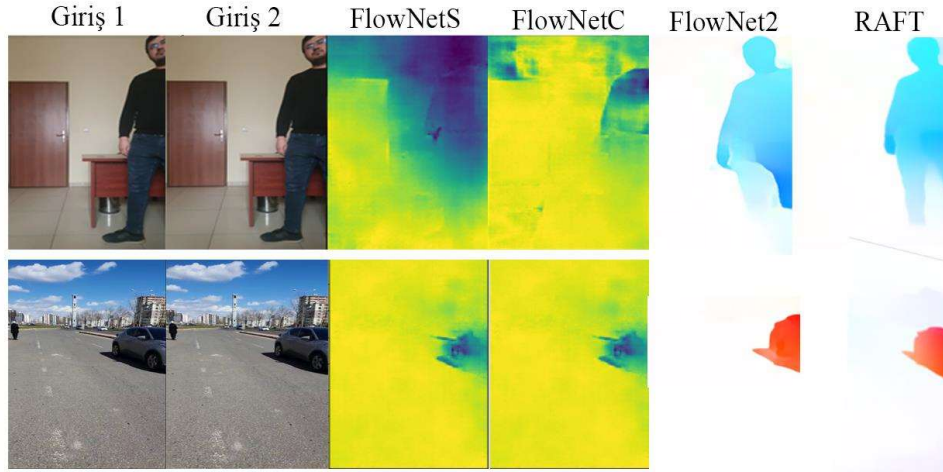
Şekil 5.25. Sintel veri seti eğitim verilerine RAFT çıktıları

6. BULGULAR VE TARTIŞMA

Yapılan çalışmada derin öğrenme tabanlı optik akış ile hareket tespit yöntemleri, farklı veri setleri üzerine kronolojik sıralı şekilde uygulanmıştır. Veri setleri ile eğitilen modeller hareket bulunan görüntüler üzerine uygulanmış ve aralarında görsel karşılaştırmalar yapılmıştır. FlowNetC, FlowNetS KITTI üzerine, FlowNet2 ise Sintel veri seti üzerinden eğitilmiş; eğitim sonucunda kayıp (L) ve EPE değerleri elde edilmiştir. Çıktılar sonucunda RAFT modelinin, FlowNetS, FlowNetC ve FlowNet2’den daha başarılı sonuçlar elde ettiği gözlemlenmiştir. RAFT, FlowNet2’ye göre daha az parametre ve kısa sürede sonuçlar ürettiği gözlemlenmiştir. Sintel veri seti için RAFT ve FlowNet2 modellerinin karşılaştırmaları Tablo 6.1’de verilmiştir. Aynı takım tarafından geliştirilen FlowNet2’nin de FlowNetS ve FlowNetC ’ye göre daha başarılı sonuçlar elde ettiği gözlemlenmiştir.

Tablo 6.1. Sintel veri seti için RAFT ve FlowNet2 modellerinin karşılaştırmaları

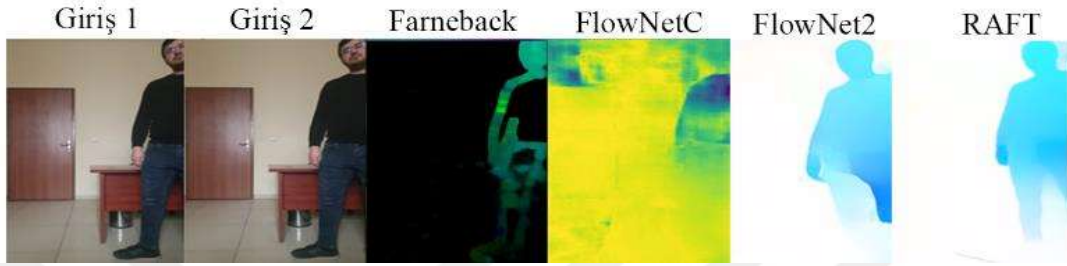
Mimari	Hesaplanan Parametre Sayısı(M)	EPE (~)	Tespit Zamanı (Sn)
FlowNet2	160	3,5	0.15
RAFT	5,3	2,8	0.10



Şekil 6.1. FlowNetS, FlowNetC, FlowNet2 ve RAFT modellerinin çıktıları

FlowNetS, FlowNetC, FlowNet2 ve RAFT modellerine aynı giriş görüntüleri verilerek işlenmesi sonucu elde edilen çıktılar Şekil 6.1’de gösterilmiştir. Şekil 6.1’de gösterilen renk haritası optik akış renk ve şiddet haritasında açıklanmıştır.

Klasik yöntemlerden Farneback ile derin öğrenme tabanlı yöntemlerden FlowNet, FlowNet2 ve RAFT yöntemlerinin aynı giriş görüntüsü sonucunda elde ettikleri çıktılar Şekil 6.2’de verilmiştir. Görsel çıktıların karşılaştırılması sonucu derin öğrenme tabanlı yaklaşımlardan olan RAFT modelinin, Farneback gibi klasik yöntemlere göre daha başarılı sonuçlar elde ettiği gözlemlenmiştir.



Şekil 6.2. Klasik ve derin öğrenme tabanlı yaklaşımların aynı giriş görüntülerine göre çıktıları

Optik akış ve derin öğrenme yöntemlerinin kullanılması ile elde edilen mimarilerin isimleri ve Sintel veri setine göre performansları Tablo 6.2’ de verilmiştir [21]. Çalışmada hareket tespitinin temelini oluşturan Optik akış yöntemi derin öğrenme ile kullanılarak daha iyi sonuçlar elde edilmesi ile mevcutta kullanılan uygulamaların performanslarının iyileştirmesi için önemli katkı sağlanacağı düşünülmektedir. Tablo 6.2’de gösterildiği gibi hareket tespit için denetimli öğrenme yöntemlerinden RAFT mimarisi diğer bütün yöntemlere göre daha başarılı sonuçlar elde etmiştir.

FlowNet2 modelinin Sintel veri kümesi ile eğitilmesi sonucunda donanım yetersizliğinden dolayı düşük döngü sayıları uygulanmıştır. Bundan dolayı çalışmada belirtilen değerlere ulaşılmamıştır. FlowNet2 modelinin belirtilen hiper parametre değerleri ile eğitilmesi başarımı karşılaştırmada önemli olacağı düşünülmektedir.

Tablo 6.2. Sintel veri seti üzerinde elde edilen mimari ve başarımları [21]

Metotlar	Tüm karelerde uç nokta hatası	Metotlar 2	Tüm karelerde uç nokta hatası2
RAFT	2,855	DCFlow	5,119
ScopeFlow	4,098	LiteFlowNet	5,381
MaskFlowNet	4,172	FlowNetC-MD	5,444
SelfFlow	6,571	EpicFlow	6,285

Tablo 6.2. (Devamı)

Metotlar	Tüm karelerde uç nokta hatası	Metotlar 2	Tüm karelerde uç nokta hatası2
PWC-Net+	4,596	Flownet2	6,016
LiteFlowNet2	4,686	DeepFlow	7,212
ProFlow	5,017	SPyNet	8,431
PWC-Net	5,042	UnFlow	10,219

Tablo 6.2’ de Sintel veri seti üzerinde optik akış ve derin öğrenme yöntemlerinin kullanılması ile elde edilen mimarilerin isimleri ve EPE performansları verilmiştir. Son kıyaslamalara bakıldığında çalışmanın konularından birini oluşturan RAFT mimarisi diğer yöntemlere göre daha başarılı sonuçlar elde etmiştir. Ayrıca denetimsiz öğrenme yöntemlerine göre denetimli öğrenme yaklaşımları daha yüksek başarımlar elde ettiği Tablo 6.2’de verilmiştir.

7. SONUÇLAR

Hareket tespiti, hareketli veya sabit bir dijital görüntüleyiciden elde edilen video verisi üzerinde takibi istenilen objenin hareket yönünün izlenmesi olarak tanımlanmaktadır. Hareket tespiti, güvenlik, trafik, uzay bilimi, sağlık ve askeri alanlar gibi çeşitli alanlarda sıklıkla kullanılmaktadır. Derin öğrenme tabanlı optik akış ile hareket tespiti, hareket tespitinde yaşanan zorlukların üstesinden gelme ve yüksek başarı kabiliyetlerinden dolayı günümüzde yaygın olarak çalışılmaktadır.

Optik akış, videoda art arda gelen görüntüdeki piksellerin yer değiştirmesi olarak tanımlanmaktadır. Derin öğrenme ile optik akış üzerine yapılan çalışmalar etkileyici sonuçlar vermektedir. Derin öğrenme ile optik akışı hesaplama, özellikle elde edilmesi zor olan büyük miktarlarda eğitim verisi gerektirir. Bundan dolayı sentetik veri kümeleri oluşturulmuş ve eğitimler bu veriler üzerine yapılmıştır. FlowNet, FlowNet2, SPyNet, PWC-Net, RAFT ve daha pek çoğu farklı testlerde birbirlerinden daha iyi performans gösteren yöntemler geliştirilmiş ve bu yöntemler optik akış hesaplama problemini derin öğrenme ile çözmüşlerdir.

Yapılan çalışmada derin öğrenme tabanlı optik akış ile hareket tespit yöntemleri, farklı veri setleri üzerine kronolojik sıralı şekilde uygulanmıştır. Veri setleri ile eğitilen modeller, hareket bulunan görüntüler üzerine uygulanmış ve aralarında görsel karşılaştırmalar yapılmıştır. KITTI ve Sintel veri setleri ile ağlar eğitilmiş ve bunun sonucunda eğitilen ağlar çekilen hareketli görüntülere uygulanmıştır. Uygulama çıktıları grafiksel ve görsel olarak karşılaştırılmıştır. Klasik yöntemlerden LK, HS ve Farneback yöntemleri ile derin öğrenme tabanlı yaklaşımlardan FlowNetS, FlowNetC, FlowNet2 ve RAFT modellerine aynı giriş değerlerine göre çıktılar elde edilmiştir. Aynı görüntü çıktılarına göre kıyaslamalarda bulunulmuştur. KITTI, Sintel veri setleri (görüntü ve karşılık gelen etiketli optik akış) kullanılarak farklı hiper parametreleri ile farklı görüntülerde eğitilmiş modeller üzerinde optik akış hesaplaması yapılmıştır. Bunun yanında FlowNetC ve FlowNetS modellerinin çıktılarının çok başarılı olmamasına karşın ürettikleri uçan sandalyeler veri seti birçok çalışmaya ışık tutmuş ve FlowNet'in optik akış hesaplamalarında derin öğrenmenin kullanılabileceğini ispatlamıştır.

FlowNet, FlowNet2 ve RAFT modelinin aynı giriş görüntüsüne karşın çıktıların karşılaştırılması sonucu derin öğrenme tabanlı modelinin Farneback gibi klasik yöntemlere göre daha başarılı sonuçlar elde ettiği gözlemlenmiştir.

Çalışmada güncel derin öğrenme yöntemler farklı veri setleri üzerine eğitilerek EPE ve Loss değerleri elde edilmiştir. Elde edilen veriler çerçevesinde FlowNet2 yönteminin FlowNet yöntemine göre daha başarılı sonuçlar elde ettiği görülmüştür. FlowNet2, FlowNet'ten yavaş olmasına rağmen tahmin hatasını %50'den fazla azaltmışlardır. Öte yandan görsel karşılaştırma ve

EPE ve Loss verilere göre RAFT yöntemi, FlowNet2 yöntemine göre daha performanslı (zaman ve iterasyon) bir şekilde sonuca ulaşmıştır.

Optik akış tahmini için mevcut CNN tabanlı yöntemlerin çoğu, sentetik veri kümeleri üzerinde optik akışı öğrenmeye odaklanır. Sentetik veri kümelerinin doğru sonuçlar üretmesi konusunda tartışmalar bulunmaktadır. Gerçek dünya verileri ile ağı öğrenmek daha doğru olduğunu düşünen araştırmacılar denetimsiz öğrenme yaklaşımına yönelmişlerdir. Denetimsiz bir şekilde optik akışı öğrenmek, mat olmayan nesneler, tıkanmalar, büyük hareketler ve aydınlatma değişikliklerinden kaynaklanan belirsizlikler nedeniyle daha karmaşıktır. Etiketli gerçek dünya eğitim verilerinin eksikliğiyle başa çıkmak için, araştırmacılar denetimsiz öğrenmeye dayalı optik akış hesaplama yöntemleri önermişlerdir. Etiketli eğitim verileri olan veri kümeleri üzerindeki denetimli yaklaşımlardan daha düşük başarılarla sahip olsa da etiketlenmemiş veriler üzerinde CNN modellerini (önceden) eğitmek için mevcut denetimsiz yöntemler geliştirilmeye devam edilmektedir.

Denetimli ve denetimsiz öğrenme yöntemlerinin incelenmesi sonucunda denetimli yöntemlerin hala daha başarılı sonuçlar elde ettiği gözlemlenmiştir. Denetimli öğrenme tabanlı yaklaşımların geliştirilmesi ile büyük veri setlerine olan ihtiyacın kalmaması ağı eğitilmesi açısından zamansal ayrıcalık sağlayacaktır.

Çalışmada optik akış tahmininde klasik yöntemlere göre derin öğrenme yöntemlerinin elde edilen sonuçlar bakımından daha başarılı olduğu gözlemlenmiştir. Bölüm 5'te detaylı çıktılar görülmektedir. Optik akış yöntemleri ve derin öğrenme yöntemlerine, güncel çalışmaları ekleyerek mevcut literatür çalışmaları genişletmekte ve en yüksek başarıyı sergileyen çalışmalar ön plana çıkarılmaktadır. Araştırma sonucunda hareket tespiti için temelini oluşturan Optik akış yöntemi, derin öğrenme ile kullanılarak daha iyi sonuçlar elde edileceği öngörülmektedir. Nitekim çalışmada gösterilen derin öğrenme tabanlı yaklaşımlar literatürde daha önce kullanılan LK, HS, Farneback, DeepFlow gibi klasik tabanlı uygulamalardan daha etkili sonuçlar elde edilmiştir. Klasik yöntemlerin uygulandığı alanlara yeni yaklaşımların uygulanmasının önemli katkı sağlanacağı düşünülmektedir. Derin öğrenme tabanlı optik akış ile hareket tespiti yapılan ve başarılı sonuçlar elde edilen çalışmaların karşılaştırılması ve karşılaştırmalardan elde edilen çıktıların farklı görsellere uygulanarak performans kriterleri açısından değerlendirilmesi ile literatüre katkıda bulunulduğu düşünülmektedir. Ayrıca çalışma sonucunda elde edilen verilerin bu alanda daha sonra yapacak araştırmacılara ışık tutacağı öngörülmektedir.

KAYNAKLAR

- [1] Hye Ye D., Li J., Wachs J., Bouman C. (2018). *Deep Learning for Moving Object Detection and Tracking from a Single Camera in Unmanned Aerial Vehicles (UAVs)*, Society for Imaging Science and Technology, 466-1-466-6(6). doi:10.2352/ISSN.2470-1173.2018.10.IMAWM-466
- [2] Balaji S.R., Karthikeyan S. (2017). A survey on moving object tracking using image processing, *2017 11th International Conference on Intelligent Systems and Control (ISCO)*, Coimbatore, Tamilnadu, India, 469-474.
- [3] Maraş B., Arıca N., Baytan Ertüzün A. (2017). Görsel Hedef Takibi Yöntemlerine Genel Bakış, *EMO Bilimsel Dergi*, C. 7, Sayı 13, 5-16.
- [4] Fan L., Wang Z., Cail B., Tao C. (2016). A survey on multiple object tracking algorithm, Information and Automation (ICIA), *2016 IEEE International Conference on, IEEE*, Ningbo, China
- [5] Hinton G., Deng L., Yu D., Dahl G. (2012). Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups, *IEEE Signal Processing Magazine*, C. 29 Sayı 6, 82- 97.
- [6] Beauchemin, S. S., Barron J. L., (1995). The computation of optical flow, *ACM Computing Surveys*, C.27 Sayı 3, 433-466.
- [7] Farneback, G. (2003). Two-Frame Motion Estimation Based on Polynomial Expansion. *Lecture Notes in Computer Science*, C. 2749, 363–370. doi:10.1007/3-540-45103-X_50
- [8] Horn, B. K. P. ve Schunck, B. G. (1981). Determining optical flow, *Artificial Intelligence*, C. 17, 185–203, doi: 10.1016/0004-3702(81)90024-2
- [9] Lucas, B. D., & Kanade, T. (1981). Iterative Image Registration Technique With An Application To Stereo Vision. C. 2, 674–679
- [10] Weinzaepfel P., Revaud J., Harchaoui Z., Schmid C. (2013). DeepFlow: Large displacement optical flow with deep matching, *Proceedings of the IEEE International Conference on Computer Vision*, 1385-1392, doi: 10.1109/ICCV.2013.175
- [11] Revaud, J., Weinzaepfel, P., Harchaoui, Z., & Schmid, C. (2015). EpicFlow: Edge-preserving interpolation of correspondences for optical flow, *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, C. 07 Sayı 12, 1164–1172. doi:10.1109/CVPR.2015.7298720
- [12] Dosovitskiy, A., Fischery, P., Ilg, E., Hausser, P., Hazirbas, C., Golkov, V., ... Brox, T. (2015). FlowNet: Learning optical flow with convolutional networks, *Proceedings of the IEEE International Conference on Computer Vision*, C. 2015 International Conference on Computer Vision, 2758–2766. doi:10.1109/ICCV.2015.316
- [13] Ilg, E., Mayer N., Saikia T., Keuper M, Dosovitskiy A., Brox T. (2017) FlowNet 2.0: Evolution of Optical Flow Estimation with Deep Networks, *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, University of Freiburg, Germany
- [14] Sun, D., Yang, X., Liu, M. Y., Kautz, J. (2018). PWC-Net: CNNs for Optical Flow Using Pyramid, Warping, and Cost Volume, *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA
- [15] Zahneisen, B., Baeumler, K., Zaharchuk, G., Fleischmann, D., Zeineh, M. (2020). Deep flow-net for EPI distortion estimation. *NeuroImage*, C. 217. doi.org/10.1016/j.neuroimage.2020.116886
- [16] Teed, Z. ve Deng, J. (2020). RAFT: Recurrent All-Pairs Field Transforms for Optical Flow, *Lecture Notes in Computer Science*, 402–419. doi: 10.1007/978-3-030-58536-5_24.
- [17] Ranjan, A., & Black, M. J. (2017). Optical flow estimation using a spatial pyramid network, *30th IEEE Conference on Computer Vision and Pattern Recognition*, C. 2017-January, 2720–2729. doi.org/10.1109/CVPR.2017.291
- [18] Introduction to Motion Estimation with Optical Flow, <https://nanonets.com/blog/optical-flow> Nano Net Technologies Inc, Erişim: 01 Nisan 2021.

- [19] Wang, W., Shen, J., & Shao, L. (2018). Video Salient Object Detection via Fully Convolutional Networks, *IEEE Transactions on Image Processing*, C. 27(1), 38–49. doi.org/10.1109/TIP.2017.2754941
- [20] Zhao, S., Sheng, Y., Dong, Y., Chang, E. I. C., & Xu, Y. (2020). Maskflownet: Asymmetric feature matching with learnable occlusion mask, *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 6277–6286. doi.org/10.1109/CVPR42600.2020.00631
- [21] MPI Sintel Flow Dataset, <http://sintel.is.tue.mpg.de/results>, Erişim: 01 Nisan 2021.
- [22] Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* C. 9351, 234–241. doi:10.1007/978-3-319-24574-4_28
- [23] Janai, J., Güney, F., Ranjan, A., Black, M., & Geiger, A. (2018). Unsupervised Learning of Multi-Frame Optical Flow with Occlusions, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* C. 11220 LNCS, 713–731. doi.org/10.1007/978-3-030-01270-0_42
- [24] Liu, P., King, I., Lyu, M. R., & Xu, J. (2019). DDFlow: Learning optical flow with unlabeled data distillation, *ArXiv*, arXiv.
- [25] Liu, P., Lyu, M., King, I., & Xu, J. (2019). Selfflow: Self-supervised learning of optical flow, *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, C. 2019-June, 4566–4575. IEEE Computer Society. doi.org/10.1109/CVPR.2019.00470
- [26] Maurer, D., & Bruhn, A. (2019). Proflow: Learning to predict optical flow. *British Machine Vision Conference 2018, BMVC 2018*, BMVA Press.
- [27] Memisevic, R., & Hinton, G. E. (2010). Learning to represent spatial transformations with factored higher-Order boltzmann machines, *Neural Computation*, C. 22 Sayı. 6, 1473–1492. doi:10.1162/neco.2010.01-09-953
- [28] Tiwari, M., & Singhai, R. (2017). A Review of Detection and Tracking of Object from Image and Video Sequences. *International Journal of Computational Intelligence Research*, C.13 Sayı 5. 745–765.
- [29] Simoncelli, E. P., Adelson, E. H., & Heeger, D. J. (1991). Probability distributions of optical flow, *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Tamilnadu, India
- [30] Roth, S., & Black, M. J. (2007). On the spatial statistics of optical flow. *International Journal of Computer Vision* Sayı. 74, 33–50. doi:10.1007/s11263-006-0016-x
- [31] D. Sun, S. Roth, J. P. Lewis, and M. J. Black, (2008). Learning optical flow, *European Conference on Computer Vision (ECCV)*.
- [32] Wulff, J., & Black, M. J. (2015). Efficient sparse-to-dense optical flow estimation using a learned basis and layers. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Sayı. 07-12-June-2015, 120–130. doi:10.1109/CVPR.2015.7298607
- [33] A project of Karlsruhe Institute of Technology and Toyota Technological Institute at Chicago, http://www.cvlibs.net/datasets/kitti/eval_flow.php, Erişim: 01 Nisan 2020.
- [34] Xu, J., Ranftl, R., & Koltun, V. (2017). Accurate optical flow via direct cost volume processing, *30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017* Sayı. 2017-January, 5807–5815. doi:10.1109/CVPR.2017.615
- [35] Zhou, S., Chen, Q., & Wang, X. (2013). Active deep learning method for semi-supervised sentiment classification. *Neurocomputing*, Sayı. 120, 536–546. doi:10.1016/j.neucom.2013.04.017
- [36] LeCun, Y., Bengio, Y., Hinton, G. (2015). Deep learning, *Nature*, Sayı. 521, 436–444. doi: 10.1038/nature14539.
- [37] E. Özüntürk (2007), *Optik Akış İle Hareket Tespiti*, Doktora Tezi, Yıldız Teknik Üniversitesi, Fen Bilimleri Enstitüsü.

- [38] Güney, F., & Geiger, A. (2017). Deep discrete flow. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Sayı. 10114, 207–224. doi:10.1007/978-3-319-54190-7_13
- [39] Hui, T. W., Tang, X., & Loy, C. (2019). A lightweight optical flow CNN – *Revisiting data fidelity and regularization*, ArXiv, arXiv. doi:0.1109/tpami.2020.2976928
- [40] Piergiovanni, A. J., & Ryoo, M. S. (2019). Representation flow for action recognition, *In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* Sayı.. 2019-June, 9937–9945. doi:10.1109/CVPR.2019.01018
- [41] Bailer, C., Varanasi, K., & Stricker, D. (2017). CNN-based patch matching for optical flow with thresholded hinge embedding loss, *In Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017* Sayı 2017-January, 2710–2719. doi:10.1109/CVPR.2017.290
- [42] Lai, W. S., Huang, J. B., & Yang, M. H. (2017). Semi-supervised learning for optical flow with generative adversarial networks, *In Advances in Neural Information Processing Systems*, Sayı 2017-December, 354–364.
- [43] Meister, S., Hur, J., & Roth, S. (2018). UnFlow: Unsupervised learning of optical flow with a bidirectional census loss, *In 32nd AAAI Conference on Artificial Intelligence*, Sayı AAAI 2018, 7251–7259.
- [44] Guan, S., Li, H., & Zheng, W. S. (2019). Unsupervised learning for optical flow estimation using pyramid convolution LSTM, *IEEE International Conference on Multimedia and Expo* Sayı 2019-July, 181–186. doi:10.1109/ICME.2019.00039
- [45] Memisevic, R., & Hinton, G. (2007). Unsupervised learning of image transformations, *In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*.
- [46] Alletto, S., Abati, D., Calderara, S., Cucchiara, R., & Rigazio, L. (2017). TransFlow: Unsupervised motion flow by joint geometric and pixel-level estimation, *ArXiv*. arXiv.
- [47] Long, G., Kneip, L., Alvarez, J. M., Li, H., Zhang, X., & Yu, Q. (2016). Learning image matching by simply watching video. *In Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* Sayı 9910 LNCS, 434–450. doi:10.1007/978-3-319-46466-4_26
- [48] Kong, L., & Yang, J. (2020). FdflowNet: Fast Optical Flow Estimation Using A Deep Lightweight Network, *International Conference on Image Processing, ICIP*, Sayı 2020-October, 1501–1505. doi:10.1109/ICIP40778.2020.9191101
- [49] Ertam, F. (2017). Data classification with deep learning using tensorflow, *In 2nd International Conference on Computer Science and Engineering*, Sayı. UBMK 2017, 755–758. doi:10.1109/UBMK.2017.8093521
- [50] Aydın B. (2018). *Derin Öğrenme Ve Büyük Veri Yaklaşımları İle Metin Analizi*, Doktora Tezi, Fırat Üniversitesi, Fen Bilimleri Enstitüsü.
- [51] Cireşan, D., Meier, U., Masci, J., & Schmidhuber, J. (2012). Multi-column deep neural network for traffic sign classification. *Neural Networks*, Sayı 32, 333–338. doi:10.1016/j.neunet.2012.02.023
- [52] Ning, F., Delhomme, D., LeCun, Y., Piano, F., Bottou, L., & Barbano, P. E. (2005). Toward automatic phenotyping of developing embryos from videos, *IEEE Transactions on Image Processing*, C. 14(9), 1360–1371. doi:10.1109/TIP.2005.852470
- [53] Tompson, J., Goroshin, R., Jain, A., LeCun, Y., & Bregler, C. (2015). Efficient object localization using Convolutional Networks, *In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Sayı 07-12-June-2015, 648–656. doi:10.1109/CVPR.2015.7298664
- [54] Sermanet, P., Kavukcuoglu, K., Chintala, S., & Lecun, Y. (2013). Pedestrian detection with unsupervised multi-stage feature learning, *In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* 3626–3633. doi:10.1109/CVPR.2013.465
- [55] Figen Ö. (2018). *Derin Öğrenme İle Yüksek Çözünürlüklü Hava Görüntülerinde Yolların Tespit*

Edilmesi, Yüksek Lisans Tezi, Fırat Üniversitesi, Fen Bilimleri Enstitüsü.

- [56] Rosenblatt, F.(1957). The Perceptron. A Perceiving and Recognizing Automaton (Project PARA), CAL Rept. lio. 05-460-1, Jan.
- [57] Makine Öğrenmesinde Temel Kavramlar, <https://koglakk.medium.com/maki%CC%87ne-%C3%B6%C4%9Frenmesi%CC%87nde-temel-kavramlar-ab75c6e5e7af> Medium, Erişim: 01 Nisan 2021.
- [58] Makine Öğrenmesi (Supervised vs Unsupervised Learning), <https://medium.com/algorithms-data-structures/makine-%C3%B6%C4%9Frenmesi-supervised-vs-unsupervised-learning-299b8c639a01> Medium, Erişim: 01 Nisan 2021.
- [59] Derin Öğrenme Uygulamalarında En Sık kullanılan Hiper-Parametreler, <https://medium.com/deep-learning-turkiye/derin-ogrenme-uygulamalarinda-en-sik-kullanilan-hiper-parametreler-ece8e9125c4> Medium, Erişim: 01 Nisan 2021.
- [60] Lecun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning, *Nature Publishing Group*, doi:10.1038/nature14539
- [61] Regularization in Deep Learning, <https://chatbotlife.com/regularization-in-deep-learning-f649a45d6e0> Chatbotlife, Erişim: 01 Nisan 2021.
- [62] Model Tuning, <http://www.r2d3.us/visual-intro-to-machine-learning-part-2>, Erişim: 01 Nisan 2021.
- [63] Akhir T, (2018). Implementasi Convolutional Neural Network Transportasi Tradisional Menggunakan Keras, *Universitas Islam Indonesia*
- [64] LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). *Gradient-based learning applied to document recognition. Proceedings of the IEEE*, C. 86, Sayı 11. 2278–2323. doi:10.1109/5.726791
- [65] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). ImageNet classification with deep convolutional neural networks, *Communications of the ACM*, C.60, Sayı 6, 84–90. doi:10.1145/3065386
- [66] Large Scale Visual Recognition Challenge, www.image-net.org, Erişim: 01 Nisan 2021.
- [67] Kim, J. H., Seo, S. Y., Song, C. G., & Kim, K. S. (2019). Assessment of Electrocardiogram Rhythms by GoogLeNet Deep Neural Network Architecture, *Journal of Healthcare Engineering*, doi:10.1155/2019/2826901
- [68] Zhang, X., Zou, J., He, K., Sun, J. (2015). Accelerating Very Deep Convolutional Networks for Classification and Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(10), doi: 10.1109/TPAMI.2015.2502579
- [69] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition, *In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Sayı 2016-December, 770–778. doi:10.1109/CVPR.2016.90
- [70] Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks, *30th IEEE Conference on Computer Vision and Pattern Recognition*, Sayı. 2017-January, 2261–2269. doi:10.1109/CVPR.2017.243
- [71] Kızrak, M. A., & Bolat, B. (2018). Derin Öğrenme ile Kalabalık Analizi Üzerine Detaylı Bir Araştırma. *Bilişim Teknolojileri Dergisi*, 263–286. doi:10.17671/gazibtd.419205
- [72] Yapay Zekanın Yeni ve Çekici Mimarisi Kapsül Ağına Uygulamalı Bir Bakış, <https://ayyucekizrak.medium.com/yapay-zekan%C4%B1n-yeni-ve-%C3%A7ekici-mimarisi-kaps%C3%BCl-a%C4%9F%C4%B1na-uygulamal%C4%B1-bir-bak%C4%B1%C5%9F-ef7310e3d847> Medium, Erişim: 01 Nisan 2021.
- [73] Cutting, J. E. (2000). Images, imagination, and movement: Pictorial representations and their development in the work of James Gibson. *Perception*, C.29, Sayı 6, 635–648. doi: 10.1068/p2976
- [74] Keçeli, A. S., & Kaya, A. (2018). Optik Akış Görüntüsü ve Bi-Lstm ile Şiddet İçeren Hareketlerin Sınıflandırılması. *European Journal of Science and Technology*, 204–208. doi:10.31590/ejosat.460257

- [75] Buxton, B., & Buxton, H. (1984). Computation of optic flow from the motion of edge features in image sequences, *Image and Vision Computing*, C. 2, Sayı 2, 59–75. doi:10.1016/0262-8856(84)90001-5
- [76] Wedel, A., Pock, T., Zach, C., Bischof, H., & Cremers, D. (2009). An improved algorithm for TV-L1 optical flow, *In Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Sayı 5604 LNCS, 23–45. doi:10.1007/978-3-642-03061-1_2
- [77] Bansal, M., Kumar, M. (2021). An efficient technique for object recognition using Shi-Tomasi corner detection algorithm, *Soft Computing*, C.25 Sayı 6, 4423–4432. doi:10.1007/s00500-020-05453-y
- [78] Middlebury Stereo Datasets, <https://vision.middlebury.edu/stereo/data>, Erişim: 01 Nisan 2021.
- [79] Yosemite NP, <https://data.world/nps/yosemite-np>, Erişim: 01 Nisan 2021.



ÖZGEÇMİŞ

Ammar ASLAN

