



T.C.
KONYA TEKNİK ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ



OTONOM SİSTEMLER İÇİN SENSÖR
FÜZYON VE GÖRSEL TABANLI
KONUMLANDIRMA

Abdullah YUSEFİ

DOKTORA TEZİ

Bilgisayar Mühendisliği Anabilim Dalı

Şubat-2024
KONYA
Her Hakkı Saklıdır

TEZ KABUL VE ONAYI

Abdullah Yusefi tarafından hazırlanan “**Otonom Sistemler için Sensör Füzyon ve Görsel Tabanlı Konumlandırma**” adlı tez çalışması 07/02/2024 tarihinde aşağıdaki jüri tarafından oy birliği ile Konya Teknik Üniversitesi Lisansüstü Eğitim Enstitüsü **Bilgisayar Mühendisliği** Anabilim Dalı’nda **DOKTORA TEZİ** olarak kabul edilmiştir.

Jüri Üyeleri

İmza

Başkan

Prof. Dr. Adem Alpaslan ALTUN

.....

Danışman

Prof. Dr. Cemil SUNGUR

.....

2. Danışman

Doç. Dr. Akif DURDU

.....

Üye

Prof. Dr. Fatih BAŞÇİFTÇİ

.....

Üye

Doç. Dr. Ahmet BABALIK

.....

Üye

Doç. Dr. Ersin KAYA

.....

Yukarıdaki sonucu onaylarım.

Prof. Dr. Mevlüt UYAN
Enstitü Müdürü

TEZ BİLDİRİMİ

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

DECLARATION PAGE

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

Abdullah YUSEFI

Tarih: 07/02/2024

ÖZET

DOKTORA TEZİ

OTONOM SİSTEMLER İÇİN SENSÖR FÜZYON VE GÖRSEL TABANLI KONUMLANDIRMA

Abdullah YUSEFİ

**Konya Teknik Üniversitesi
Lisansüstü Eğitim Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı**

Danışman: Prof. Dr. Cemil SUNGUR

2024, 119 Sayfa

Jüri

**Prof. Dr. Cemil SUNGUR
Doç. Dr. Akif DURDU
Prof. Dr. Adem Alpaslan ALTUN
Prof. Dr. Fatih BAŞÇİFTÇİ
Doç. Dr. Ahmet BABALIK
Doç. Dr. Ersin KAYA**

İnsanlar gibi kendi kararlarını veren ve görevlerini gerçekleştiren otonom mobil robot uygulamalarının ihtiyaç duyduğu bir özellik, insansı görevleri robotlara yaptırma gerekliliğidir. Otonom bir robot, bulunduğu ortamın geometrik yapısını anlamalı, kendini doğru bir şekilde konumlandırmalı ve bu bilgileri kullanarak belirlenen görev noktasına yönlendiren bir hareket yörüngesi oluşturmalıdır. Özellikle tek bir sensörün yetersiz olduğu durumlarda, farklı sensörlerin birleştirilmesiyle elde edilen sensör füzyonu, mobil robot konumlandırma çalışmalarında önemli bir rol oynamaktadır. Son yıllarda, işlemci hızındaki gelişmeler sayesinde, düşük maliyetli monoküler kameralar kullanılarak gerçekleştirilen Görsel Odometri (Visual Odometry (VO)) yöntemlerine daha fazla odaklanılmıştır. Ayrıca, kameralara ek olarak düşük maliyetli Atalet Ölçü Birimi (Inertial Measurement Unit (IMU)) sensörlerini içeren VIO çözümleri, konumlandırmaya katkı sağlamak amacıyla tercih edilmeye başlanmıştır. Geleneksel geometrik tabanlı çözümler genellikle karmaşık dünyayı doğru bir şekilde temsil etmekte zorlanır ve güvenilir sonuçlar elde etmekte zorluk yaşar. Bu nedenle, günümüzde Yapay Zeka tabanlı çözümler, farklı ortamlara ve kolay uyarlanabilme avantajları nedeniyle geleneksel çözümlerin yerini almaktadır.

Bu tez çalışması, yukarıda belirtilen bilgiler ışığında, tek bir sensörün yetersiz olduğu durumlarda otonom araç geliştirmek için iki farklı uygulama önermektedir. İlk uygulama, iç ortamda hareket eden bir İHA'nın konumunu tahmin etmek amacıyla görsel ve IMU bilgilerine dayalı derin öğrenme tabanlı bir hibrit mimari sunmaktadır. İkinci uygulama ise ardışık kamera görüntülerini işleyerek otonom aracın konumunu başarılı bir şekilde tahmin eden yapay zeka tabanlı farklı bir VIO uygulamasını, farklı bir füzyon tekniğiyle gerçekleştirmektedir. Her iki uygulama da otonom aracın konumlandırması için yenilikçi yöntemler sunmaktadır. Bu yöntemler, önceki çalışmalara kıyasla üstün performans sergileme eğilimindedir. Ayrıca, gerçekleştirilen uygulamaların gerçek zamanlı sistemlerde çalışabilecek nitelikte olduğu belirlenmiştir.

Anahtar Kelimeler: Derin Öğrenme, Konumlandırma, Otonom Araç, Sensör Füzyonu, VIO.

ABSTRACT

PhD THESIS

SENSOR FUSION AND VISUAL-BASED LOCALIZATION FOR AUTONOMOUS SYSTEMS

Abdullah YUSEFI

**Konya Technical University
Institute of Graduate Studies
Department of Computer Engineering**

Advisor: Prof. Dr. Cemil SUNGUR

2024, 119 Pages

Jury

Prof. Dr. Cemil SUNGUR

Assoc. Dr. Akif DURDU

Prof. Dr. Adem Alpaslan ALTUN

Prof. Dr. Fatih BAŞÇİFTÇİ

Assoc. Dr. Ahmet BABALIK

Assoc. Dr. Ersin KAYA

One of the essential features for autonomous mobile robot applications that make their own decisions and perform tasks like humans is the necessity of delegating humanoid tasks to robots. An autonomous robot should understand the geometric structure of its environment, accurately localize itself, and create a motion trajectory guiding it to the specified point using this information. Sensor fusion obtained by combining different sensors plays a significant role in mobile robot localization, especially in cases where a single sensor is insufficient. In recent years, advancements in processor speed have led to increased focus on Visual Odometry (VO) methods using low-cost monocular cameras. Additionally, Visual-Inertial Odometry (VIO) solutions, incorporating low-cost Inertial Measurement Unit (IMU) sensors alongside cameras, have been preferred to contribute to localization. Traditional geometric-based solutions often struggle to accurately represent complex environments and face difficulties in obtaining reliable results. Therefore, Artificial Intelligence-based solutions are increasingly replacing traditional methods due to their adaptability to different environments.

In light of the information mentioned above, this thesis proposes two different applications for developing autonomous vehicles in cases where a single sensor is insufficient. The first application presents a deep learning-based hybrid architecture using visual and IMU information to predict the position of a UAV (Unmanned Aerial Vehicle) moving indoors. The second application implements an artificial intelligence-based VIO (Visual-Inertial Odometry) application successfully predicting the position of an autonomous vehicle by processing consecutive camera images, using a different fusion technique. Both applications offer innovative methods for the localization of autonomous vehicles, demonstrating a tendency to outperform previous studies. Furthermore, it has been determined that the implemented applications are capable of operating in real-time systems.

Keywords: Autonomous Vehicle, Deep Learning, Localization, Sensor Fusion, VIO.

ÖNSÖZ

Bu tez çalışması, benim için hem akademik hem de kişisel bir gelişim sürecinin yansımasıdır. Bu süreçte desteklerini esirgemeyen ve katkıda bulunan herkese teşekkür etmek istiyorum.

İlk olarak, rehberliği ve bilgeliği ile her daim yanımda olan 1. Danışmanım Prof. Dr. Cemil Sungur'a ve 2. Danışmanım Doç. Dr. Akif DURDU'ya içten teşekkürlerimi sunarım. RACLAB ailesine de bu yolculuktaki birlikte geçirdiğimiz zaman için müteşekkirim.

Beni yetiştiren, destek olan ve her zaman yanımda olan aileme, özellikle anneme, babama ve diğer aile bireylerime teşekkürlerimi iletmek istiyorum.

Ayrıca, Türkiye Bursları'na (YTB) katkılarından dolayı teşekkür etmeyi de unutmamak önemlidir. Bu burs programı, çalışmalarımı sürdürebilmemde sağladığı destekle büyük bir etkisi oldu. Bu vesileyle, Türkiye Bursları'na teşekkürlerimi sunarım.

Son olarak, evliliğimi renklendiren, beni hayata daha çok bağlayan oğlum Hamza Taha'ya, ve bana her daim destek olan, başarılarımın temelini atan en büyük etken, hayatımı süsleyen ve sevgiyle anılacak sevgili eşim Nafisa'ya, teşekkürlerimi sunarım.

Bu tez, sizin değerli katkılarınızla daha anlam kazandı. Hepinize teşekkür ederim.

Abdullah YUSEFI
KONYA-2024

İÇİNDEKİLER

ÖZET	iv
ABSTRACT.....	v
ÖNSÖZ	vi
İÇİNDEKİLER	vii
ÇİZELGELER LİSTESİ	ix
ŞEKİLLER LİSTESİ	x
SİMGELER VE KISALTMALAR	xii
1. GİRİŞ.....	1
1.3. Araştırma Soruları ve Hipotezleri.....	3
1.4. Tez Yapısı	4
2. LİTERATÜR ARAŞTIRMASI	6
3. METODOLOJİ.....	14
3.1 Otonom Araç Nedir?.....	14
3.2. Konum ve Konum Tahmini (Konumlandırma)	15
3.2.1 Görsel Tabanlı Konumlandırma	17
3.2.2. Sensör Füzyon Tabanlı Konumlandırma	18
3.2.3. Derin Öğrenme Tabanlı Konumlandırma	18
3.3. Derin Öğrenme Teknikleri.....	19
3.3.1. Çok Katmanlı Algılayıcı (Multi Layer Perceptron (MLP)).....	19
3.3.2. Evrimsel Sinir Ağları	20
3.3.3. Doğrultulmuş Doğrusal Birim (Rectified Linear Unit (ReLU)).....	23
3.3.4. LSTM ve Bi-LSTM (Bi-directional LSTM):.....	24
3.3.5. Siamese Ağları	28
3.3.6. Rastgele PCA (Principal Component Analysis) Artırma	29
3.3.7. Hata Fonksiyonu	30
3.4. Sensor Füzyonu ve IMU İşleme	32
3.4.1. IMU Verisi İşleme:	33
3.4.2. Çoklu Adım Düzeltme Filtresi (Multi-Step Correction Filter):.....	37
3.4.3. Kokusuz Kalman Filtresi (Unscented Kalman Filter)	39
4. OTONOM KONUMLANDIRMA UYGULAMALARI.....	43
4.1 Uygulamarda Kullanılan Veri Setleri	43
4.1.1 EuRoC Veri Seti	43
4.1.2 Simülasyon Veri Seti	43
4.1.3 RAC-LAB Veri Seti.....	44

4.1.4. KITTI Veri Seti.....	45
4.2. Uygulama 1: İHA'larda Kapalı Alan Küresel Konumlandırma İçin LSTM ve Filtre Tabanlı Karşılaştırma Analizi Uygulaması	46
4.2.1. Uygulamada Kullanılan Derin Öğrenme Mimarisi	47
4.2.2. Model Eğitimi	51
4.5. Uygulama 2: Genelleştirilebilir D-VIO ve GNSS/IMU ile Birleştirilmiş İyileştirilmiş Otonom Araç Konumlandırma Uygulaması.....	52
4.5.1. Mimari	53
5. DENEYSEL SONUÇLAR	66
5.1. Uygulama 1 Sonuçları	66
5.1.1. IMU-LSTM ve IMU-Filtre Karşılaştırması	66
5.1.2. Literatürle Karşılaştırma	73
5.1.3. Eksiltme çalışması ve sonuçların değerlendirilmesi	75
5.2. Uygulama 2 Sonuçları	76
5.2.1. GD-VIO Sonuçları	76
5.2.2. GNSS/GD-VIO/IMU Füzyonu Sonuçları.....	83
6. SONUÇ VE ÖNERİLER.....	88
6.1. Öneriler	90
KAYNAKLAR	92

ÇİZELGELER LİSTESİ

Çizelge 4.1. Görsel-CNN Özellik Çıkartıcı Yapılandırması.	48
Çizelge 4.2. IMU-LSTM Özellik Çıkartıcı Yapılandırması.	49
Çizelge 4.3. Full-LSTM Küresel Konum Regresör Yapılandırması.	50
Çizelge 4.4. Alt Ağ CNN özellik çıkartıcısı konfigürasyonu.	56
Çizelge 4.5. MLP Dahili Özellik Çıkartıcı Konfigürasyonu.	57
Çizelge 4.6. Bi-LSTM ve Konum Regresörü Konfigürasyonu.	59
Çizelge 5.1. EuRoC veri seti üzerinde IMU-LSTM tabanlı ve IMU-Filtre tabanlı pozisyon tahmini kurulumlarının karşılaştırma sonucu.	70
Çizelge 5.2. IMU-LSTM tabanlı ve IMU-Filtre tabanlı pozisyon tahmini kurulumlarının gerçek zamanlı deneme üzerindeki karşılaştırma sonuçları.	72
Çizelge 5.3. EuRoC dizileri için karşılaştırmalı sonuçlar.	74
Çizelge 5.4. Eğitilmiş sekanslarında sayısal ATE ve RPE sonuçları.	80
Çizelge 5.5. Test sekanslarında sayısal ATE ve RPE sonuçları.	82
Çizelge 5.6. Test 1'deki Sayısal ATE ve RPE sonuçları.	84
Çizelge 5.7. Test 2'deki Sayısal ATE ve RPE sonuçları.	85

ŞEKİLLER LİSTESİ

Şekil 3.1. Otonom araç genel blok şeması.	14
Şekil 3.2. Mobil Robotun 2D pozisyonu (x, y) ve rotasyonu (θ).	16
Şekil 3.3. CNN Mimarisi.	22
Şekil 3.4. Sinir Ağlarındaki Aktivasyon Fonksiyonlarının ve Gradyanların Görselleştirilmesi.	24
Şekil 3.5. Bi-LSTM Ağ Mimarisi.	27
Şekil 3.6. Siamese Ağ Mimarisi.	29
Şekil 3.7. Mahony filtresi blok şeması.	33
Şekil 3.8. Madgwick filtresi blok şeması.	35
Şekil 3.9. EKF filtresi blok şeması.	36
Şekil 3.10. UKF filtresi blok şeması.	44
Şekil 4.1. Simülasyon ortamı, iki boyutlu ızgara haritası ve Tello İHA'nın örnek kamera görüntüleri.	40
Şekil 4.2. DJI Tello İHA ve laboratuvarından elde edilen örnek bozulmuş görüntüler.	45
Şekil 4.3. KITTI veri seti örnek resimleri.	46
Şekil 4.4. Önerilen Derin RCNN Konum Tahmini Sisteminin Mimarisi.	47
Şekil 4.5. EuRoC veri seti V1_02 orta dizisinde beş farklı ağ formu kullanarak eğitim ve doğrulama doğruluğunun ve kaybın karşılaştırılması.	52
Şekil 4.6. Önerilen GNSS/D-VIO/IMU yönteminin akış şeması.	53
Şekil 4.7. Önerilen GD-VIO'nun mimarisi.	55
Şekil 5.1. Test edilen simülasyon ortamının Gerçek Yol Yörüngesi.	67
Şekil 5.2. Simülasyon ortamındaki IMU-LSTM tabanlı ve IMU-Filtre tabanlı konum tahmin kurulumunun karşılaştırma sonucu.	68
Şekil 5.3. EuRoC veri setinin V1_01 dizisi üzerinde IMU-LSTM alt modelinin sonuçları.	69
Şekil 5.4. IMU-LSTM tabanlı ve IMU-Filtre tabanlı pozisyon tahmini kurulumlarının EuRoC V1_02 dizileri üzerindeki yörünge karşılaştırması.	70

Şekil 5.5. IMU-LSTM tabanlı ve IMU-Filtre tabanlı pozisyon tahmini kurulumlarının EuRoC V1_03 dizileri üzerindeki yörünge karşılaştırması.	71
Şekil 5.6. IMU-LSTM tabanlı ve IMU-Filtre tabanlı pozisyon tahmini kurulumlarının gerçek dünya deneyimi üzerindeki yörünge karşılaştırması.	73
Şekil 5.7. Eğitim ve test kaybını Siamese Tipi CNN + Lambda katmanı ile ve olmadan karşılaştırm. Sol: Aşırı öğrenme ile hata. Sağ: İyi öğrenme hata.	78
Şekil 5.8. ORB-SLAM2, VISO2, SfMLearner, Depth-Vo-Feat ve önerilen yöntemin Eğitilmiş Sequencelerde Ölçek Hizalaması Sonrasında Yörüngeleri. ..	80
Şekil 5.9. Test Sequencelerinde ORB-SLAM2, VISO2, SfMLearner, Depth-Vo-Feat ve önerilen yöntemin Scale Alignment sonrasındaki yörüngeleri.	82
Şekil 5.10. İki kentsel ortamda 'Ref', 'UKF', 'UKF w/o' yolculukları.	84
Şekil 5.11. Multi-step düzeltme filtresi şemasıyla ve şemasız UKF tabanlı yaklaşımlar kullanılarak, gürültülü RTK-GNSS verilerinde öngörülen yörüngeler ile referans arasındaki karşılaştırma.	87
Şekil 5.12. UKF tabanlı GD-VIO/IMU füzyonu kullanılarak GNSS reddedilen senaryolarda öngörülen yörüngeler.	88

SİMGELER VE KISALTMALAR

Simgeler

t	: Zaman Faktörü
k	: Zaman Faktörü
$P(x, y, z)$	: x, y, z Ekseninde Pozisyon
R	: Rotasyon
q	: Kuaterniyon Şeklinde Rotasyon
P_{est}	: Tahmini Konum Vektörü
$P_{observed}$	: Gözlemlenen Sensör Ölçümleri
$E(P, P_{observed})$	: Hata Fonksiyonu
f_t	: Forget Kapısı Çıkışı
i_t	: Input Kapısı Çıkışı
g_t	: Giriş Kapısı Çıkışı
o_t	: Output Kapısı Çıkışı
c_t	: Hücre Durumu
σ	: Sigmoid Fonksiyonu
W_i	: Ağırlık Matrisi
b_i	: Sapma Terimi
$I_t^{\hat{a}}$	: Normalleştirilmiş İvmeölçer Ölçümleri
$v(I_t^q)$	: Kuaterniyon Cinsinden t Zamanındaki Rotasyonu
I_t^{ω}	: Güncellenmiş Jiroskop Ölçümleri
ΔI_t^W	: Jiroskoptan Gelen Rotasyon Artışı
I_t^W	: Rotasyon Artışının Entegrasyonu
e_i	: Hata Değeri
I_t^a	: İvmeölçer Ölçümleri
$I_t^{\hat{W}}$	: Kuaterniyon Rotasyonu
$W_{\hat{g}}$	: Normalize Edilmiş Yerçekimi
$p(x)$	: Gaus Dağılımlı Konum
x_k	: Durum Vektörü
z_k	: Ölçüm Vektörü
u_k	: Kontrol Girişi
P_k	: Kovaryans Matrisi

Q_k	: İşlem Gürültüsü Kovaryans Matrisi
A_k	: Durum Geçiş Matrisi
$\mathcal{N}$	: Normal Dağılımı
R_k	: Ölçüm Gürültüsü
S_k	: Yenilik Kovaryansı
K_k	: Kalman Kazancı
I_t	: Kamera Görüntüleri
IMU_t	: IMU Verileri
$P(x_t u_t, x_{t-1})$	: Hareket Modeli
$P(z_t x_t)$	: Ölçüm Modeli
θ^*	: Optimal Ağırlıklar
pc_i	: PCA Kovaryans Matrisi
b_i	: PCA Özdeğeri
a_i	: PCA Rastgele Değeri
$conca(a, b)$	: a ve b vektörlerini birleştiren fonksiyon
λ	: Rotasyonu Ayarlayan Ölçek Değeri
$y_{a,t}$	: IMU İvmeölçer Ölçümü
$y_{w,t}$	: IMU Jiroskop Ölçümü
Ω	: Çarpıklık Simetrisi
τ_a	: İvmeölçer Zaman Sabiti
τ_w	: Jiroskop Zaman Sabiti
g	: Yerçekimi İvme Vektörü
y_p	: Jeosantrik Koordinat Sistemindeki Pozisyon
n_p	: GNSS Ölçüm Gürültüsü
e	: Dünya Eksantrikliği
$\hat{x}$	: Tahmin Edilen Değer
x	: Gerçek Değer

Kısaltmalar

ATE	: Absolute Trajectory Error
Bi-LSTM	: Bidirectional LSTM
BRIEF	: Binary Robust Independent Elementary Features
BRISK	: Binary Robust Invariant Scalable Keypoints
CNN	: Convolutional Neural Network
CPU	: Central Processing Unit
DoF	: Degree of Freedom
DTAM	: Dense Tracking and Mapping
EKF	: Extended Kalman Filter
FAST	: Features from Accelerated Segment Test
FC	: Fully Connected
GB	: Giga Byte
GD-VIO	: Generalizable Deep Visual Inertial Odometry
GNSS	: Global Navigation Satellite System
GPU	: Graphical Processing Unit
İHA	: İnsansız Hava Aracı
IMM	: Interacting Multiple Model
IMU	: Inertial Measurement Unit
KF	: Kalman Filtresi
KLT	: Kanade-Lucas-Tomasi
LC	: Loop Closure
LiDAR	: Light Detecton and Ranging
LSTM	: Long Short Term Memory
MAE	: Mean Absolute Error
MAV	: Micro Aerial Vehicle
MLP	: Multi Layer Perceptron
MSE	: Mean Squared Error
OKPS	: Optimized Kalman Particle Swar
OKVIS	: Open Keyframe-based Visual-Inertial SLAM
ORB	: Oriented FAST and Rotated BRIEF
ORB-SLAM	: Oriented FAST and Rotated BRIEF-SLAM
PCA	: Principal Component Analysis

PPK	: Post-Processing Kinematic
RAM	: Random Access Memory
RCNN	: Recurrent Convolutional Neural Network
ReLU	: Rectified Linear Unit
RGB	: Red-Green-Blue
RGB-D	: Red-Green-Blue-Depth
RMSE	: Root Mean Squared Error
RNN	: Recurrent Neural Network
ROS	: Robotic Operating System
ROVIO	: Robust Visual Inertial Odometry
RPE	: Relative Position Error
SFM	: Structure From Motion
SGD	: Stochastic Gradient Descent
SGDM	: Gradient Descent with Momentum
SIFT	: Scale-Invariant Feature Transform
SLAM	: Simultaneous Localization And Mapping
SURF	: Speeded Up Robust Features
SVO	: Semi-Direct Visual Odometry
UKF	: Unscented Kalman Filter
VIO	: Visual-Intertial Odometry
VO	: Visual Odometry
YTÜ	: Yıldız Teknik Üniversitesi

1. GİRİŞ

Otonom sistemler, çağımızda teknolojik ilerlemelerin öncüsü olarak öne çıkmaktadır. Bu sistemler, karar alma yeteneklerine sahip olan ve çevreleriyle etkileşimde bulunabilen bilgisayar tabanlı makinelerdir. Sürücüsüz otomobiller ve İnsansız Hava Aracı (İHA) gibi otonom araçlar, çeşitli sensor verilerini kullanarak çevrelerini sürekli tatar ve bu sayede güzergah planlaması ve diğer karmaşık görevleri gerçekleştirme yeteneğine sahip olurlar.

Ancak, otonom sistemlerin başlıca zorluklarından biri, çevrelerindeki dinamik engellerle etkileşimde bulunma ve bu engellerin farkında olma ihtiyacıdır. Bu önemli zorluk, otonom araçların güvenli ve etkili bir şekilde seyahat etmelerini sınırlayan kritik bir unsurdur. Dinamik engeller, aniden değişen trafik koşulları, yaya geçişleri, diğer araçlar ve yol çalışmaları gibi faktörlerden kaynaklanabilir.

Bu karmaşık zorluğun üstesinden gelmek amacıyla, otonom sistemlerin donanımına entegre edilen çeşitli sensörler devreye girmektedir. Kameralar, radarlar ve diğer algılayıcılar, otonom araçların çevresini sürekli olarak izleyerek anlık bilgiler elde etmelerini sağlamaktadır. Kameralar, görüntü işleme teknikleri kullanarak çevredeki nesneleri tanımlayabilir ve takip edebilir. Radarlar, çevredeki objelerin mesafesini ve hızını ölçerek potansiyel tehlikeleri belirleyebilir.

Otonom sistemler için diğer bir kritik mesele ise konumlandırma problemidir. Konumlandırma, otonom sistemlerin kendi pozisyonlarını belirleme ve “Ben neredeyim?” sorusuna cevap verme sürecidir. Bu süreç, çeşitli sensörlerden elde edilen verilerin entegrasyonu ve anlamlandırılmasıyla gerçekleşir. Örneğin, sürücüsüz otomobillerde kullanılan Global Navigation Satellite System (GNSS), kamera, IMU ve encoder gibi sensörlerden gelen bilgiler kullanılarak, aracın konumu belirlenir. GNSS geniş bir coğrafi alan üzerinde doğru konum bilgisi sağlarken, kamera çevresel detayları algılar ve IMU aracın hızını ve yönelimini belirler. Encoder, tekerlek hareketlerini ölçerek ek bir bilgi katkısı sağlamaktadır.

Ancak, konumlandırmayı sağlayan bu sensörlerden biri arızalandığında veya hatalı veri geldiğinde, sistemlerin doğru bir şekilde konumlandırması mümkün olmamaktadır. Bu durum, otonom sistemlerin güvenilir konumlandırma yeteneklerini

etkileyebildiği gibi sistemlerin çevresindeki dünyayı doğru bir şekilde anlamasını engelleyebilmektedir.

Bu tür durumlarla başa çıkma amacıyla sensör füzyonu önemli bir rol oynamaktadır. Sensör füzyonu, çeşitli sensörlerden gelen verileri entegre ederek sistem performansını artırır. Örneğin, bir sensör arızalandığında, diğer sensörlerin bilgilerini kullanarak eksikliği dengeleyebilir. Bu yaklaşım, otonom sistemlerin karmaşık trafik koşulları veya değişken çevresel faktörlerle karşılaştıklarında güvenilir bir konumlandırma sağlamalarına yardımcı olabilir. Bu tez çalışması, sensör füzyonu ve görsel tabanlı konumlandırma tekniklerinin birleştirilmesiyle otonom sistemlerin konumlandırma zorluklarına etkili çözümler sunma hedefini taşımaktadır.

Bu tez çalışmasında, otonom sistemlerin konumlandırma ve dinamik engel kaçınmalarına çözüm getirmek adına sensör füzyonu ve görsel tabanlı konumlandırma tekniklerini birleştirmek amaçlanmıştır. Farklı sensörlerden gelen bilgilerin etkili bir şekilde entegre edilmesi ve görsel bilgi kullanımıyla, otonom sistemlerin daha kesin, tutarlı ve güvenilir bir konumlandırma ve engellerden kaçınma yeteneğine ulaşmalarına yönelik bir katkı sağlanması hedeflenmiştir.

Otonom sistemlerin konumlandırma süreçleri, çeşitli teknik zorluklarla karşılaşmaktadır. Bu zorluklar, sistemlerin doğru bir şekilde konumlanma yeteneklerini sınırlandırabilir ve güvenilirliklerini etkileyebilir. Konumlandırma süreçlerinde ortaya çıkan temel zorlukları aşağıdaki gibi sıralanabilir:

- **Sensör Hata ve Arızaları:** Kullanılan sensörlerin arızalanması veya hatalı veri sağlaması durumunda, otonom sistemlerin konumlandırma doğruluğu ciddi şekilde azalabilir. Bu durum, sistemlerin güvenilirliğini zayıflatan kritik bir zorluktur.
- **Sensör Sınırlamaları ve Gürültü:** Konumlandırmada kullanılan GNSS, Light Detection and Ranging (LiDAR), kameralar ve Inertial Measurement Unit (IMU)'lar gibi sensörler, yapısından kaynaklanan sınırlamalara sahiptir. GNSS sinyalleri, şehir kanyonlarında veya sinyal kesiciler ile engellenebilir veya yanıltıcı olabilir. LiDAR ve kamera verileri, aydınlatma koşullarından etkilenebilir, IMU'larda ise zaman içinde kayma yaşanabilir. Sensör gürültüsü ve

kalibrasyon sorunlarıyla başa çıkmak, doğru konumlandırmayı başarmanın diğer kritik bir yönüdür.

- **Entegrasyon Zorlukları:** Farklı sensörlerden gelen bilgilerin etkili bir şekilde entegre edilmesi, konumlandırma sürecinde önemli bir adımdır. Ancak, bu farklı veri kaynaklarının uyumlu bir şekilde birleştirilmesi (sensör füzyonu), karmaşık bir konumlandırma zorluğunu ortaya çıkarmaktadır.
- **Çevresel Değişkenlik:** Otonom sistemler, çeşitli çevresel koşullar altında çalışabilirler. Ancak, bu değişken koşullar, konumlandırma hassasiyetini etkileyebilir. Özellikle hava şartları, ışık koşulları ve çevresel engeller, konumlandırma sürecini etkileyen faktörler arasında yer almaktadır.

Bütün bu zorlukları içinde barındıran bu tez çalışmasında, görsel ve diğer sensör verilerinin füzyonuna odaklanılmıştır. Sensör füzyonuna ilaveten, görüntü işleme ve yapay zekâ teknikleri de kullanılarak çok disiplinli bir yaklaşım kabiliyetleri kazanılmıştır. Otonom gerçek zamanlı bir sistem üzerinde geliştirilen tekniklerin denenmesi ile elde edilen sonuçlar paylaşılmıştır. Bu çerçevede içindeki sinerjileri ve çözümleri keşfetme arzumuz, otonom konumlandırmada içsel zorlukların üstesinden gelmeye adanmış büyüyen bilgi yelpazesine katkıda bulunmayı amaçlamaktadır.

1.3. Araştırma Soruları ve Hipotezleri

Belirlenen araştırma boşluğunu ve güçlü konumlandırma zorunluluğunu ele almak için, bu tez boyunca aşağıdaki araştırma soruları incelenmiştir:

Görsel ve sensör füzyonu tabanlı konumlandırmanın güçlü ve doğru performans elde etmek adına çeşitli ve dinamik ortamlardaki zorlayıcı koşullar, örneğin sınırlı GNSS erişimi veya aydınlatma problemleri, nasıl aşılabılır? Ayrıca, görsel kameralardan gelen verilerin gerçek zamanlı ve güvenilir konumlandırma için sorunsuz entegrasyonunu kolaylaştırmak için yeni algoritmalar ve metodolojiler nasıl geliştirilebilir? Makine öğrenimi, özellikle derin öğrenme modelleri, otonom sistemlerin algısal yeteneklerini artırmak ve görsel ve sensör füzyonu tabanlı konumlandırmanın doğruluğunu optimize etmek için nasıl kullanılabilir? Son olarak, gelişmiş görsel ve sensör füzyonu tabanlı konumlandırma tekniklerinin ulaşım, robotik ve hava görevleri gibi çeşitli alanlardaki otonom sistemlerdeki pratik uygulamaları ve elde edilen sonuçlar nelerdir?

Araştırma sırasında test edilen hipotezler şunlardır:

- H0: Görsel ve sensör füzyonu tabanlı konumlandırma teknikleri, uygun şekilde adapte edildiğinde ve geliştirildiğinde, zorlu ortamlarda tek sensörlü yaklaşımlara kıyasla konumlandırmanın doğruluğunu ve sağlamlığını önemli ölçüde artırabilmektedir.
- H1: Görsel ve sensör füzyonu için özel olarak tasarlanmış yeni algoritmalar ve metodolojiler, çoklu sensör entegrasyonu için gerçek zamanlı ve güvenilir konumlandırma performansına ulaşabilmektedir.
- H2: Derin öğrenme modelleri, sensör füzyon sürecine eğitildiğinde ve entegre edildiğinde, otonom sistemlerin karmaşık verileri yorumlama yeteneğini artırabilir ve özellikle zorlayıcı aydınlatma koşullarının olduğu senaryolarda daha yüksek konumlandırma doğruluğuna ulaşabilmektedir.
- H3: Gelişmiş görsel ve sensör füzyonu tabanlı konumlandırma tekniklerinin gerçek dünya uygulamalarındaki uygulanması, çeşitli alanlardaki otonom sistemler için verimlilik ve güvenlikte dikkate değer iyileştirmelere yol açmaktadır.

Bu araştırma sorularına yanıt verilerek ve hipotezleri test edilerek, otonom sistemler bağlamında güçlü ve hassas konumlandırmanın zorluklarını aşmayı hedefleyen bir tez çalışması gerçekleştirilmiştir.

1.4. Tez Yapısı

Bu tez çalışması, ana başlık olarak beş bölüme ayrılmıştır. Giriş bölümünde, araştırma problemi, problem zorlukları, sorular ve hipotezler ele alınmakta, ayrıca tez yapısının genel bir özeti sunulmaktadır. Literatür Araştırması bölümü, otonom sistemler ve konumlandırma alanındaki mevcut araştırmalara derinlemesine bir bakış sunarak, araştırmaların temelini oluşturan ilgili çalışma metodolojileri ve bulguları hakkında bilgi vermektedir. Metodoloji bölümü, tezde kullanılan araştırma yöntemlerini ve yaklaşımlarını detaylı bir şekilde açıklamaktadır. Otonom konumlandırma uygulamaları bölümü, tez döneminde geliştirilen uygulamaları ve bu uygulamaların sonuçlarını ayrıntılı bir anlatmaktadır. Sonuç bölümü ise tezin temel bulgularını özetler, yapılan katkıları vurgular ve bu katkıların otonom sistemler ve konumlandırma alanında önemini

tartırır. Bu bölüm aynı zamanda araştırma yolculuğunun bir özetı olarak, otonom sistemler ve konumlandırma alanındaki gelişmelerden bahsetmektedir.

Özet olarak, otonom sistemlerdeki konumlandırma sorunları ve bu alandaki zorluklar bu bölümde derinlemesine ele alınmıştır. Tez çalışması ile ilgili temel soruları ve bu sorulara yönelik hipotezleri belirlenmiş ve tezin odak noktaları ortaya koyulmuştur. Bir sonraki literatür Araştırması bölümünde, otonom sistemlerde konumlandırmaya dair mevcut ve önceki araştırmalar detaylı bir şekilde irdelenmiştir. Bu sayede tezin dayandığı temel prensipler açıklığa kavuşturulacaktır.



2. LİTERATÜR ARAŞTIRMASI

Bu bölümde, otonom sistemlerde görsel ve sensör füzyon tabanlı konumlandırma alanındaki mevcut çalışmalar kapsamlı bir şekilde incelenmiştir.

Chang ve Hebert'in çalışmasıyla birlikte çeşitli kamera kurulumları üzerine odaklanan konumlandırma araştırmalar, kamera parametrelerini yapılandırarak konum tahmini optimizasyonu gibi konuları ele almıştır (Chang & Hebert, 2000). Onların bu çalışmaları, özellik tabanlı görsel odometri konumlandırma yöntemlere odaklanmıştır. 2000'li yılların başında, SLAM yaklaşımlarında önemli bir dönemeç gerçekleşmiştir. Jin ve ark. (2003) ve Molton ve ark. (2004)'nin çalışmaları, hareket tahmini için ilk adımları atan, başlangıçta Structure From Motion (SFM)'a dayanan doğrudan yöntemleri içermiştir. Bu doğrudan tabanlı görsel odometri (Visual Odometry (VO)) yöntemlerin yanı sıra, Nistér (2004) ve Scaramuzza ve ark. (2006)'nın önerileriyle kamera parametrelerinin ego-hareketini (kamera veya sensör hareketini) tahmin etmekte kullanılmasıyla, literatürde bir özellik tabanlı VO boyutu eklenmiştir.

Caron ve ark., özellikle GNSS ve IMU verilerini birleştiren KF uygulamalarını içeren sensör füzyonuna odaklanmıştır (Caron ve ark., 2006). Ancak, bu yöntem doğrusal olmayan senaryolarla başa çıkmada sınırlamalar içermektedir. Bunun devamında, Scaramuzza ve ark.'nın öncü çalışmasıyla birlikte, hibrit bir yaklaşım ortaya çıkmıştır (Scaramuzza ve ark., 2008). Bu yaklaşım, pozisyon tahmini için özellik tabanlı ve doğrudan tabanlı VO yöntemlerini düz bir yüzeyde dönüş tahmini için birleştirmiş, bu da farklı paradigmaların başarılı bir birleşimini gerçekleştirmiştir.

Araştırmacılar konum tahmin sürecini optimize etmek amacıyla Demet Ayarlama (Bundle Adjustment (BA)) yöntemlerini keşfetmeye başlamışlardır, bu da konumlandırmanın doğruluğunu iyileştirmeye yönelik bir adım olarak dikkat çekmiştir (Triggs, 2000). BA, görüntüdeki ilişkili noktalar arasındaki mesafeyi en aza indirmeyi hedefleyen bir yinelemeli optimizasyon tekniğidir, bu da modelin yansıması (reprojection) ile ilgilidir. BA kullanarak, Kitt ve ark.'nın LIBVISO'su, özellik tabanlı VO yöntemlerin tekli ve stereo kamera yapılandırmalarında tahmin yeteneklerini geliştirmiştir (Kitt ve ark., 2010). Araştırma devam ettikçe, Silveira ve ark. (2008) ve Pretto ve ark. (2011)'nin çalışmalarıyla birlikte, çerçeve içinde hareket tahmini için daha gelişmiş optimizasyon yaklaşımlarına doğru bir metodolojik değişim yaşanmıştır.

Newcombe ve ark. (2011), anahtar çerçevelerin (key-frame) yoğun derinlik haritalarını değerlendirmeyi amaçlamıştır. Bu metodoloji, doğru konum tahmini için küresel bir enerji fonksiyonunu optimize etme konseptini tanıtmış, ancak güçlü bir GPU üzerinde çalışma ihtiyacını beraberinde getirmiştir. 2011'de, modelin karmaşıklığını azaltmak için fiziksel sınırlamalardan yararlanan bir nokta hareket değerlendirme algoritması önerilmiştir (Scaramuzza, 2011). Bu metodolojik yenilik, doğruluğu korurken hesaplama sürecini basitleştirmeyi amaçlamıştır. Ayrıca, görsel bilginin GNSS veya IMU verileri gibi destekleyici sensörler ile entegrasyonu, konumlandırma tahminlerinin sağlamlığını artıran önemli bir eğilim haline gelmiştir (Kneip ve ark., 2011). Aynı yılda, EKF'nin performansını artırmak amacıyla ortaya çıkan kokusuz kalman filtresi (Unscented Kalman Filter (UKF)), metodolojik bir ilerleme olarak dikkat çekmiştir. UKF, EKF'ye kıyasla daha geniş tahmin dizisini inceleyerek doğruluğu artırma potansiyeli sunmuştur (Ndjeng ve ark., 2011). Alternatif yöntemleri keşfetmek adına, Kazik ve ark. (2012), stereo kamera sistemini taklit etmek için örtüşmeyen (non-overlapping) kameraları kullanan bir öneri sunmuştur. Bu metodoloji ile her kameradan monoküler VO tahmin edilmesini içerir ve ardından stereo kısıtlaması uygulanır. Böylece, VO araştırmalarındaki metodolojik stratejilerin çeşitlenmesi gösterilmektedir. DTAM (Dense Tracking and Mapping) tarafından ortaya konan hesaplama zorluklarına yanıt olarak, Engel ve ark. 2013 yılında tanıttıkları bir yöntemle, bu gereksinimleri hafifletmeye çalışarak daha hesaplama açısından verimli doğrudan tabanlı VO yaklaşımlara doğru bir metodolojik geçiş önermiştir (Engel ve ark., 2013).

Hibrid yöntemlere olan ilgi, özellikle monoküler VO ve VIO alanında giderek artan bir ilgi olarak belirginleşmiştir (Engel ve ark., 2013). Bu kronolojik ilerleme, hibrid yaklaşımların gerçek dünya uygulamalarında etkinliğinin artan bir şekilde tanınmasını göstermektedir. Forster ve ark.'nın SVO (Semi-Direct Visual Odometry)'yu tanıtmaları, hibrid yöntemler içinde başka bir metodolojik ilerleme temsil etmektedir (Forster ve ark., 2014). SVO, özellik ayarlamalarının projeksiyon hatalarını azaltarak konumlandırmaları optimize etmiş ve böylece, daha sofistike optimizasyon stratejilerine doğru bir geçişi göstermiştir. LSD-SLAM'ın (Engel ve ark., 2014) tanıtılması, hızlı bir doğrudan monoküler SLAM algoritması sunarak kronolojik bir ilerlemeyi temsil etmiştir. Bu metodolojik yenilik, doğrudan yaklaşımlarda gerçek zamanlı performans ihtiyacına bir yanıt olarak ortaya çıkmıştır.

Silva ve ark., pozisyon tahminini özellik tabanlı bir VO yaklaşımla birleştirip ardından bir Kalman filtresi aracılığıyla rafine etmeyi önererek ego-hareket tahmini için metodolojilerin birleşimini tanıtarak hibrid bir yaklaşım önermiştir (Silva ve ark., 2015). Bu erken gelişmeleri temel alan Leutenegger ve ark., 2015 yılında OKVIS yöntemini tanıtmıştır (Leutenegger ve ark., 2015). OKVIS, hem IMU ölçümlerini hem de görsel reproje hatalarını dikkate alan doğrusal olmayan optimizasyon tekniklerini kullanarak sıkı bağlı paradigmayı daha da rafine etmiştir. Bu yöntem, her anahtar çerçevenin konumunu bir kayan pencere içinde optimize ederken eski çerçeveleri marjinalleştirerek verimliliği korumaktadır. Bu çalışma, Harris köşe dedektörleri ve BRISK tanımlayıcıları kullanarak, VIO için özellik eşleme konusundaki metodolojik ilerlemeyi örnekleyen bir çalışma olmuştur.

ConvNets'in PoseNet (Kendall ve ark., 2015) içinde mutlak kamera konumu regresörleri olarak kullanılması, küresel konumlandırmada önemli bir dönemeçtir. PoseNet, kamera konumlarını tahmin etmek için uçtan uca bir yaklaşım getirerek, derin öğrenme tabanlı bir küresel konumlandırma yöntemini temsil eder. Erken dönemde, Konda ve Memisevic ego-hareketi çıkaran bir sınıflandırma tekniği tanıtmıştır (Konda & Memisevic, 2015). Bu yöntem, kamera görüntüleri arasındaki ego-hareketi çıkarmak için convNet ve softmax katmanını kullanmıştır. Yaklaşımları, yönlendirme ve hızda belirgin değişikliklere odaklanmış ve ego-hareket tahminine erken bir katkıda bulunmuştur.

Bu erken gelişmeleri temel alan bir çalışma (Costante ve ark., 2015), CNN (Convolutional Neural Network) kullanarak yoğun optik akışlardan görsel özellikler çıkarmış ve bu özelliklere dayalı olarak ardışık hareketi tahmin etmiştir. Ancak, bu yaklaşımlar hala görüntülerden doğrudan uçtan uca konum tahmini yapmada sınırlamalarla karşılaşmıştır. Metodolojiyi çeşitlendirmek adına, IMM (Interacting Multiple Model) tekniği (Bacha ve ark., 2016) ve OKPS (Optimized Kalman Particle Swarm) (Noureldin ve ark., 2011), parçacık filtreleme yöntemini sürü yöntemleriyle birleştirerek, yeni füzyon stratejilerine yönelik bir arayış içinde olduklarını yansıtmıştır.

Nicholai ve ark. (2016), ego-hareketi tahmin etmek için görüntü ve LiDAR verilerini birleştiren yenilikçi bir yaklaşım önermiştir. İki boyutlu görüntülerden nokta bulutunu çıkararak ve bu veriyi bir sinir ağına aktararak, çoklu duyuşal verilerin ego-hareket tahmininde entegrasyonunu sergileyerek bir yol tahmini elde etmişlerdir. Zorluğu aşmak için Mohanty ve ark. (2016), uçtan uca robot konum tahminine olanak tanıyan bir

Siamese ve RNN tabanlı bir mimari tanıtmıştır. Yaklaşımları, eş değer ağırlıklarla bir L2 hata katmanı içeriyordu ve konum regresyonunun doğruluğunu artırmaktaydı. Melekhov ve ark. (2017) ilgili bir çalışmada, konum parametrelerinin kaybını dengelemek için bir ağırlık kavramını tanıtmış ve bu, geliştirilmiş bir konum regresyonu sağlamıştır. Ayrıca, çeşitli görüntü kalitelerinde metodun sağlamlığını artırmak için bir çerçeve ve bir havuzlama katmanını içeren bir çerçeve de dahil etmişlerdir. Bazı durumlarda, konum tahmini ile ilgili zorluklar, stereo veya RGB-D (Red-Green-Blue-Depth) kameralar gibi özel kamera kurulumlarını içermek suretiyle ele alınmıştır. Mur-Artal ve Tardos (Mur-Artal & Tardós, 2017), ORB-SLAM2'yi tanıtarak ORB-SLAM'ın yeteneklerini hem stereo hem de RGB-D kamera yapılandırmalarını içerecek şekilde genişletmiştir. Bu, özel kamera donanımını kullanarak konum tahmininin karmaşıklıklarını ele alma konusunda önemli bir değişimi temsil etmektedir.

Engel ve ark. (2017), fotometrik kusurun doğrudan bir yaklaşım içinde seyrek demet modifikasyonu olarak optimize edildiği bir yöntem sunmuştur. Bu yaklaşım, genellikle özelliklerle kodlanan geometrik önyargıların gerekliliğini ortadan kaldırmış ve ego-hareket tahmini için desensiz alanlarda bile tüm resim noktalarını kullanmayı başarmıştır. Bu, doğrudan tabanlı VO yöntemlerde önemli bir optimizasyon ilerlemesini temsil etmektedir. Bu metodolojik evrim, Silva ve ekibinin (Silva ve ark., 2017) çalışmasıyla tamamen yoğun bir stereo ego-hareket olasılık sistemini oluşturmak için daha da genişletilmiştir. Bu gelişme, sistemin sağlamlığını artırmakla kalmayıp aynı zamanda daha zorlu ortamlara uyum sağlama yeteneğini de göstermiştir. PoseNet'in temelini oluşturan çalışmaların ardından, Kendall ve Cipolla'nın (Kendall ve Cipolla, 2017) gibi sonraki araştırmalar, konum tahmininin verimliliğini artırmak için çoklu görüş geometrisinin entegrasyonunu keşfetmiştir. ResNet34'ün orijinal boru hattında Melekhov ve ark. (2017) kullanılması ve boyut indirgeme için LSTM'nin benimsenmesi gibi daha fazla metodolojik yenilik devreye girmiştir.

Wu ve ark.'nın (2017) ve Naseer ve Burgard'ın (Naseer ve Burgard, 2017) çalışmaları, eğitim verilerini artırmak için sentetik hedef görüntü oluşturmayı kullanmış ve küresel konumlandırma sürecini geliştirmek için veri odaklı bir yaklaşımı tanıtmıştır. Zamanla ilgili yönleri değerlendirmek için Clark ve ark. (2017) çalışması, mekânsal kısıtlamaların yanı sıra zaman kısıtlamalarını da entegre etmiş ve görüntüler arasında zaman bağlantıları kurarak metodolojik bir değişiklik yapmıştır. Sentetik hedef görüntü

oluşturma, metodolojik gelişimde bir rol oynamaya devam etmiş ve Purkait ve ark. (Purkait ve ark., 2018) çalışmasında gösterildiği gibi, eğitim sürecini iyileştirmedeki kullanışlılığını daha da vurgulamıştır. Ancak, yukarıda bahsedilen tüm ego-hareket yaklaşımlarının gözetimli yöntemler olduğunu, eğitim için gerçek (ground-truth) etiketlerine dayandığını belirtmek önemlidir. Gerçek (ground-truth) etiketlerini elde etmek, pratikte zorlayıcı ve maliyetli olabilir, bu da etiketli eğitim veri setlerinin sınırlı olmasına neden olabilir. Bu kısıtlamalara yanıt olarak, gözetimsiz öğrenme yöntemlerine yönelik artan bir ilgi olmuştur. Gözetimsiz yöntemler, etiketlenmemiş veriden öğrenebilir, insan etiketleme çabalarının ihtiyacını azaltabilir ve etiketli verilerin kısıtlı olabileceği görünmeyen ortamlarda daha büyük adaptasyon sağlayabilir. Zhou ve ark. (2017) çalışmasında olduğu gibi, video resimlerinden derinlik ve kamera hareketini öğrenmek için görünüm sentezi tekniklerini kullanan bir öz-denetim mimarisinde örnekendirilmiştir.

Düşük maliyetli ve enerji verimli sensörler olarak değerlendirilen IMU'lar, araştırmacılar tarafından derin öğrenme temelli VO'ya entegre edilmeye başlanmıştır. Öncü bir çalışma olan VINet (Clark ve ark., 2017), VIO'nun ardışık öğrenme sorununa kapsamlı bir ağ mimarisi ile çözüm getirmiştir. VINS-Mono (Qin ve ark., 2018) ise sıkı bağlı veri füzyonunda metodolojik bir ilerleme temsil etmektedir. Görsel ve ataletsel verilerin entegrasyonunu vurgulayan doğrusal olmayan optimizasyona dayanan bu yaklaşım, sensör füzyon tabanlı bir yöntemdir. GNSS bilgilerini kullanarak hareket tahminini iyileştirmek için Brahmbhatt ve ark. (2018) tarafından öne sürülen metodoloji, derin öğrenme tabanlı bir küresel konumlandırma yöntemidir. Valada ve ark. (2018) tarafından gösterildiği gibi, küresel konumlandırmanın göreceli konum tahmin ağlarıyla entegrasyonu, bu iki bileşenin kapsamlı konum tahmini için birleştirilmesinin potansiyelini vurgulayan metodolojik bir birleşim olarak kabul edilebilir.

Radwan ve ark. (2018), karmaşık bilgi kısıtlamalarını tanıtarak ve bu bilgileri konum regresyon ağlarıyla birleştirerek, yüksek konum tahmini doğruluğu elde etmek için yenilikçi bir metodolojik yaklaşımı göstermiştir. Ancak, öz-denetim yöntemlerinin kendi zorlukları olduğunu, ölçek belirsizliği ve kamera örtüşmelerine duyarlılığı içerdiğini belirtmek önemlidir. Bu sorunları ele almak için (Li ve ark., 2018; Yin & Shi, 2018; Zhan ve ark., 2018; Yang ve ark., 2018; Zhao ve ark., 2018; Alamlioglu ve ark., 2019; Li ve ark., 2019; Sheng ve ark., 2019) gibi sonraki çalışmalar, gözetimsiz ego-

hareket tahmin metodolojilerini daha da iyileştirmiştir. Yapay zekanın sensör füzyonuna tanıtılması için Kim ve ark. (2019), gecikmeli girişli bir sinir ağı kullanarak IMU'daki hataları tahmin etmeye yönelik bir metodoloji geliştirmiştir. Böylece, GNSS verilerini IMU verileriyle birleştirmek için yapay zeka tekniklerini kullanma konusuna giriş yapılmıştır. Kim ve ark. (2019), LSTM ağlarını sensör füzyonu için kullanmışlardır. Yapay zekanın uygulanmasını daha fazla geliştirmek için Chen ve ark. (2019), IMU hatalarını azaltmak için derin öğrenme tekniklerini kullanmış ve bu, navigasyon sistem sonuçları üzerinde olumlu bir etki yapmıştır.

Huang ve ark. (2019), konumlandırma doğruluğunu artırmak amacıyla özellikle karmaşık ortamlarda öz-dikkat (self-attention) modüllerini tanıtmıştır. Bu modüller, ağın ilgili bilgilere odaklanma yeteneği kazanmasını sağlarken aynı zamanda öne çıkan unsurları göz ardı etme özelliğine sahiptir. Xue ve ark. (2019), küresel konumlandırma ve göreceli konum tahmini ağlarının entegrasyonu üzerine çalışmış ve bu metodolojilerin birleştirilmesinin potansiyel faydalarını vurgulamıştır. Semantik segmentasyonu, sensör füzyonu ve bunların konumlandırma için birleştirilmesi, Larsson ve ark. (2019) tarafından daha fazla geliştirilmiştir. Bu, bilgi entegrasyonunu geliştirerek konum tahminini iyileştirme yolundaki metodolojik bir eğilimi göstermektedir. Saputra ve ark. (2019), ego-hareket tahminini geometrik hata kısıtlamalarını kullanarak genişletmiştir. Xue ve ark. (2019) da, bir bellek modülü ve rafine modülü uygulayarak küresel bilgi depolamak için ve bu şekilde konumlandırmayı iyileştirmek için bir yöntem geliştirmiştir.

Chen ve ark. (2019), VIO için bağlam farkındalıklı temsilleri seçici bir şekilde öğrenen bir sensör füzyon mimarisi tanıtmıştır. Ayrıca, ego-hareket tahminini geliştirmek için IMU verilerini entegre etmenin örneklerini sunmuştur. VIO Learner (Shamwell ve ark., 2019), IMU'nun içsel parametreleri veya IMU ve kamera arasındaki dışsal kalibrasyonu gerektirmeyen ham ataletsel verilerden doğrudan hareket tahminleri elde etmek için benzersiz bir yaklaşım benimsemiştir. DeepVIO (Han ve ark., 2019), IMU ve stereo kamera verilerini entegre etmiş, küresel ölçekte konum tahmin etmek için özel bir hata ile eğitim yapmıştır. Sonrasında, Sun ve ark. (2020), iki aşamalı EKF tabanlı bir algoritma kullanarak IMU, GNSS ve monoküler VO verilerini birleştiren etkili bir yöntem sunmuştur. Bu, çeşitli sensör modalitelerini birleştirerek konumlandırma doğruluğunu artırmayı amaçlayan bir Sensör Füzyon Tabanlı yöntemdir. RNN tabanlı

teknikler, Dai ve ark. (2020) tarafından, GNSS ve IMU sensörlerini sürekli ve güvenilir bir navigasyon elde etmek için birleştirerek merkezi bir rol oynamıştır. RNN, önceki IMU verilerinden yararlanarak güncel ölçümlerdeki hataları öngörmek için kullanılmış, bu da sensör füzyonunda tekrarlayan ağlar ve bellek fonksiyonlarının rolünü vurgulamıştır.

Wang ve ark. (2020), karmaşık ortamlarda konumlandırma doğruluğunu artırmak için öz-dikkat (self-attention) modüllerinin etkinliğini göstermiştir. Bu metodolojik yenilik, sağlam küresel konumlandırma elde etme konusundaki önemi pekiştirmiş, derin öğrenme tabanlı bir küresel konumlandırma yaklaşımını temsil eder. Sonrasında, Sun ve ark. (2021), GNSS/IMU entegre konumlandırmada çoklu hataları tespit etmek ve dışlamak için bir algoritma tanıtmıştır. Bu yöntem, IMU verileri ve GNSS sahte menzil ölçümlerini kullanarak gerçek zamanlı hata tespitini sergileyerek, sensör füzyon tabanlı bir yaklaşımı temsil etmektedir. Ayrıca, Zhu ve ark. (2021) ve Wei ve ark. (2021) da LiDAR verilerini de ekleyerek derin öğrenme ve sensör füzyonu yöntemlerini kullanarak konumlandırma işlemini daha da iyileştirmişlerdir.

Sensör füzyonu ve derin öğrenme sayesinde, görsel ve sensör bilgilerini birleştirmek ve konumlandırma için seçici dikkat mekanizmaları da geliştirilmiştir (Zhao ve ark., 2022). (Chen ve ark., 2022) çalışmada, nokta bulutları ile sensör bilgisi bir araya getirilerek çoklu sensörlü bir yol tahmin ağı oluşturulmuştur. Derinlik tahmininde geometrik bilgiyi netleştirmek için (Liu ve ark., 2022), çerçeve genelinde geometrik öncül ve piksel düzeyinde hassasiyet eklemiştir. Bu yaklaşım, ego-hareket ve derinlik tahmininde zamanında sektörde en iyi performansı elde etmiştir. 4DRVO-Net (Zhuo ve ark., 2023), görüntü ve radar bilgisini birleştirmek için çok ölçekli bir birleştirme modülü kullanarak, konumlandırmada doğru olmayan veri eşleştirmeleri sorununu çözmeye çalışmıştır. Tüm gün boyunca derinlik tahmini sorununu çözmek, derinlik tahminine çok spektral görüntüler eklemiştir. (Guo ve ark., 2023), çoklu spektrum arasındaki özellik farkını azaltmak için çerçeve genelinde geometrik öncül ve piksel düzeyinde hassasiyet eklemiştir. (Shao ve ark., 2023), çerçeveye sahte lidar derinliği ekleyerek derinlik tahmininin doğruluğunu ve bu şekilde konumlandırma hassasiyetini de artırmıştır.

Literatür Araştırmasını sonlandırırken, konumlandırma ve navigasyon için sensör füzyonu alanının dönüştürücü bir yolculuğa tanıklık ettiği açık hale gelmektedir. Kalman filtreleme gibi klasik yöntemlerden, gelişmiş yapay zeka ve derin öğrenme tekniklerinin

entegrasyonuna kadar, araştırmacılar konumlandırma sistemlerinin doğruluğunu ve güvenilirliğini artırmaya çalışmışlardır. Her yaklaşım, sensör füzyonunun genel anlayışına ve potansiyel uygulamalarına katkıda bulunmuştur. Ancak, mevcut literatürde devam eden kısıtlamaları ve boşlukları kabul etmek önemlidir. Bu kısıtlamalar, doğrusallıklarla başa çıkma, yaklaşım hataları ve çeşitli sensör modalitelerini kapsamlı bir şekilde entegre etme ihtiyacını içermektedir. Ayrıca, derin öğrenme yaklaşımları umut vaat etse de, sensör füzyonundaki uygulamaları henüz yeterince keşfedilmemiş olup, bu yönde daha fazla araştırmaya ihtiyaç bulunmaktadır.

Bu tez, sensör füzyonu temelli konumlandırma alanındaki kısıtlamaları ele almak için yenilikçi çözümler önermektedir. Çeşitli sensörleri entegre etme ve doğruluğu artırma amacıyla derin öğrenme tekniklerinin gücünden faydalanılmıştır. Otonom sistemlerin ve ilgili teknolojilerin ilerlemesine katkıda bulunulmuş, konumlandırma ve navigasyon alanında elde edilebileceklerin sınırları zorlanmıştır.

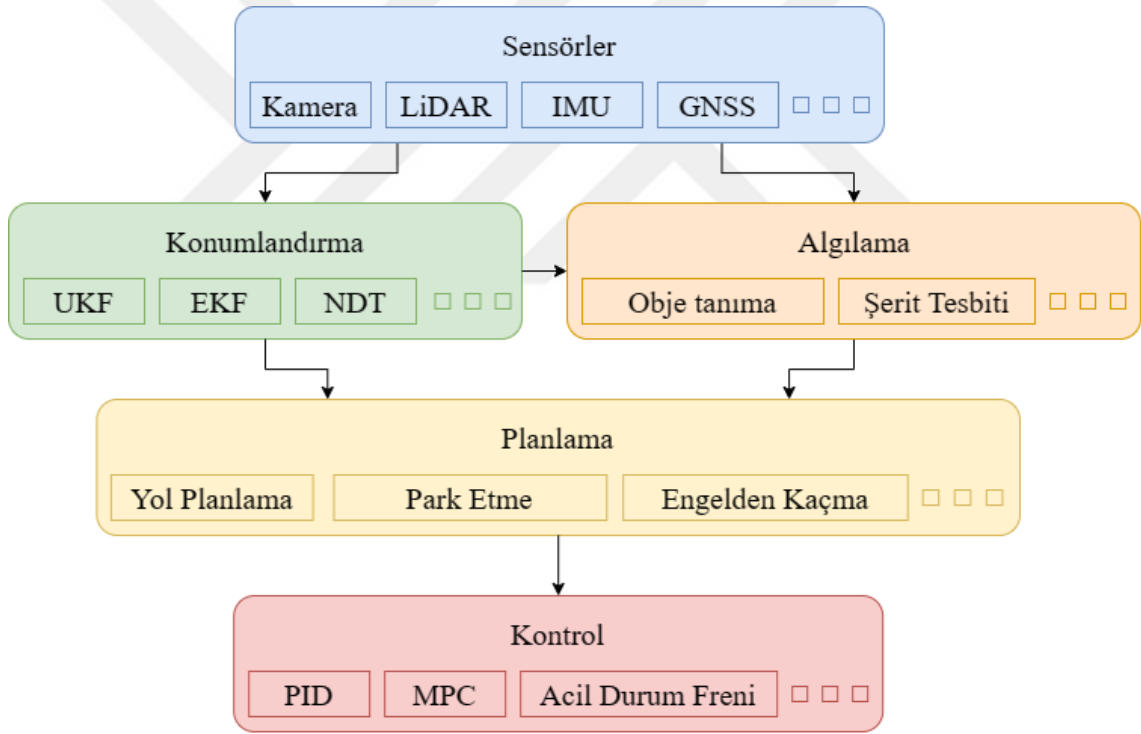
Özet olarak, Literatür Araştırması bölümünde otonom sistemlerdeki konumlandırma sorunlarına dair geniş bir perspektif kazanılmıştır. Bu alandaki önceki çalışmalar incelenerek, mevcut yöntemlerin avantajları ve zayıf yönleri anlaşılmıştır. Sonraki metodoloji bölümünde, önceki literatürden elde edilen bilgileri temel alarak, otonom sistemlerde konumlandırma için geliştirilen yeni yaklaşımlar ve bu yaklaşımların dayandığı temel prensipler detaylı bir şekilde açıklanacaktır. Bu metodoloji, tezin temel yapı taşlarını oluşturacak ve elde edilen sonuçları daha iyi anlaşılmasına yardımcı olacaktır.

3. METODOLOJİ

Bu bölümde, tez boyunca kullanılan metodolojileri ve teknikleri, konum tahmini problemini odak noktası olarak sunulmaktadır.

3.1 Otonom Araç Nedir?

Otonom araçlar, kamera, LiDAR, IMU, GNSS ve radar gibi sensörlerle donatılmış, insan müdahalesi olmadan hareket edebilen araçlardır. Bu araçlar, algı verilerini yorumlamak, kararlar almak ve güvenli bir şekilde hareket etmek için gelişmiş algoritmalar ve yapay zeka kullanmaktadır. Şekil 3.1, bir otonom aracın genel blok şemasını göstermektedir. Bu blok şeması içinde sensörler, konumlandırma, algılama, planlama ve kontrol modüllerini içermektedir.



Şekil 3.1. Otonom araç genel blok şeması.

Sensörler modülü, otonom aracın çevresini algılayan donanım birimlerini içermektedir. LiDAR, kamera, radar, IMU ve GNSS gibi sensörler aracılığıyla çevresel veriler toplanır. LiDAR, nesnelerin uzaklık ve konum bilgilerini sağlar; kamera, görüntü işleme ve nesne algılama için kullanılır; radar, mesafe ölçümleri ve hava durumu izleme amacıyla kullanılır; IMU ise aracın hızını, ivmesini ve yönelimini belirlemektedir. Bu

sensörlerden gelen veriler, çevresel harita oluşturma ve aracın konumunu belirleme süreçlerinde kullanılmaktadır.

Konumlandırma modülü, sensörlerden alınan verileri kullanarak otonom aracın konumunu belirlemektedir. GPS, LiDAR, kamera ve IMU gibi sensörler aracılığıyla elde edilen bilgiler entegre edilerek, aracın hassas bir konumlandırması sağlanmaktadır. Bu modül, aracın çevresel haritasını oluştururken ve yol planlaması yaparken kritik bir rol oynamaktadır. Bu modülde EKF, UKF gibi diğer konumlandırma algoritmalar kullanılmaktadır.

Algılama (Perception) modülü, çevresel sensörlerden gelen verileri işleyerek çevredeki nesneleri algılamakta ve tanımaktadır. Görüntü işleme, derin öğrenme ve diğer algılama algoritmaları kullanılarak, aracın etrafındaki yollar, araçlar, yayalar ve diğer nesneler belirlenmektedir. Algılama modülü, otonom aracın çevresini anlamasına ve güvenli bir şekilde hareket etmesine yardımcı olmaktadır.

Planlama modülü, algılama ve konumlandırma sonuçlarına dayanarak aracın hareket stratejisini belirlemektedir. Hedef konuma ulaşmak için uygun bir rota oluşturur ve çevresel engelleri dikkate alarak güvenli bir yol planı yapmaktadır. Bu modül, aracın önceden belirlenmiş hedeflere yönlendirilmesini sağlar ve ani değişikliklere tepki verme yeteneğiyle otonom sürüşü güvenli kılmaktadır.

Kontrol modülü, planlama modülünden gelen komutları alır ve aracın gerçekte nasıl hareket etmesi gerektiğini belirlemektedir. PID denetleyicileri ve diğer kontrol algoritmaları, aracın hızını, yönlendirilmesini ve diğer dinamik parametrelerini düzenlemektedir. Bu modül, otonom aracın planlanan hareketi gerçekleştirmesini sağlar ve çevresel değişikliklere dinamik bir şekilde adapte olabilmektedir.

3.2. Konum ve Konum Tahmini (Konumlandırma)

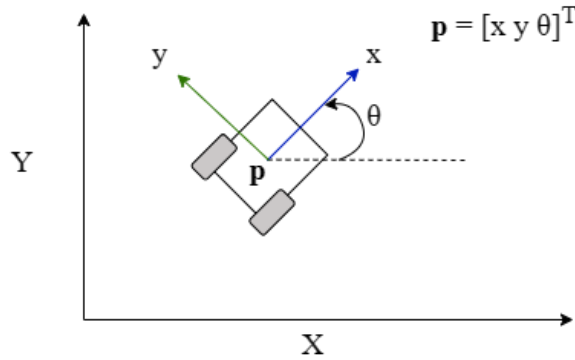
Otonom sistemler ve robotik alanlarında "konum" terimi, bir nesnenin veya cihazın üç boyutlu uzaydaki konumunu ve rotasyonunu ifade etmektedir. Konum, matematiksel olarak, iki bileşenin kombinasyonu olarak temsil edilebilir:

1. Pozisyon: Bir nesnenin pozisyonu genellikle 3B vektör olarak temsil edilir ve $P(x,y,z)$ olarak gösterilir. Burada x,y ve z sırasıyla x,y ve z eksenlerindeki koordinatları ifade etmektedir. Bu vektör, nesnenin uzaydaki konumunu tanımlamaktadır.

2. Rotasyon: Nesnenin referans koordinat sistemine göre açısal hizalanmasını açıklar. Genellikle bir rotasyon matrisi R veya bir kuaterniyon q olarak temsil edilir. Bu matematiksel yapılar, nesnenin uzayda nasıl döndürüldüğünü yakalar.

Kuaterniyon, matematiksel olarak dört bileşenli bir sayı sistemini ifade eder ve genellikle w , x , y , ve z bileşenlerinden oluşur. Bu bileşenler, bir nesnenin üç boyutlu uzaydaki rotasyonunu temsil etmektedir. w bileşeni, kuaterniyonun büyüklüğünü belirtirken, x , y , ve z bileşenleri ise rotasyon eksenini tanımlar. Kuaterniyonlar, özellikle üç boyutlu uzaydaki dönmeleri ifade etmek için kullanılır ve rotasyon matrisleriyle eşdeğerdir. Matematiksel olarak karmaşık gibi görünebilirler, ancak pratikte kuaterniyonlar, rotasyonların etkileşimli bir şekilde birleştirilmesi için özellikle avantajlıdır. Kuaterniyon, oryantasyon problemleriyle başa çıkmak ve özellikle robotik sistemlerde hassas rotasyonları ifade etmek için kullanılan güçlü bir araçtır.

Pozisyon ve rotasyonun birleşimi, nesnenin üç boyutlu uzaydaki konumunu tam olarak ifade etmektedir. Şekil 3.2'de, bir mobil robotun 2B pozisyonunu temsil eden x ve y bileşenleri ile rotasyonunu belirten θ 'nın bir $\mathbf{p}$ vektörü olarak gösterildiği görülmektedir.



Şekil 3.2. Mobil Robotun 2D pozisyonu (x , y) ve rotasyonu (θ).

Konum Tahmini, mevcut sensör ölçümlerini kullanarak bir nesnenin veya cihazın konumunu belirleme sürecidir. İnsansız hava araçları veya robotlar gibi otonom sistemlerde navigasyon, haritalama ve çevre ile etkileşim için konum tahmini hayati öneme sahiptir.

Konum tahmini, matematiksel olarak, öngörülen ve gözlemlenen sensör ölçümleri arasındaki hatayı minimize ederek optimal konum vektörünü (pozisyon ve rotasyon) P_{est} bulmayı içerir. Denklem (3.1)'de; P_{est} , tahmini konum vektörünü, $P_{observed}$, ise gözlemlenen sensör ölçümlerini temsil etmektedir. Hata fonksiyonu $E(P, P_{observed})$, tahmini konum P ile gözlemlenen sensör ölçümleri arasındaki farklılığı ölçer ve Denklem (3.1) ile bu hata minimize edilerek en iyi konum tahmini elde edilmeye çalışılır.

$$P_{est} = \operatorname{argmin} E(P, P_{observed}) \quad (3.1)$$

Belirli bir uygulamaya ve mevcut sensör verilerine bağlı olarak, konum tahminini doğru ve güvenilir bir şekilde gerçekleştirmek için derin öğrenme modelleri, sensör füzyonu ve filtreleme yöntemleri gibi çeşitli teknikler kullanılabilir. Konum tahmini veya konumlandırmada, görsel tabanlı konumlandırma, sensör füzyon tabanlı konumlandırma ve derin öğrenme tabanlı konumlandırma olmak üzere üç ana kategori bulunmaktadır.

3.2.1 Görsel Tabanlı Konumlandırma

Görsel tabanlı konumlandırma, genellikle görsel odometri (Visual Odometry (VO)) olarak adlandırılır ve ardışık alınan anlık kamera görüntüleri değerlendirilerek referans koordinat sistemine göre kamera hareketi tahmin edilmektedir. Genellikle, literatürde geleneksel VO yöntemleri, özellik tabanlı, doğrudan tabanlı ve hibrid tabanlı yaklaşımlar olmak üzere üç farklı türe ayrılmıştır.

- **Özellik Tabanlı Yöntemler:** Özellik tabanlı yöntemler, SURF (Speeded Up Robust Features) (Bay ve ark., 2006), FAST (Features from Accelerated Segment Test) (Trajković ve Hedley , 1998), SIFT (Scale-Invariant Feature Transform) (Lowe, 2004), ORB (Oriented FAST and Rotated BRIEF) (Rublee ve ark., 2011) ve Harris ve Stephens (Harris & Stephens, 1988) gibi çeşitli özellik çıkartma tekniklerini kullanarak önemli özellik noktaları tespit etmektedir. Bu özellikler, daha sonra bir ortak özellik takipçisi kullanılarak bir sonraki ardışık karede izlenir. En yaygın olanı KLT (Kanade-Lucas-Tomasi) takipçisi optik akışa dayanmaktadır (Tomasi & Kanade, 1992), (Shi, 1994).
- **Doğrudan Tabanlı Yöntemler:** Görüntü piksellerinin yoğunluğuna dayanan ve fotometrik hata optimizasyonu kullanan doğrudan tabanlı yöntemler, genellikle

hareket tahmini için kullanılmaktadır. Bu yöntemler, çeşitli hesaplama ve doğruluk zorluklarına çözüm getirmek amacıyla geliştirilmiştir.

- **Hibrid Yöntemler:** Bu yöntemler, özellik tabanlı ve doğrudan tabanlı yöntemlerin avantajlarını birleştiren, özellikle karmaşık ve yapılandırılmamış senaryolarda algoritmik sağlamlığı artırmak için tercih edilmektedir.

3.2.2. Sensör Füzyon Tabanlı Konumlandırma

VO, yalnızca görsel verilere dayalı olarak konum tahmin etme konusunda önemli ilerlemeler kaydetmiş olsa da, özellikle düşük ışıklı ortamlarda, düz duvar gibi doku eksikliği olan ortamlarda veya ani görüntü değişikliklerinde sınırlamalara sahiptir. Yalnızca VO'ya güvenmek, doğruluk ve konum kaymaları sebebiyle doğru bir tercih olmayabilir. Ancak görsel ve sensör füzyon tabanlı konumlandırma yöntemini benimseyerek, bu dezavantajlar etkili bir şekilde hafifletilebilir. IMU, LiDAR veya derinlik sensörleri gibi ek sensör verilerini entegre etmek, otonom sistemlerde konumlandırmanın doğruluğunu artıran tamamlayıcı bilgiler sunmaktadır. Görsel ve sensör verilerinin bu şekilde birleştirilmesi, çevrenin daha kapsamlı bir anlaşılmasına olanak tanır, bu da zorlu koşullarda bile hassas konum tahmini ve gezinme için daha güvenilir bir temel sağlamaktadır. Geleneksel olarak, iki tür veri füzyon şeması vardır: gevşek ve sıkı bağlı paradigma. Gevşek bağlı durumda, konum tahmini görsel ve IMU verileri ayrı ayrı hesaplanır ve ardından birleştirilir. Bunun yerine, sıkı bağlı şemada, hem ham görsel hem de IMU verisinin birleşimini ele alarak konum tahmini gerçekleştirilir (Scaramuzza & Zhang, 2019).

3.2.3. Derin Öğrenme Tabanlı Konumlandırma

Derin öğrenmenin konumlandırma yöntemlerine entegrasyonu, nispeten yeni bir gelişme olmasına rağmen, derin öğrenmenin bu yeteneklerini kullanmak için önemli araştırmalar yapılmaktadır. Araştırmacılar bu çabaları genellikle iki ana alan olarak kategorize etmiştir: belirli bir ortamda bir robotun küresel konumunu doğru bir şekilde tahmin etmeyle “küresel konumlandırma” odaklanırken, ardışık görüntüler arasındaki göreceli hareketi hesaplamalarıyla da “göreceli konumlandırma” yapılmaktadır.

- **Küresel Konumlandırma:** Küresel konumlandırma, bilinen bir ortam içinde bir robotun mutlak pozisyonunu belirleme sürecidir. Araştırmacıların yöntemleri

sürekli olarak geliştirdiği ve yenilikçi teknikleri entegre ettiği önemli ilerlemeler bu alanda devam etmektedir.

- **Göreceli Konumlandırma:** Robotikte kritik bir problem olan ego-hareket tahmini, bir robotun ardışık kamera görüntülerini kullanarak kademeli hareketini öngörmektedir. Zaman içinde, araştırmacılar ego-hareket tahmininin doğruluğunu ve sağlamlığını artırmak için çeşitli yaklaşımları ve metodolojileri keşfetmişlerdir.

3.3. Derin Öğrenme Teknikleri

Derin öğrenme, büyük veri setlerinden anlamlı örüntüler çıkarmak ve bu örüntüleri analiz etmek amacıyla geliştirilen matematiksel modellerle öne çıkmaktadır. Bu teknikler, bilgisayar sistemlerinin çeşitli alanlarında etkili bir şekilde kullanılmaktadır. Bu bölümde, derin öğrenme tekniklerinin temel prensipleri ve bu tekniklerin tezdeki potansiyel etkileri açıklanmıştır.

3.3.1. Çok Katmanlı Algılayıcı (Multi Layer Perceptron (MLP))

MLP, özellikle doğrusal olmayan veri eşleme ve karmaşık özellik çıkarma görevlerinde temel bir sinir ağı mimarisi olarak bu tezde kullanılmaktadır. Bu bölümde, MLP'lerin temel özellikleri ve tez ile olan ilgisi anlatılmaktadır.

MLP'ler, giriş, gizli ve çıkış katmanlarına düzenlenmiş bir dizi düğme veya nöron içeren çok katmanlı bir mimariye sahip yapay sinir ağı sınıfıdır. Bu ağlar, veri içindeki karmaşık desenleri ve ilişkileri yakalama yetenekleriyle bilinir, bu da onları geniş bir makine öğrenimi görevleri yelpazesi için çok yönlü bir seçenek haline getirir (Haykin, 1994).

Bir MLP'nin temel yapısı üç ana katmandan oluşmaktadır:

- **Giriş Katmanı:** Bu katman, ham verileri veya özellikleri giriş olarak alır.
- **Gizli Katmanlar:** Bu ara katmanlar, çok sayıda nöron içerir ve giriş verilerini doğrusal olmayan aktivasyon fonksiyonları aracılığıyla dönüştürme görevinden sorumludur.
- **Çıkış Katmanı:** Bu katman, nihai tahmin veya sınıflandırma çıkışını sağlar.

Bir MLP içindeki hesaplama matematiksel olarak şu şekilde ifade edilebilir:

x 'in giriş vektörü, h 'nin gizli katman aktivasyonları ve y 'nin çıkış katmanı aktivasyonları olarak adlandırılmasına izin verin. Giriş katmanından gizli katmana olan dönüşüm, Denklem (3.2) ile ifade edilebilir. Bu denklemde, h gizli katmandaki çıkışı temsil eder ve $W_i h$, giriş ve gizli katmanlar arasındaki ağırlıkları ifade eder. b_h gizli katman için bias vektörüdür. Aktivasyon fonksiyonu “ σ ” genellikle bir düzeltilmiş doğrusal ünite (ReLU) veya sigmoid fonksiyonudur.

$$h = \sigma(W_{ih}x + b_h) \quad (3.2)$$

Çıkış katmanı dönüşümü benzer şekilde Denklem (3.3) ile hesaplanır. Burada, y çıkış katmanındaki çıkışı temsil eder ve $W_h o$, gizli ve çıkış katmanları arasındaki ağırlıkları ifade eder. b_o , çıkış katmanı için bias vektörüdür. Aktivasyon fonksiyonu σ kullanılarak çıkış katmanındaki sonuç elde edilir.

$$y = \sigma(W_{ho}x + b_o) \quad (3.3)$$

Bu tez’de MLP’ler, özellik çıkarma, sınıflandırma ve regresyon görevleri de dahil olmak üzere çeşitli amaçlar için kullanılmıştır. Sensör verileri içindeki karmaşık ilişkileri modelleme ve doğrusal olmayan dönüşümleri gerçekleştirme yetenekleri, algoritmaların doğruluğunu ve adaptasyon kabiliyetini artırmak için çok değerlidir.

3.3.2. Evrişimsel Sinir Ağları

Evrişimsel Sinir Ağları (Convolutional Neural Networks (CNN)), özellikle sensör verilerini işleme alanında temel bir rol oynayan bir derin öğrenme bileşeni olarak ortaya çıkmıştır. Bu özel sinir ağları, görüntü analizi, bilgisayarlı görüş ve özellik çıkarma gibi alanlarda devrim niteliğinde bir etki yaratmıştır. CNN’ler, özellikle karmaşık görsel verilerle çalışmak için optimize edilmiş ve bu sayede otomatik öğrenme süreçlerinde büyük başarı elde etmiştir. Bu başarı, özellikle otonom sistemlerin algılama, tanıma ve karar verme süreçlerinde kullanıldığında önemli bir katkı sağlamaktadır (Teuwen ve Moriakov, 2020).

Temelde CNN’ler, giriş verilerinden hiyerarşik özellikleri otomatik olarak öğrenip çıkarmak üzere tasarlanmış birkaç katmandan oluşmaktadır. Bu katmanlar şunları içermektedir:

- **Evrişim Katmanı:** Evrişim katmanı, giriş verileri üzerinde küçük filtreleri kaydırarak yerel desenleri, kenarları ve diğer görsel özellikleri tespit eder. Her bir filtre, evrişim işlemi gerçekleştirir ve özellik haritasını oluşturur. Bu sayede, görüntüdeki önemli özellikler hiyerarşik bir şekilde öğrenilir.
- **Doğrusal Olmayanlık Katmanı:** Evrişim işlemleri genellikle ardından doğrusal olmayan bir aktivasyon fonksiyonu ile geçirilir. Bu katman, modelin daha karmaşık ilişkileri öğrenmesine yardımcı olur. Doğrultulmuş Doğrusal Birim (Rectified Linear Unit (ReLU)), sıkça kullanılan bir aktivasyon fonksiyonudur ve doğrusallık olmayan özellikleri etkili bir şekilde yakalar. ReLU, Bölüm 3.3.3'te daha detaylı anlatılmıştır.
- **Havuzlama Katmanı:** Havuzlama katmanı, özellik haritasının mekansal boyutlarını azaltarak hesaplama maliyetini düşürür ve aynı zamanda özellikleri daha genelleştirir. Maksimum havuzlama gibi teknikler, önemli özellikleri korur ve vurgular.
- **Düzleştirme Katmanı:** Evrişimsel ve havuzlama katmanları tarafından oluşturulan özellik haritasını düzleştirir. Bu, tam bağlantılı katmanlara giriş için uygun bir vektör elde etmeyi sağlar.
- **Tam Bağlantılı Katman:** Tam bağlantılı katmanlar, önceki katmandaki tüm nöronları mevcut katmandaki her nörona bağlar. Bu katmanlar, öğrenilen özellikleri temsil eden geniş bir özellik vektörünü kullanarak sınıflandırma görevini gerçekleştirir.

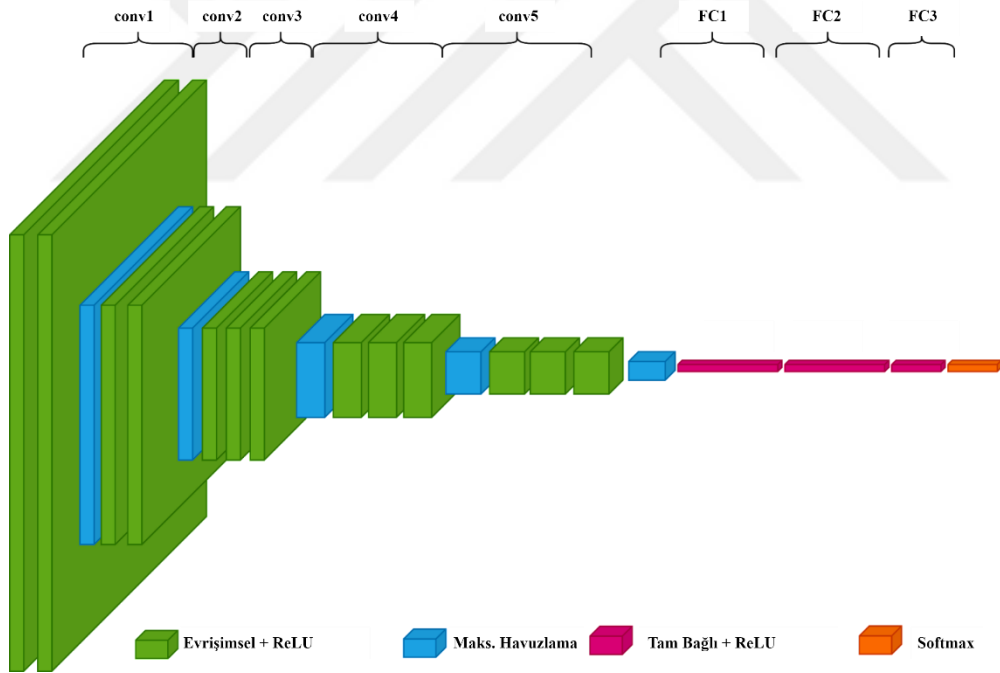
CNN'ler, özellikle görüntüler gibi görsel verilerden özellik çıkarma konusunda üstün performans göstermektedir. Bu tezde, kritik görsel ipuçlarını tanımlamak için görüntüleri analiz etmede kullanılan CNN'ler ile konum tahmini ve konumlandırma doğruluğu artırmaya katkıda bulunulmuştur.

CNN mimarileri, sensör verilerindeki ilişkileri öğrenmeye olanak tanımaktadır. Örneğin, görsel verilere CNN uygulandığında, nesneler, kenarlar, şekiller ve dokular otomatik olarak algılanabilir böylece kesin konumlandırma için bu tanımlamalar kullanılabilir. CNN'ler özellikle görüntü işleme görevleri için daha uygundur. Nesneleri

tespit edebilir, desenleri tanıyabilir ve ilgili detayları vurgulayabilirler. Otonom sistemlerde ise çevreyi algılamayı ve konum tahminini doğru bir şekilde yapma açısından çok değerlidir.

CNN'leri eğitmek, etiketli büyük sensör veri setleri ile nöron ağırlıklarını optimize etmeyi içermektedir. Eğitim sırasında ağ tahminleri ile gerçek değer arasındaki farkı en aza indirmek hedeflenir, böylece konum tahminini doğru bir şekilde yapma yeteneği kazanılır.

Bazı senaryolarda, CNN'ler genel doğruluk ve sağlamlığı artırmak için diğer derin öğrenme mimarileri veya sensör füzyon teknikleri ile birleştirilebilir. Şekil 3.3'de, CNN'deki evrişim katmanlarının konseptini gösteren bir örnek bulunmaktadır, bu katmanların bir giriş görüntüsünden özellikleri nasıl tespit ettiğini ve çıkardığını vurgulamaktadır.



Şekil 3.3. CNN Mimarisi.

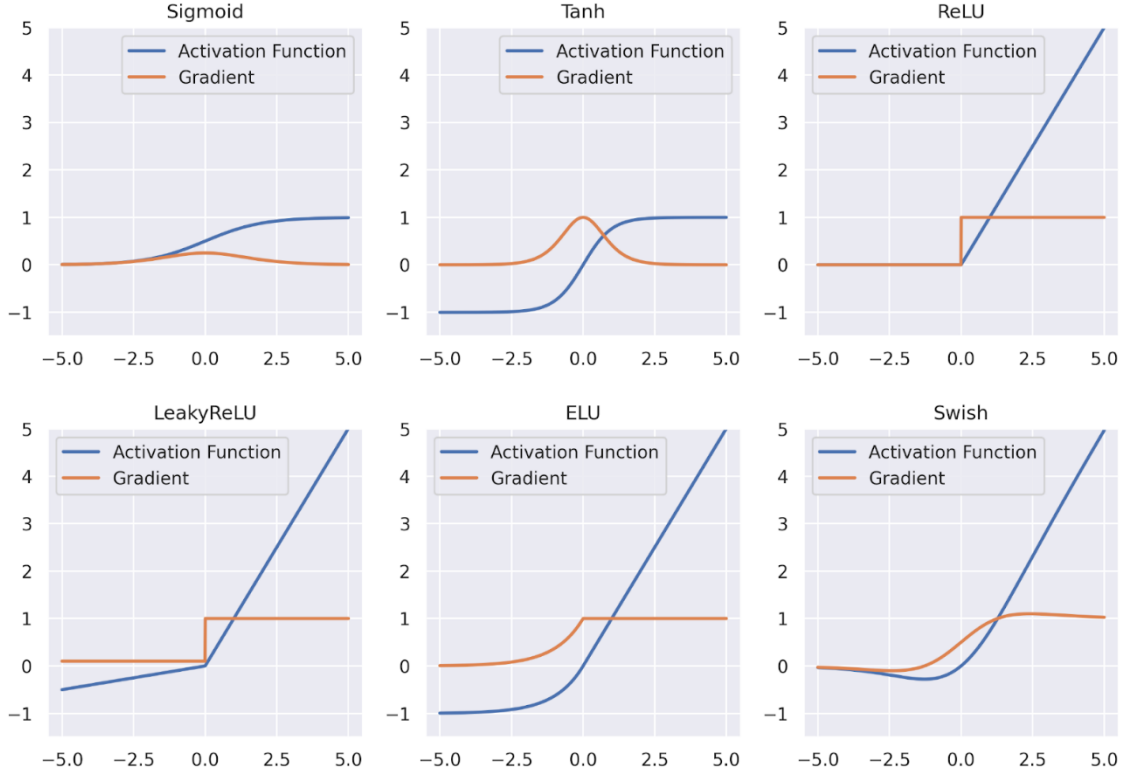
Bu tezde, CNN'ler, özellikle görüntüler gibi sensör verilerini işlemek ve ilgili özellikleri çıkarmak için temel bir araç olarak kullanılmıştır. Otomatik olarak desenleri öğrenme ve tanıma yetenekleri, doğru konumlandırma ve konum tahminiyle ilgili zorluklara önemli katkı sağlamıştır.

3.3.3. Doğrultulmuş Doğrusal Birim (Rectified Linear Unit (ReLU))

ReLU, derin sinir ağlarında, özellikle CNN gibi, kullanılan temel bir aktivasyon fonksiyonudur. Modele doğrusallık dışı bir öğrenme yeteneği kazandırarak, sensör verilerindeki karmaşık ilişkileri öğrenilmesinde önemli bir rol oynamaktadır. ReLU aktivasyon fonksiyonları, bu tez çalışmasında konum tahmini görevleri için sinir ağı mimarilerine entegre edilmiş ve konumlandırma algoritmalarının doğruluğuna katkı sağlamıştır. ReLU aktivasyon fonksiyonunun matematiksel ifadesi Denklem (3.4)'te gösterilmiştir. Bu denklemde, ReLU aktivasyon fonksiyonu, giriş değeri x 'in sıfırdan küçük olması durumunda sıfır, x 'in sıfırdan büyük olması durumunda ise x 'i çıktı olarak vermektedir. Bu sayede, negatif giriş değerlerini sıfıra eşitleyerek doğrusal olmayan özelliklerin öğrenilmesine katkıda bulunmaktadır (Yamashita ve ark., 2018).

$$ReLU(x) = \max(0, x) \quad (3.4)$$

Şekil 3.4, altı aktivasyon fonksiyonunu örnek olarak göstermektedir. Bu şekilde, x-ekseni giriş değerlerini ve y-ekseni de aktivasyon değerlerini göstermektedir. Mavi çizgiler aktivasyon fonksiyonlarını temsil etmektedir; örneğin, sigmoid fonksiyonu 0 ile 1 arasında pürüzsüz bir geçiş yaparken, ReLU parça doğrusallığı sergilemektedir. Diğer yandan, turuncu çizgiler gradyan eğrilerini temsil etmekte olup, giriş değişikliklerine olan hassasiyetlerini gösterir ve farklı fonksiyonlar arasındaki diklik ve düzlük varyasyonlarını ortaya koymaktadır.



Şekil 3.4. Sinir Ağlarındaki Aktivasyon Fonksiyonlarının ve Gradyanların Görselleştirilmesi.

3.3.4. LSTM ve Bi-LSTM (Bi-directional LSTM):

Uzun Kısa Vadeli Hafıza (Long Short-Term Memory (LSTM)), bir tür RNN mimarisi olup, sensör verilerindeki ardışık bağımlılıkları yakalama ve modelleme yeteneği nedeniyle bu tez çalışmasında önemli bir rol oynamaktadır (Gers ve ark., 2002).

RNN'ler, ardışık verileri işleme konusunda iyi olan sinir ağı sınıfıdır, bu da onları zaman serisi tahmini ve doğal dil işleme gibi görevler için çok değerli kılmaktadır. Bununla birlikte, standart RNN'ler kaybolan gradyan probleminden dolayı verideki uzun vadeli bağımlılıkları, yani geçmiş zamanlardaki bilgilerin etkili bir şekilde kullanılmasını, yakalama yeteneklerini sınırlamaktadır.

LSTM ağları, özel hafıza hücreleri ve kapı mekanizmalarını tanıtarak standart RNN'lerin sınırlamalarını ele almaktadır. Bu bileşenler, LSTM'lerin uzun menzilli bağımlılıkları yakalamasını ve uzun diziler boyunca ilgili bilgileri hatırlamasını sağlamaktadır. LSTM'ler birkaç temel bileşeni içermektedir:

- **Hücre Durumu:** LSTM ağlarının merkezinde yer alan kritik bir bileşen olarak öne çıkar. Bu özel hafıza hücresi, bilgiyi zaman adımları boyunca taşıma yeteneği ile adeta bir taşıyıcı bant görevi görür. Yalnızca küçük doğrusal etkileşimlerle tam zincir boyunca düzgün bir şekilde çalışarak, bilgiyi koruma açısından ideal bir bileşen oluşturur. Hücre durumu, geçmişten gelen önemli bilgileri muhafaza ederek, LSTM ağlarının uzun vadeli bağımlılıkları başarıyla ele almasına olanak tanır. Ayrıca, diğer bileşenlerle etkileşim içinde çalışarak, gelen verileri günceller ve yeni bilgilerle birleştirir, böylece modelin güvenilir ve tutarlı tahminler yapmasına katkıda bulunur. Bu sayede, hücre durumu, LSTM ağlarının adaptasyon yeteneklerini artırarak, karmaşık ve dinamik ortamlarda etkili bir şekilde çalışmasını sağlar, özellikle otonom sistemlerin sensör verileri üzerinde başarılı bir performans elde etmesinde kilit bir rol oynar.
- **Gizli Durumu:** Hücre durumuyla birlikte, LSTM ağlarının temel yapı taşlarından biri olan gizli durum, her zaman adımında kritik bir rol oynar. Mevcut girişin, önceki gizli durumun ve hücre durumunun bir fonksiyonu olarak tanımlanan gizli durum, LSTM biriminin çıktısını oluşturur. Bu durum, ağın içsel öğrenme sürecindeki gizli özellikleri temsil eder ve çeşitli zaman adımları arasındaki ilişkileri modellemede etkili bir rol oynar. Gizli durum, geçmiş bilgilerin yeni bilgilerle birleştiği noktada, ağın gelecekteki tahminlerini şekillendirir. Dolayısıyla, gizli durum, LSTM ağlarının genel performansını artıran ve ardışık veriler üzerinde etkili öğrenme sağlayan önemli bir bileşendir.
- **Kapılar:** LSTM'ler, bilgi akışını düzenlemek ve kontrol etmek amacıyla üç ana kapıya sahiptir: giriş kapısı, unutma kapısı ve çıkış kapısı. Bu kapılar, ağın öğrenme sürecini yönlendirir ve hafıza hücresine akışını düzenleyerek bilginin nasıl işlendiğini belirler. Giriş kapısı, mevcut bilgiyi hafıza hücresine ekler; unutma kapısı, gereksiz veya geçersiz bilgileri hafıza hücresinden çıkarır; çıkış kapısı ise hafıza hücresinden gelen bilgiyi ağın çıktısına entegre eder. Bu kapılar, ağın adaptif öğrenme yeteneğini artırarak, hangi bilgilerin önemli olduğunu belirleyen dinamik bir mekanizma sunar. Bu sayede, LSTM ağları, çeşitli uygulamalarda daha esnek ve özelleştirilebilir bir şekilde kullanılabilir hale gelir.

LSTM birimindeki işlemler matematiksel olarak, Denklem (3.3)-(3.8) ile tanımlanabilir. LSTM birimindeki işlemler, bir zaman adımından diğerine geçişleri

modellemek için kullanılan matematiksel ifadeleri içermektedir. f_t , i_t , g_t , ve o_t kapıları, girişlerin ve önceki gizli durumun (l_t) uygun ağırlıklarla çarpılması ve bir sapma terimi eklenmesiyle hesaplanır. Bu kapılar, hücre durumu (c_t) üzerindeki etkilerini kontrol eder. Ayrıca, hücre durumu ve gizli durum, f_t , i_t , g_t değerleri kullanılarak güncellenir. c_t ve h_t sonuçları, ardından çıkış kapısı tarafından kontrol edilen bir tanh fonksiyonundan geçirilir. Bu işlemler, LSTM biriminin iç dinamiklerinin ve bilgi geçişinin modellemesini sağlamaktadır.

LSTM birimindeki matematiksel işlemler, ağın içsel dinamiklerini ve bilgi geçişini ayrıntılı bir şekilde modellemek amacıyla Denklem (3.5)-(3.10) ile tanımlanır. Bu denklemler, bir zaman adımından diğerine geçişlerde gerçekleşen karmaşık işlemleri açıklar. Özellikle, f_t , i_t , g_t ve o_t kapıları, girişlerin ve önceki gizli durumun (l_t) uygun ağırlıklarla çarpılması, bir sapma terimi eklenmesi ve sigmoid aktivasyon fonksiyonuna tabi tutulması yoluyla hesaplanır. Bu kapılar, hücre durumu (c_t) üzerindeki etkilerini kontrol eder. Hücre durumu ve gizli durum güncellemeleri için kullanılan f_t , i_t ve g_t değerleri, çeşitli matematiksel operasyonlarla elde edilir. Hücre durumu, önceki hücre durumunun bir kısmının unutulması (f_t ile çarpılması) ve yeni bilginin eklenmesi (i_t ve g_t ile çarpılması) ile güncellenir. Gizli durum ise, hücre durumu üzerinden elde edilen bilgilerin çıkış kapısı (o_t) tarafından kontrol edilen bir tanh fonksiyonuna tabi tutularak oluşturulur.

Bu süreç, LSTM biriminin içsel mekanizmalarını anlamamıza yardımcı olur. Bu birim, geçmiş bilgileri hatırlayabilen, uzun vadeli bağımlılıkları başarıyla ele alan ve ardışık adımlar arasında bilgi geçişini etkili bir şekilde yöneten bir yapıya sahiptir. Matematiksel ifadeler, bu kompleks yapıyı açıklarken aynı zamanda ağıın öğrenme yeteneklerini optimize etmek ve karmaşık ilişkileri modellemek için kullanılır. Bu detaylı matematiksel anlatım, LSTM'nin içsel işleyişini anlamak isteyenler için önemli bir rehberlik sağlar.

$$f_t = \sigma(W_f \cdot [l_t, x_t] + b_f) \quad (3.5)$$

$$i_t = \sigma(W_i \cdot [l_t, x_t] + b_i) \quad (3.6)$$

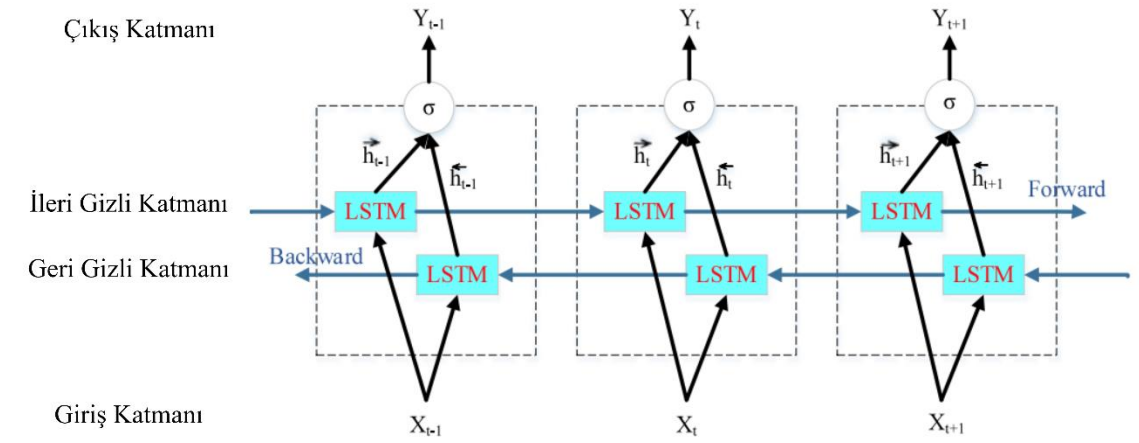
$$g_t = \tanh(W_c \cdot [l_t, x_t] + b_c) \quad (3.7)$$

$$o_t = \sigma(W_o \cdot [l_t, x_t] + b_o) \quad (3.8)$$

$$c_t = f_t \cdot c_{t-1} + i_t \cdot g_t \quad (3.9)$$

$$h_t = o_t \cdot \tanh(c_t) \quad (3.10)$$

Bi-LSTM, LSTM mimarisinin ardışık modelleme yeteneklerini geliştiren bir uzantıdır. Geleneksel LSTM'den farklı olarak, Bi-LSTM, dizileri baştan sona değil, aynı anda hem ileri hem de geri yönlerde işleyen bir mimariye sahiptir. İki LSTM katmanından oluşur: biri diziyi ileri işleyen diğeri ise ters yönde işleyen. Bu çift yönlü işleme, bir veri noktasının yalnızca mevcut bağlamını değil, aynı zamanda sonraki bağlamı da yakalar. Sonuç olarak, Bi-LSTM, doğal dil işleme ve sensör verisi analizi gibi ardışık veri içeren görevler için özellikle kullanışlıdır, burada hem geçmiş hem de gelecekteki gözlemlerden elde edilen bilgilerin doğru tahminler veya sınıflandırmalar yapmak için kritik olduğu durumlar söz konusudur. Bi-LSTM, ardışık veri analizi ve konum tahmini görevlerinin doğruluğunu artırmak için bu tezde değerli bir araçtır. Bi-LSTM ağının mimarisinin ana bileşenleri arasındaki etkileşimi Şekil 3.5'te gösterildiği gibidir.



Şekil 3.5. Bi-LSTM Ağ Mimarisi (Aslan ve ark., 2022).

LSTM ve Bi-LSTM ağları, ardışık sensör verilerini etkili bir şekilde işlemek için bu tez çalışmasında kullanılmıştır. Zamansal bağımlılıkları yakalama yetenekleri, görsel-ataletsel odometri ve sensör füzyonu gibi görevler için idealdir.

3.3.5. Siamese Ağları

Siamese Ağları, benzerlik ölçümü ve sensör verilerinde özellik çıkarma gibi görevlerde önemli bir bileşendir. Bu bölümde, Siamese Ağları'nın ayrıntılarına ve bu tezdeki önemine değinilmektedir.

Siamese Ağları, özellikle iki veri noktası arasındaki benzerlik veya farklılık değerlendirmesi için tasarlanmış sinir ağları kategorisine girmektedir. İki giriş arasındaki benzerlik veya karşıtlık derecesini ayırt etmenin önemli olduğu senaryolarda, özellikle konumlandırma ve sensör verisi hizalaması gibi görevlerde kullanılmaktadır (Koch ve ark., 2015).

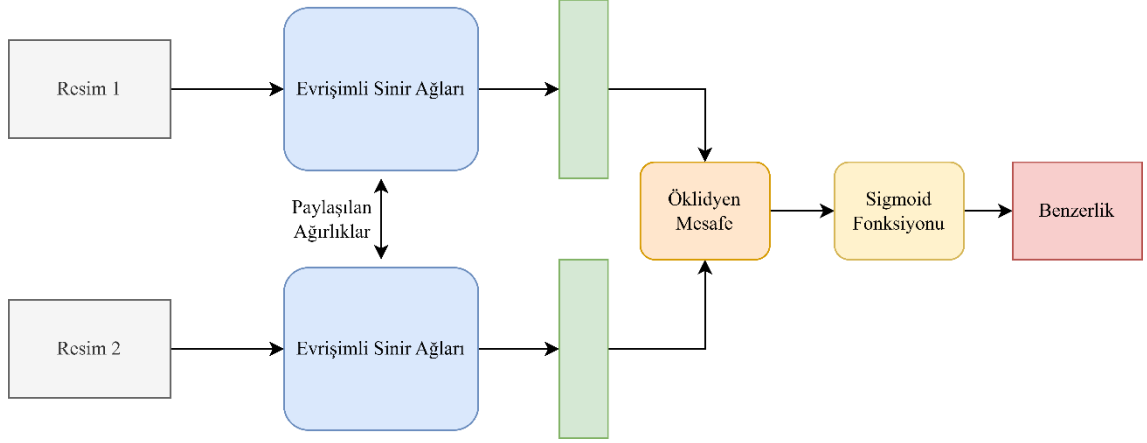
Siamese Ağları'nın temelinde, ağırlıkları ve parametreleri paylaşan iki paralel alt ağın bulunduğu bir mimari bulunmaktadır. Bu paralel alt ağlar, iki giriş veri noktasını bağımsız olarak işler. Daha sonra, bu alt ağlardan gelen çıkışlar, genellikle öklit mesafe veya kosinüs benzerliği gibi yöntemleri kullanarak benzerliklerinin ölçmektedir.

Siamese Ağları, bilinen benzerlik veya farklılık etiketlerini taşıyan veri noktaları çiftleriyle eğitime tabi tutulur. Eğitim süreci boyunca, benzer çiftler arasındaki mesafe en aza indirilirken farklı çiftler arasındaki mesafe maksimize edilerek ağ'daki öğrenme gerçekleşir. Bu eğitim programı, Siamese Ağlarının, veri içindeki temel benzerlik yapısını ustaca ayırt etme ve nicelendirme yeteneği ile başarılıdır.

Siamese Ağları'nın temel matematiksel formülasyonu, iki paralel alt ağın çıkışları arasındaki mesafeyi (veya farklılığı) hesaplamayı içerir. Bu amaçla, alt ağlardan çıkarılan x_1 ve x_2 özellik vektörleri arasındaki öklit mesafe genellikle kullanılır. Bu durum, Denklem (3.11) ile ifade edilir. Bu denklemde, d çıkarılan özellik vektörleri arasındaki hesaplanmış mesafeyi temsil eder. x_1 ve x_2 , paralel alt ağlardan çıkarılan özellik vektörlerini ifade eder. n ise, örnek sayısını ifade etmektedir.

$$d = \sqrt{\sum_{i=1}^n (x_{1i} - x_{2i})^2} \quad (3.11)$$

Şekil 3.6, bir Siamese Ağ'ının mimarisini gösterir ve ikiz alt ağları ile mesafe hesaplama sürecini anlatır:



Şekil 3.6. Siamese Ağ Mimarisi.

3.3.6. Rastgele PCA (Principal Component Analysis) Artırma

Rastgele PCA Artırma (Random PCA Augmentation), bu tez çalışmasında başlıca kullanılan bir teknik olarak öne çıkar. Özellikle sensör verilerinde veri artırma ve boyut indirgeme için kullanılır. Bu bölümde, Rastgele PCA Artırma'nın inceliklerine ve tezdeki önemine değinilmektedir (Pearson ve ark., 1901).

Rastgele PCA Artırma, yüksek boyutlu sensör verileriyle karşılaşıldığında özellikle makine öğrenme modellerinin dayanıklılığını artırmayı amaçlayan bir veri ön işleme tekniğidir. PCA, verinin boyutunu azaltırken temel bilgilerini korur. Bu teknik, veri boyutunu hafifletme ve model genelleştirme konusunda önemli bir yöntemdir.

Bu tezde, rastgele PCA artırma yöntemi görüntü verisine aşağıdaki amaçlarla uygulanmaktadır:

- **Veri Artırma:** Mevcut görüntülerin rastgele PCA projeksiyonları aracılığıyla varyasyonları oluşturularak veri seti genişletilir ve özellikle sınırlı etiketli veri senaryolarında makine öğrenme modellerinin performansı artırılır.
- **Özellik Geliştirme:** Görüntülerden önemli özellikler çıkarmada yardımcı olur ve modelin ilgili bilgileri yakalama yeteneğini artırabilir.

3.3.7. Hata Fonksiyonu

Hata Fonksiyonu, derin öğrenme modellerinin temel bir bileşeni olup, modelin eğitim hedefleri üzerinde ne kadar iyi performans gösterdiğini ölçmektedir. Bu bölümde, Hata Fonksiyonu kavramına, model eğitimindeki rolüne ve tezdeki önemi hakkında bilgi verilmektedir.

Derin öğrenmede Hata Fonksiyonu, model tarafından üretilen tahmin değerleri ile eğitim verilerindeki gerçek değerler arasındaki farkı vermektedir. Böylece modelin tahminlerinin istenen sonuçlardan ne kadar sapma gösterdi bulunur.

Derin öğrenmede Hata Fonksiyonunun temel amaçları şunlardır (Janocha ve ark., 2017):

- Optimizasyon: Hata Fonksiyonu, SGD (Stochastic Gradient Descent) veya Adam gibi optimizasyon algoritmaları için amaç fonksiyonu olarak kullanılır. Bu fonksiyonu en aza indirme amacıyla uygulanan optimizasyonlar ile, modelin doğru tahminler yapma yeteneği iyileştirilir.
- Model Değerlendirmesi: Eğitim sırasında kullanılan modelin performansını değerlendirmek için yapılır. Düşük hata değerleri, genellikle tahminler ile gerçek değerler arasındaki daha iyi uyum olduğunu gösterir.

Farklı türde Hata Fonksiyonları bulunmaktadır, bunlar farklı türdeki makine öğrenme görevleri için tercih edilmektedir. Yaygın Hata Fonksiyonları şunlardır:

- MSE (Mean Squared Error): Regresyon görevleri için kullanılan MSE, tahmin edilen ve gerçek değerler arasındaki farkların karelerinin ortalamasını hesaplar.
- Çapraz-Entropi Kaybı: Sınıflandırma problemlerinde uygulanan bu yöntem, tahmin edilen sınıf olasılıkları ile gerçek sınıf etiketleri arasındaki benzerliği nicelendirir.

- İkili Capraz Entropi (Binary Cross-Entropy) Kaybı: Özel olarak ikili sınıflandırma görevleri için tasarlanmıştır ve tahmin edilen olasılıklar ile ikili etiketler arasındaki farkı ölçer.
- Kategorik Çapraz-Entropi Kaybı: Çoklu sınıf sınıflandırması için uygun olan bu hata fonksiyonu, tahmin edilen sınıf olasılıkları ile gerçek sınıf etiketleri arasındaki sapmayı değerlendirir.

Hatanın Geri Yayılımı (Backpropagation), derin öğrenme modellerinin eğitimi sırasında kullanılan temel bir algoritmadır. Bu algoritma, bir ağın ürettiği tahmin değerleri ile gerçek değerler arasındaki hatayı azaltmak amacıyla ağı güncellemektedir. Hatanın Geri Yayılımı Algoritması'nın adımları aşağıdaki gibidir:

1. İleri Yayılım (Forward Propagation): İlk olarak, ağın girişine bir veri örneği uygulanmaktadır. Bu veri örneği, ağın her bir katmanındaki nöronlarda aktivasyon değerlerini üretir ve sonunda çıkış katmanında bir tahmin elde edilir. Bu adım, giriş verisinden başlayarak ağı ileriye doğru geçmeyi içerir. Bu süreç matematiksel olarak Denklem (3.12) ile ifade edilebilir:

$$a^{(l)} = \sigma(W^{(l)}a^{(l-1)} + b^{(l)}) \quad (3.12)$$

burada $a^{(l)}$ katman l 'deki aktivasyon vektörünü, $W^{(l)}$ ve $b^{(l)}$ sırasıyla ağırlık ve bias matrislerini, σ aktivasyon fonksiyonunu temsil etmektedir.

2. Hata Hesaplama (Error Calculation): Elde edilen tahmin değerleri ile gerçek değerler arasındaki hata hesaplanmaktadır. Hata genellikle bir kayıp fonksiyonu kullanılarak ölçülür, örneğin, regresyon problemleri için Mean Squared Error (MSE) veya sınıflandırma problemleri için Çapraz-Entropi Kaybı. Bu süreç matematiksel olarak Denklem (3.13) ile ifade edilebilir:

$$L = \frac{1}{2m} \sum_{i=1}^m \|y^{(i)} - \hat{y}^{(i)}\|^2 \quad (3.13)$$

burada m veri örneklerinin sayısını, $y^{(i)}$ gerçek değeri, $\hat{y}^{(i)}$ ise modelin tahminini temsil etmektedir.

3. Geri Yayılım (Backward Propagation): Hatanın geri yayılması adımdır. Bu adımda, ağın çıkışındaki hatanın katmanlara doğru nasıl geri yayılacağı hesaplanmaktadır. Bu

hesaplama, zincir kuralı kullanılarak gerçekleştirilmektedir. Her katmanın ağırlıkları ve biasları, hatanın kendisi ve aktivasyon fonksiyonunun türevi kullanılarak güncellenmektedir. Zincir kuralını kullanarak, her katmanın ağırlıkları $W^{(l)}$ ve biasları $b^{(l)}$ Denklem (3.14) ve (3.15) ile güncellenmektedir:

$$\frac{\partial L}{\partial W^{(l)}} = \frac{\partial L}{\partial a^{(l)}} \cdot \frac{\partial a^{(l)}}{\partial z^{(l)}} \cdot \frac{\partial z^{(l)}}{\partial W^{(l)}} \quad (3.14)$$

$$\frac{\partial L}{\partial b^{(l)}} = \frac{\partial L}{\partial a^{(l)}} \cdot \frac{\partial a^{(l)}}{\partial z^{(l)}} \cdot \frac{\partial z^{(l)}}{\partial b^{(l)}} \quad (3.15)$$

burada $z^{(l)}$ aktivasyon öncesindeki ağırlıklı girişi ifade etmektedir.

4. Ağırlık ve Bias Güncelleme (Weight and Bias Update): Geri yayılma sonucunda elde edilen hata bilgisi kullanılarak, ağırlıklar ve biaslar güncellenir. Bu, optimizasyon algoritmaları (örneğin, Stochastic Gradient Descent - SGD veya daha gelişmiş algoritmalar) tarafından belirlenen bir öğrenme oranı ile gerçekleştirilmektedir. Bu süreç matematiksel olarak Denklem (3.16) ve (3.17) ile ifade edilebilir:

$$W^{(l)} = W^{(l)} - \alpha \frac{\partial L}{\partial W^{(l)}} \quad (3.16)$$

$$b^{(l)} = b^{(l)} - \alpha \frac{\partial L}{\partial b^{(l)}} \quad (3.17)$$

5. Tekrarlama (Iteration): Bu üç adım (ileri yayılım, hata hesaplama, geri yayılım) bir veri örneği için tamamlandıktan sonra, aynı adımlar diğer veri örnekleri için tekrarlanır. Bu işlem, ağın genel performansını iyileştirmek için birden çok epoch (eğitim iterasyonu) boyunca devam etmektedir.

Bu tez çalışmasında Hata Fonksiyonu seçimi kritiktir ve özel makine öğrenme görevlerine uyarlanmaktadır. Hata Fonksiyonu, görsel sınıflandırma, sensör füzyonu ve görsel-ataletsel odometri gibi görevler için modelleri eğitmede kilit bir rol oynamaktadır.

3.4. Sensor Füzyonu ve IMU İşleme

Sensor füzyonu ve IMU işleme, otonom sistemlerin doğru konumlandırmasında kritik bir rol oynamaktadır. Çeşitli sensör bilgilerini entegre etme ve IMU verilerini etkili bir şekilde işleme üzerine odaklanarak, otonom sistemlerin karmaşık çevrelerde konumlandırma performansını artırmaktadır.

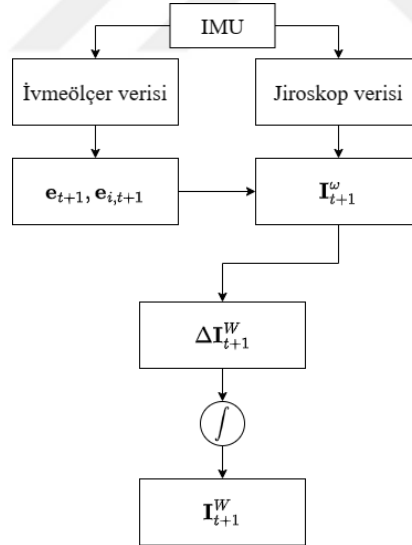
3.4.1. IMU Verisi İşleme:

IMU'lar, otonom sistemler alanında kritik sensörlerdir ve hareket ile rotasyon hakkında temel bilgiler sağlar. Bu bölümde, IMU verisinin önemini, içerdiği işleme tekniklerini ve otonom sistemlerin yeteneklerini artırmadaki rolünü incelemekteyiz.

Bir IMU genellikle doğrusal ivme ölçen ivmeölçerler ve açısal hızı ölçen jiroskopları içerir. Bu sensörler, insansız hava araçlar ve insansız kara araçlar gibi otonom sistemlere entegre edilir ve hareket ve rotasyonları hakkında gerçek zamanlı kesin bilgiler elde etmek için kullanılır.

3.4.1.1. Mahony Filtresi

Mahony filtresi, jiroskop ölçümlerini ivmeölçer ölçümleri ile birleştirerek konum, açı ve rotasyonu tahmin eden bir tamamlayıcı filtre türüdür. Şekil 3.7, Mahony filtresinin blok diyagramını göstermekte olup, diyagram üzerindeki değişkenlerin tanımlamaları aşağıda verilmiştir (Mahony ve ark., 2008).



Şekil 3.7. Mahony filtresi blok şeması.

Bunun için, önce ivmeölçer ölçümlerini kullanarak önceki tahminlerden konum hatasını Denklem (3.18) ve (3.19) ile hesaplar.

$$e_{t+1} = I_{t+1}^{\hat{a}} \times v(I_t^q) \quad (3.18)$$

$$e_{i,t+1} = e_{i,t} + e_{t+1} \Delta t \quad (3.19)$$

burada $I_{t+1}^{\hat{a}}$ ifadesi normalleştirilmiş ivmeölçer ölçümünü, $v(I_t^q)$ ifadesi kuaterniyon cinsinden t anındaki rotasyonu, t ise t ile $t + 1$ arasındaki zaman damgasını gösterir. Sonraki adım, t ile $t + 1$ arasındaki zamanda artan rotasyonu birleştirmek ve tahmin etmektir (Denklem (3.20) – (3.22)).

$$I_{t+1}^{\omega} = I_{t+1}^{\omega} + K_p e_{t+1} + K_i e_{i,t+1} \quad (3.20)$$

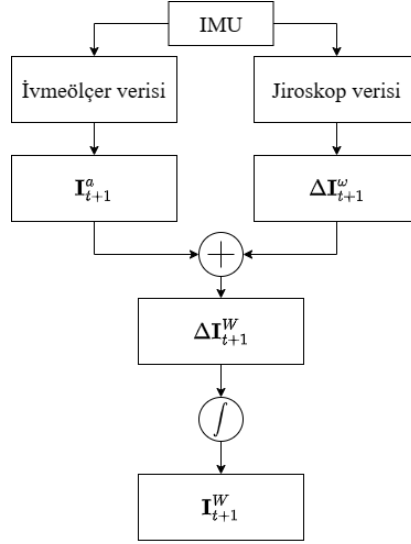
$$\Delta I_{t+1}^W = \frac{1}{2} I_t^{\hat{W}} \otimes [i, I_{t+1}^{\omega}]^T \quad (3.21)$$

$$I_{t+1}^W = I_t^{\hat{W}} + \Delta I_{t+1}^W \Delta t \quad (3.22)$$

burada, I_{t+1}^{ω} , birleştirmenin ardından güncellenmiş jiroskopu temsil eder, ΔI_{t+1}^W , jiroskop ölçümlerinden gelen rotasyon artışıdır ve I_{t+1}^W , rotasyon artışının entegrasyonu sonucu tahmin edilen rotasyonu göstermektedir.

3.4.1.2. Madgwick Filtresi

Madgwick filtresi, rotasyon tahminini bir kuaterniyon alanında formüle etmektedir. Buradaki temel fikir, jiroskop ve ivmeölçerler tarafından ölçülen açısal hız ve ivme ölçümlerini birleştirerek I_{t+1}^W 'yi tahmin etmektir. Şekil 3.8, Madgwick filtresinin blok şemasını gösterir ve şema üzerindeki parametrelerin açıklamaları aşağıda sunulmuştur (Madgwick, 2010).



Şekil 3.8. Madgwick filtresi blok şeması.

İlk adım, ivmeölçer ölçümlerinden rotasyon artışını hesaplayan bir gradyan adımıdır (Denklem (3.23)).

$$I_{t+1}^a = - \frac{\text{argmin } f(I_t^{\hat{W}}, W_{\hat{g}}, I_{t+1}^{\hat{a}})}{\|f(I_t^{\hat{W}}, W_{\hat{g}}, I_{t+1}^{\hat{a}})\|} \quad (3.23)$$

burada I_{t+1}^a , ivmeölçer ölçümlerinden gelen durum bilgisini, $I_t^{\hat{W}}$, bir kuaterniyon rotasyonunu, $W_{\hat{g}}$, normalize edilmiş yerçekimi ve $I_{t+1}^{\hat{a}}$, normalize edilmiş ivmeölçer ölçümlerini temsil etmektedir. İkinci adım, jiroskopdan rotasyon artışını hesaplamaktır ve son olarak ivmeölçer ölçümleriyle birleştirilerek rotasyonu tahmin etmektedir (Denklem (3.24) – (3.26)).

$$\Delta I_{t+1}^{\omega} = \frac{1}{2} I_t^{\hat{W}} \otimes [0, I_{t+1}^{\omega}]^T \quad (3.24)$$

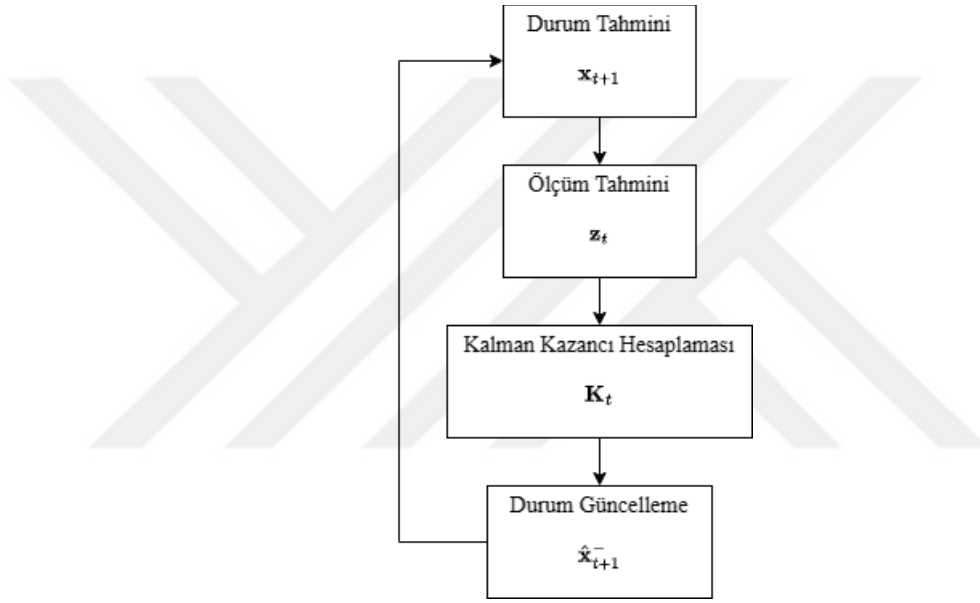
$$\Delta I_{t+1}^W = \Delta I_{t+1}^{\omega} + I_{t+1}^a \quad (3.25)$$

$$I_{t+1}^W = I_t^{\hat{W}} + \Delta I_{t+1}^W \Delta t \quad (3.26)$$

burada, ΔI_{t+1}^{ω} , jiroskop rotasyon artışını; ΔI_{t+1}^W , jiroskop ve ivmeölçer rotasyon artışının birleşimini; I_{t+1}^W ise tahmini rotasyonu temsil etmektedir.

3.4.1.3. EKF Filtresi

Gerçek sistem dinamikleri ve gözlem modelleri, genellikle doğrusal değildir. Ancak, yerel olarak iyi bir şekilde doğrusal fonksiyonlara yaklaşabilirler. Bu fonksiyonları doğrusallaştırmak için Taylor Genişlemesi ve Jacobianlar kullanılabilir. Bu sayede EKF sistemi üzerinde doğrusallaştırma işlemi gerçekleştirilebilir. Şekil 3.9, EKF blok şemasını gösterir ve şema üzerindeki parametrelerin açıklamaları aşağıda sunulmuştur (Hartikainen ve ark., 2011).



Şekil 3.9. EKF filtresi blok şeması.

Burada, önceki $p(x_0)$ bir Gaus dağılımıdır (yani, $p(x_0) \sim N(\mu_0, \Sigma_0)$) burada $x = [q_w, q_x, q_y, q_z, \omega_x, \omega_y, \omega_z]$, yedi durumlu bir durum vektörüdür. Süreç (Process) modeli $\hat{x} = f(x, u, n)$, doğrusal olmayan ve Gaus gürültü Q_t olan bir modeldir. Gözlem modeli de $h(x, v)$ doğrusal olmayan bir modeldir ve Gaus gürültü R_t içermektedir.

Dinamik modelin ve gözlem modelinin doğrusallaştırması Denklem (3.27) ve (3.28) ile yapılır:

$$x_{t+1} \approx f(x_t, t) + Q_t \quad (3.27)$$

$$z_t \approx h(x_t, t) + R_t \quad (3.28)$$

Durum vektörü ve durum kovaryans matrislerini başlattıktan sonra, Kalman kazanç matrisi Denklem (3.29) ile hesaplanır:

$$K_t = P_t^- H_t^T [H_t P_t^- H_t^T + R_t]^{-1} \quad (3.29)$$

burada, $H_t \approx \frac{\partial h}{\partial x} |_{x=\hat{x}_t^-}$. Sonraki adım, Denklem (3.30)'ü kullanarak durum düzeltme vektörünü hesaplamak ve durum vektörünü güncellemektir.

$$\hat{x}_t = \hat{x}_t^- + K_t [z_t - \hat{z}_t^-], \text{ with } \hat{z}_t^- = h(\hat{x}_t^-, t) \quad (3.30)$$

ve hata kovaryansı Denklem (3.31) ile güncellenir:

$$P_t = [I - K_t H_t] P_t^- \quad (3.31)$$

Yeni durum vektörü ve durum kovaryans vektör matrisinin tahmini Denklem (3.32) ve Denklem (3.33) ile yapılır:

$$\hat{x}_{t+1}^- = f(\hat{x}_t, t) \quad (3.32)$$

$$P_{t+1}^- = \Phi_t P_t \Phi_t^T + Q_t \quad (3.33)$$

burada $\Phi_t = \frac{\partial f}{\partial x} |_{x=\hat{x}_t^-}$.

3.4.2. Çoklu Adım Düzeltme Filtresi (Multi-Step Correction Filter):

GNSS çoklu yol hataları (multipath), sadece doğrudan uydu iletimi aracılığıyla alınan sinyallerden değil, aynı zamanda yakındaki nesnelerden yansımalar yoluyla da elde edilmektedir. Çoklu yol sinyali, doğrudan sinyalden daha uzun bir mesafe kat ettiği için dolayı, pozisyon doğruluğunu etkileyen bir ölçüm hatasına neden olmaktadır. Özellikle ağaçlar ve binalar gibi nesnelerden kaynaklanan kısa vadeli GNSS kesintilerinin ardından, GNSS sinyalleri tekrar erişilebilir olduğunda önemli boşluklar ortaya çıkmaktadır. Alıcıya yakın binalar, ağaçlar ve diğer nesnelerin varlığı, alıcıya farklı yollardan gelen birden fazla sinyalin ulaşmasına neden olarak ölçüm hatasını artırabilir. Çoklu yol etkisi ve bu etki sonucunda oluşan ölçüm hatasını etkileyebilecek faktörler arasında alıcı anteni yerleşimi, sinyal gücü ve atmosfer koşulları yer almaktadır. Bu faktörler, önerilen yaklaşımın etkinliğini minimize etmeyi amaçlamaktadır.

Tez çalışmasında, önerilen yaklaşım, çoklul yol iletiminden kaynaklanan ölçüm hatalarını en aza indirgeyerek genel konumlandırma sisteminin doğruluğunu artırmayı hedeflemektedir. Bu hedefe ulaşmak için, konum tahmincisinin çoğul yol etkilerine karşı dirençli olup olmadığını değerlendirmek amacıyla çeşitli testler uygulanmaktadır. Bu testler, GNSS sinyallerine özgü olarak uygulanmakta, gürültüleri filtreleyerek daha kesin sonuçlar elde etmeyi amaçlamaktadır. Bu şekilde, çoklu yol etkilerine dayalı hatalarla başa çıkma yeteneğini geliştirmek ve konum tahmincisinin genel doğruluğunu artırmak için önerilen yaklaşımın etkili bir strateji olduğu doğrulanmaktadır.

- Hareketsizlik Filtresi: Bu filtrede, GNSS sinyalleri ile aracın hareket durumu kontrol edilmektedir. GD-VIO (Generalizable Deep Visual Inertial Odometry) çıkışlarından elde edilen hız bilgileri kullanılarak bu durum doğrulanmaktadır. GNSS ölçümleri, araç hareket etmiyorsa sürekli bir konum üretmez; bu durum GD-VIO çıkışları tarafından belirlenebilir. Hareketsizlik filtresi, aracın sabit bir konumda olduğunu sağlamak için uygulanır ve bu durumu Denklem (3.34) tarafından ifade edilen kısıtlamalarla güvence altına alır. Bu sayede, GNSS ölçümlerinden kaynaklanan hataların önlenmesi ve genel konumlandırma doğruluğunun artırılması hedeflenir.

$$\begin{cases} \text{eğer } V_{GD-VIO} \neq 0 \text{ and } V_{GNSS} \neq 0, & \text{kabul edilir.} \\ \text{diğer durumlarda,} & \text{kabul edilmez.} \end{cases} \quad (3.34)$$

- Alıcıya Uydu Görünürlüğü: Bu filtrede, GNSS sinyallerinin kalitesi kontrol edilir. Alıcıya dört veya daha fazla uydu görünüyorsa, GNSS bilgileri kabul edilir (Denklem (3.35)).

$$\{\text{eğer görünür uydu sayısı} \geq 4, \quad \text{kabul edilir.} \quad (3.35)$$

- Yatay/Dikey Düşülme Filtresi: Bu filtrede, GNSS ölçümlerinin doğruluğu kontrol edilir. Ölçümler, yatay veya dikey doğruluk düşmesi belirli bir eşik değerinden büyükse kabul edilir (Denklem (3.36)).

$$\begin{cases} \text{eğer Yatay doğruluk düşümü} > \text{eşik değeri,} \\ \text{eğer Dikey doğruluk düşümü} > \text{eşik değeri,} \end{cases} \quad \text{kabul edilir.} \quad (3.36)$$

- Konum Değişikliği Doğrulama Filtresi: Bu filtrede, araç böyle ani konum değişiklikleri yapamayacağından sıçramalı GNSS ölçümleri filtrelenir. Konum değişikliği $p = p_t - p_{t-1}$ eşik değerinden büyükse, GNSS ölçümleri reddedilir. Bu,

enlem, boylam ve yükseklik eksenlerindeki konum değışikliklerini içerir (Denklem (3.37)).

$$\{eğer P_{değişiklik} > eşik değeri, \quad kabul edilir. \quad (3.37)$$

- EKF Filtresi: Son olarak, hem GNSS ölçümleri hem de GD-VIO çıkışları bir EKF kullanılarak filtrelenir. Uygun çıkış ölçümleri daha sonra GNSS/GD-VIO/IMU UKF birleştirme güncelleme adımında kullanılır.

Bu birleştirme filtresi için durum-uzay denklemi, metinde açıklanan çeşitli filtreleme adımlarından (Hareketsizlik Filtresi, Alıcıya Görünür Uydu Sayısı, Yatay/Dikey Doğruluk Filtresi, Konum Değişikliği Onay Filtresi ve EKF gibi) çıkarılabilir. EKF, özellikle doğrusal olmayan sistemler için uygun olan yaygın olarak kullanılan bir tahmin algoritmasıdır. EKF, mevcut tahmin etrafında sistem modelini doğrusallaştırır, bu da doğrusal bir zamanla değişen modeli ortaya çıkarır ve standart Kalman filtre teknikleri kullanılarak çözülebilir. Bu adımlar, GNSS ve GD-VIO'dan gelen ölçümlerin UKF birleştirme güncelleme adımında kullanılmadan önce tutarlı ve doğru olduğundan emin olmak için kullanılır. UKF'deki güncelleme adımı, aslında çok adımlı düzeltme filtresinin çıkışından gelir. Durum uzay modelinde rotasyonu temsil etmek için bir çarpan gösterimi kullanılarak tahmin adımı ve güncelleme adımı değişkenleri arasındaki tutarsızlıkla başa çıkılır ve böylece, IMU'dan gelen rotasyon bilgisi GNSS ve GD-VIO'dan gelen konum ve hız bilgisiyle birleştirilerek çoklu adım düzeltme sağlanır. UKF'nin çıkışındaki filtrelenmiş durumlar, daha sonra çarpan gösteriminden düzeltilmiş olarak elde edilir.

3.4.3. Kokusuz Kalman Filtresi (Unscented Kalman Filter)

Kokusuz Kalman Filtresi (UKF), otonom sistemlerdeki sensör birleştirme ve konumlama görevlerinde kullanılan güçlü bir tahmin tekniğidir. EKF aksine, UKF, doğrusal olmayan fonksiyonların doğrusallaştırılmasına dayanmaz, bu da onu doğrusal olmayan dinamiklere ve doğrusal olmayan Gaus olmayan gürültülü karmaşık sistemler için özellikle uygun kılar.

UKF algoritmasının temel adımları, durum ve kovaryans tahminlerini öngörme ve ardından sensör ölçümlerine dayalı olarak bunları güncelleme içerir. Şekil 3.10, UKF

blok şemasını gösterir ve şema üzerindeki parametrelerin açıklamaları aşağıda sunulmuştur (Wan ve ark., 2000).



Şekil 3.10. UKF filtresi blok şeması.

1. Tahmin Adımı

Bu adımda durum $\hat{\mathbf{x}}_k^-$ ve kovaryans $\mathbf{P}_k^-$ tahmini Denklem (3.38) ve Denklem (3.39) ile yapılır. Bu denklemlerde, f fonksiyonu sistem dinamiklerini, u_k kontrol girişini, Q_k işlem gürültüsü kovaryansını ve A_k durum geçiş matrisini temsil eder.

$$\hat{\mathbf{x}}_k^- = f(\hat{\mathbf{x}}_{k-1}, u_k) + \mathcal{N}(0, Q_k) \quad (3.38)$$

$$\mathbf{P}_k^- = A_k \mathbf{P}_{k-1} A_k^T + Q_k \quad (3.39)$$

2. Sigma Noktası Üretimi:

UKF, doğrusal olmayan dağılımın istatistiksel özelliklerini yakalamak için sigma noktalarını kullanır. Sigma noktaları, önceden tahmin edilen durum ve kovaryans tahminlerinden üretilir. Bu adımda sigma noktalarının tahmini Denklem (3.40) ile yapılır. Bu denklemde, n durum vektörünün boyutudur ve λ bir ölçekleme parametresidir.

$$X_k^- = \begin{bmatrix} x_k^- & x_k^- + \sqrt{(n+\lambda)P_k^-} & x_k^- - \sqrt{(n+\lambda)P_k^-} \end{bmatrix} \quad (3.40)$$

3. Ölçüm Tahmini:

Bu adımda, ölçüm $\hat{z}_k$ tahmini Denklem (3.41) ve sigma noktası ölçüm Z_k tahmini Denklem (3.42) ile yapılır. Burada, h ölçüm fonksiyonunu ve R_k ölçüm gürültüsü kovaryansını temsil eder.

$$\hat{z}_k = h(\hat{x}_k^-) + \mathcal{N}(0, R_k) \quad (3.41)$$

$$Z_k = \begin{bmatrix} z_k & z_k + \sqrt{(n+\lambda)P_k^-} & z_k - \sqrt{(n+\lambda)P_k^-} \end{bmatrix} \quad (3.42)$$

4. Güncelleme Adımı:

Bu adımda çapraz-kovaryans matrisi P_{xz} , yenilik kovaryansı S_k , kalman kazancı K_k , durum $\hat{x}_k$ ve kovaryans P_k güncellemeleri Denklem (3.43)-(3.47) ile yapılır.

$$P_{xz} = \sum_{i=1}^{2n+1} W_i^{(x,z)} (X_i^- - \hat{x}_k^-)(Z_i - \hat{z}_k)^T \quad (3.43)$$

$$S_k = \sum_{i=1}^{2n+1} W_i^{(z)} (Z_i - \hat{z}_k)(Z_i - \hat{z}_k)^T + R_k \quad (3.44)$$

$$K_k = P_{xz} S_k^{-1} \quad (3.45)$$

$$\hat{x}_k = \hat{x}_k^- + K_k (z_k - \hat{z}_k) \quad (3.46)$$

$$P_k = P_k^- - K_k S_k K_k^T \quad (3.47)$$

UKF, otonom sistemlerde sensör birleştirme ve konumlandırma görevleri için sağlam ve doğru sonuçlar vermektedir. Sigma noktalarını kullanarak ve doğrusallaştırma hatalarından kaçınarak, doğrusal olmayan dinamikler ve gaus olmayan gürültü ile başa çıkma konusunda mükemmel bir performans sergilemektedir. Ayrıca UKF, karmaşık gerçek dünya senaryolarında yüksek hassasiyetli konumlandırma ve sensör birleştirme için temel bir araç haline gelmiştir.

Bu bölümde, otonom sistemlerdeki konumlandırma problemleri ele alınmış ve çeşitli derin öğrenme teknikleriyle sensör füzyonunu incelemesi verilmiştir. Konum

tahmini için temel kavramlar, derin öğrenme yöntemleri, sensör füzyonu ve IMU işleme teknikleri detaylı bir şekilde incelenmiştir. Bu bağlamda, CNN, ReLU, LSTM, Bi-LSTM, Siamese ağları, MLP, rastgele PCA artırma ve hata fonksiyonları gibi derin öğrenme teknikleri üzerinde durulmuş; aynı zamanda IMU verilerini işleme yöntemleri olan Mahony, Madgwick, EKF ve UKF gibi yöntemler detaylı bir şekilde değerlendirilmiştir.

Ayrıca, çoklu adım düzeltme filtresi ve UKF gibi sensör füzyonu tekniklerine odaklanılmıştır. Bu kapsamlı metodoloji, otonom sistemlerin konumlandırma performansını artırmak ve çeşitli zorluklarla başa çıkmak için özgün bir yaklaşım sunmayı amaçlamaktadır.



4. OTONOM KONUMLANDIRMA UYGULAMALARI

Bu bölümde bu tez kapsamında uygulanan iki farklı uygulama, bu uygulamalarda kullanılan veri setleri ve uygulama sonuçları anlatılmıştır.

4.1 Uygulamarda Kullanılan Veri Setleri

Tezin kapsamında yer alan iki konumlandırma uygulaması için seçilen veri setleri, araştırmanın temel dayanaklarından biridir. Veri setlerinin seçimi, uygulamaların doğruluğunu ve performansını değerlendirmede kritik bir rol oynamaktadır. Veri setlerinin seçim kriterleri, uygulama bağlamında nasıl kullanıldığı ve elde edilen sonuçların bu veri setlerine dayandığı detaylı bir şekilde açıklanmıştır. Bu analizler, uygulamaların başarıları ve zorlukları hakkında daha fazla içgörü sağlayarak, tezin genel katkılarına ışık tutmuştur.

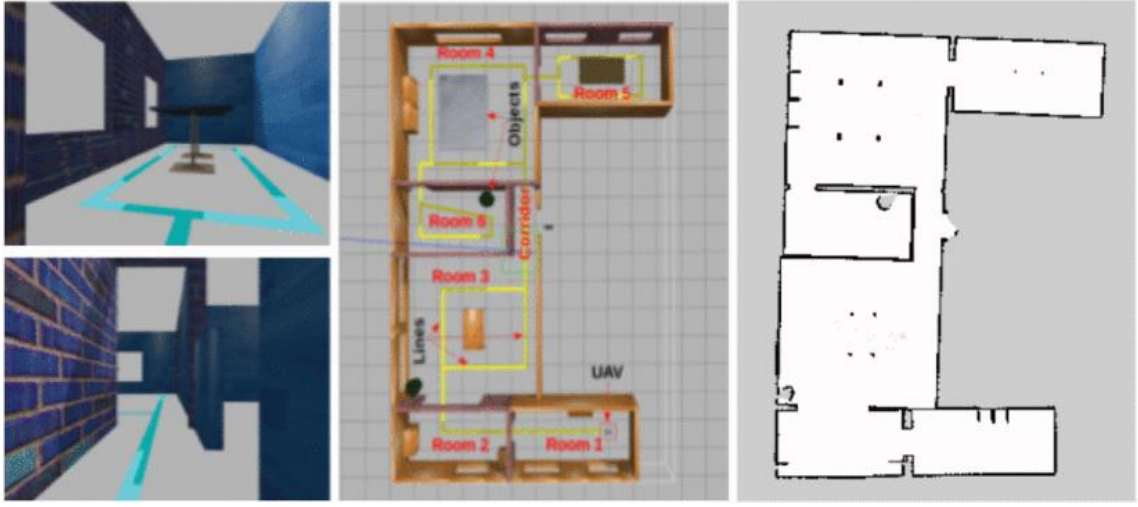
4.1.1 EuRoC Veri Seti

EuRoC veri seti (Burri ve ark., 2016), VIO algoritmalarını değerlendirmekte sıkça kullanılan ve ETH tarafından kamuya açık olarak paylaşılan bir veri setidir. Bu veri seti, bir İHA'nın iki tamamen farklı ortamda uçurularak elde edilmiştir. Kamera, IMU ve LiDAR gibi gerçek sensörleri içeren on bir sıradan oluşmaktadır. Senkronize edilmiş stereo gri tonlamalı görüntüler ve IMU bilgileri, sırasıyla 20Hz, ve 100Hz'lik hızlara sahiptirler. Konum gerçek değerleri Leica MS50 lazer izleyici ve Vicon hareket yakalama sistemi ile ölçülmüştür. Ayrıca, veritabanı, görüntü parlaklık seviyesi, görüntü bulanıklığı, uçuş hareket hızı vb. gibi kriterlere göre kolay, orta ve zor olmak üzere üç zorluk seviyesine ayrılmıştır. Bu seviyelendirme, VIO algoritmalarının farklı zorluklarla başa çıkmasına ve bu nedenle daha iyi VIO yöntemlerinin oluşturulmasına olanak tanır. Bu tezde, bu veritabanındaki farklı zorluk seviyelerinde bulunan altı sıra (V1_01-03, V2_01-03) kullanılmıştır.

4.1.2 Simülasyon Veri Seti

Simülasyon veri seti toplamak için, gerçek dünyayı simüle etmek amacıyla tasarlanmış açık kaynaklı bir kütüphane olan Gazebo kullanılmıştır (Koenig & Hsu, 2013). Bu simülasyon, ROS (Robotic Operating System) için bir eklenti olarak kullanılır ve buna uygun ortamları simüle eder (Quigley ve ark., 2009). Eğitim verileri ROS'ta Gazebo sanal ortamdan toplanmaktadır. Şekil 4.1, simülasyon ortamını ve Hokuyo UTM-

30LX LiDAR tarafından alınan iki boyutlu ızgara haritasını, Tello İHA'nın örnek kamera görüntülerini içerir. İHA, rastgele uçan ve kapalı mekanlarda sabit bir hızda iniş yapan simülasyon ortamındaki tüm bilgileri toplar. IMU ve kamera sensör verileri sırasıyla 300Hz ve 30Hz hızlarında elde edilmiştir. Ardından, gerçek konum ve sensör verilerinin ön işleme ve senkronizasyonu gerçekleştirilmiştir. Simülasyonda toplanan verilerden, gereken konum ölçümleri ile RGB (Red-Green-Blue) kamera görüntüleri etiketlenmiştir. Son olarak, işleme maliyetlerini en aza indirmek amacıyla görüntü çözünürlükleri 640×480'e düşürülmüştür.



Şekil 4.1. Simülasyon ortamı, iki boyutlu ızgara haritası ve Tello İHA'nın örnek kamera görüntüleri.

4.1.3 RAC-LAB Veri Seti

Önerilen Derin RCNN (Recurrent Convolutional Neural Network) mimarisinin küresel konum tahmini üzerindeki performansını daha iyi değerlendirmek için, Tello İHA'nın kullanıldığı bir gerçek dünya deneyi, kapalı bir laboratuvar ortamında gerçekleştirilmiştir. Eğitim veri seti, farklı nesneler içeren (örneğin, sandalyeler, masalar, bilgisayarlar, yer ve hava robotları vb.) 2015 metrekare boyutundaki bir laboratuvar ortamından toplanmıştır. Tüm uygulamalar, Python programlama dilini kullanarak ROS sisteminde gerçekleştirilmiştir. Tello, düşük maliyetli bir drone olduğu için minimal işlem gücüne sahiptir, bu nedenle yalnızca tek boyutlu görüntüler, IMU ölçümleri ve gerçek konum verilerini toplamak için kullanılmıştır. Ana konum tahmini işlemi, Tello İHA'nın kablosuz bağlantı aracılığıyla bağlı olduğu uzaktan işlem biriminde gerçekleştirilmiştir.

Drone, ROS paketleri tarafından yayınlanan konular aracılığıyla gerçek zamanlı bilgiyi uzaktaki işlem birimine gönderir ve aynı şekilde kontrol komutlarını alır.

Deney sırasında karşılaşılan bir sorun, asenkron veri toplama olmuştur. Tello İHA'dan alınan görsel verilerin h264 sıkıştırılmış video formatında olması, veri senkronizasyonunu zorlaştırmıştır. Bu sorunu çözmek için gerçek zamanlı bir algoritma uygulandı ve ardından görselin, IMU ve gerçek konum bilgilerini 30Hz hızına senkronize etmek için bir enterpolasyon tekniği kullanılmıştır. Simülasyon ortamına göre gerçek, Tello İHA'nın görsel veri kalitesi farklıdır. Veriler, h264 sıkıştırılmış videolardan kodlanmış ve kablosuz bir bağlantı aracılığıyla toplandığından, görsel veri de kalite kaybına uğramıştır ve Şekil 4.2'de gösterilmiştir.



Şekil 4.2. DJI Tello İHA ve laboratuvar ortamından elde edilen örnek bozulmuş görüntüler.

4.1.4. KITTI Veri Seti

KITTI veri seti, genellikle VO yöntemlerini değerlendirmek için kullanılan ve Alman Karlsruhe Teknoloji Enstitüsü ile Amerikan Teknik Araştırma Enstitüsü'nden

halka açık olarak temin edilebilen bir veri setidir. Bu veri seti, iki gri tonlamalı ve iki renkli kamera, HDL64E LIDAR Velodyne, dört optik lens ve veri toplama için bir GNSS/INS navigasyon sistemi içermektedir. Tez çalışmasında, yalnızca bir Sony ICX267 CCD küresel perdeye sahip sol kamera görüntülerinin verileri kullanılır ve çözünürlüğü 1241×376 'dır. Yüksek hassasiyetli bir GNSS/INS navigasyon sistemi, 100Hz frekansta ve %10 cm'den daha az hata oranıyla elde edilen gerçek zemini denemede kullanır. Odometri ölçümleri, yalnızca 00-10 sıraları gerçek değerleri içerirken, 11-21 sıraları değerlendirme amaçlarıyla gerçek değerleri içermemektedir. Şekil 4.3, bu tezde kullanılan KITTI veri setinden örnek görüntüleri göstermektedir (Geiger ve ark., 2012).



Şekil 4.3. KITTI veri seti örnek resimleri.

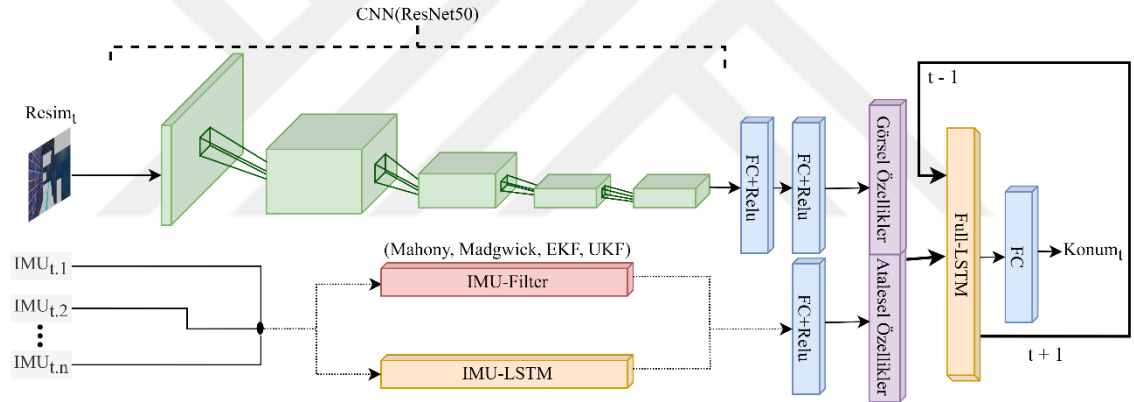
4.2. Uygulama 1: İHA'larda Kapalı Alan Küresel Konumlandırma İçin LSTM ve Filtre Tabanlı Karşılaştırma Analizi Uygulaması

Bu tez uygulamasında, derin öğrenme tabanlı konumlandırma ve SLAM teknikleri araştırılarak, İHA'ların GPS'siz kapalı alan küresel konumlandırma hassasiyeti geliştirilmiştir. Geleneksel geometrik algoritmaların ötesine geçen çalışma, klasik IMU filtreleme tekniklerini derin öğrenme metodolojileriyle birleştiren yenilikçi bir yöntem önermektedir. Bu hibrit yaklaşım, her iki yöntemin güçlü yönlerinden faydalanarak kapalı

alan İHA konumlandırma sistemlerinin doğruluğunu ve verimliliğini önemli ölçüde artırmaktadır.

4.2.1. Uygulamada Kullanılan Derin Öğrenme Mimarisi

Bu mimarinin temel amacı, geometrik uyumsuzluk kaybını en aza indirerek küresel pozisyonu doğru bir şekilde tahmin etmektir. Küresel konum bilgisini tahmin etmek için ardışık, uçtan uca ve çoklu model eğitimi formüle edilmiştir. Küresel konumlandırma ağı, ortamın daha fazla bilgisine sahip olmak için önceki katmanlardaki CNN ve LSTM çıkışlarını kullanır. Daha spesifik olarak, önerilen sinir ağı mimarisi üç katmandan oluşur; görüntü bilgisi kullanan ConvNet ağı, IMU bilgisi kullanan çift yönlü LSTM ve konum tahmini için çift yönlü LSTM ağı. Önerilen mimarinin genel bir görünümü Şekil 4.4'te gösterilmiştir.



Şekil 4.4. Önerilen Derin RCNN Konum Tahmini Sisteminin Mimarisi.

Şekil 4.4'teki mimariye göre, sıralı bir dizi monoküler RGB görüntü $(\dots, I_{t-1}, I_t, I_{t+1}, \dots)$ ve IMU ölçümleri $(\dots, IMU_{t-1}, IMU_t, IMU_{t+1}, \dots)$ verilmiş olsun, ağ küresel konum $p_t = [x_t, q_t]$ 'yi tahmin eder, burada $x \in R^3$ yer değiştirmeyi ve $a \in R^4$ dönmeyi kuaterniyon temsiliinde belirtir. CNN ve küçük LSTM akışların girişi, sırasıyla I_t ve ataletsel ölçüm IMU_t verileridir. Bu ikisinin çıkışı, küresel konum akışına giriş olarak verilir ve birleştirilir. Derin RCNN ağı etkili bir şekilde aşağıdaki Denklem (4.5)'deki eşlemeyi öğrenir, bu da görüntü ve IMU verisi giriş dizilerini küresel konumlara dönüştürür. Denklem (4.1)'de, $W \times H$, giriş kamera görüntülerinin genişliği ve yüksekliğidir ve $1:T$ zaman adımlarıdır.

$$DerinRCNN : \{(R^{W \times H}, R^6)_{1:T}\} \rightarrow \{(R^7)_{1:T}\} \quad (4.1)$$

4.2.1.1. Görsel-CNN Özellik Çıkartıcı

Bir CNN, monoküler RGB görüntü I_t 'den görsel özellikler çıkartmak ve konum tahmini için uygun özellikleri etkili bir şekilde öğrenmek amacıyla uygulanmıştır. Görsel-CNN özellik çıkartıcı konfigürasyonu Çizelge 4.1'de listelenmiştir.

Çizelge 4.1. Görsel-CNN Özellik Çıkartıcı Yapılandırması.

Katman	Aktivasyon Fonksiyonu	Giriş	Çıkış
ResNet50 + GlobalAvgPool2D	-	224×224	$Batch \times 2048$
Tam Bağlı Katman 1 (FC1)	ReLU	$Batch \times 2048$	$Batch \times 1024$
Tam Bağlı Katman 2 (FC2)	ReLU	$Batch \times 1024$	$Batch \times 200$

Görsel-CNN Özellik Çıkartıcı modülü, ResNet50 mimarisine dayanmaktadır (Dosovitskiy ve ark., 2015). ResNet50, son ortalama havuzlama (average pooling) katmanından önce kırılmış ve softmax katmanları kaldırılmıştır. Her konvolüsyon katmanını takiben bir grup normalleştirme ve bir ReLU bulunmaktadır. Ortalama havuzlama daha sonra küresel ortalama havuzlama ile değiştirilir, ardından iki FC katmanı eklenir. Son konvolüsyon katmanının çıkışı düzeltilir ve ilk FC katmanına giriş olarak verilir. Son olarak, görsel özelliklerin vektörü z_I , ikinci FC katmanı tarafından elde edilir (Denklem (4.2)).

$$z_I = VisualCNN(R^{W \times H}) \quad (4.2)$$

Ön işlenmiş görüntüler veya nokta bulutları yerine CNN, düşük boyutlu konum tahmini için optimal özellikleri çıkarmayı öğrenen Görsel-CNN'nin girişi olarak ham RGB görüntüleri I_t alır. RGB görüntüsündeki yüksek boyut düzeyini kompakt bir vektörde azaltmanın yanı sıra, bu öğrenilmiş özellik temsilleri sıralı eğitim sürecini iyileştirir. Bu nedenle, ardışık modelleme için son konvolüsyonel özellikler Full-LSTM'ye iletilir.

4.2.1.2. IMU-LSTM Özellik Çıkartıcı

Görsel ataletsel konum tahmini probleminde, IMU ölçümlerinin işlenmesinde bir zorluk vardır çünkü bunlar kamera kare hızından (örneğin, 10 - 30 Hz) daha yüksek bir

hızda alınır (örneğin, 100 - 300 Hz). Önerilen ağda bununla başa çıkmak için, (Chen ve ark., 2018) kaynağından ilham alınarak oluşturulan küçük çift yönlü bir LSTM kullanılır ve sürekli kamera verileri arasında altı boyutlu vektör oluşturan ham ataletsel veri IMU_t 'nin toplu işlenmesi yapılır ve karşılık gelen özellik vektörü z_{IMU} olarak verilir (Denklem (4.3)).

$$z_{IMU} = IMULSTM(R^6) \quad (4.3)$$

Çizelge 4.2, IMU-LSTM özellik çıkartıcı modülünün konfigürasyonunu göstermektedir. IMU-LSTM, $IMU_t = (\alpha_t, \omega_t) \in R^6$ şeklindeki ham ataletsel ölçümlerden oluşan bir toplu işleme alır, burada $\alpha_t \in R^3$ doğrusal ivme ve $\omega_t \in R^3$ açısal hızdır. IMU modülü, her zaman çerçevesinde aynı boyutta giriş alır. Bu işleme modülü, biri çift yönlü LSTM ve diğeri tam bağlantılı bir katmandan oluşan iki ardışık dal kullanır. FC katmanının çıkış özellik vektörü daha sonra Full-LSTM'ye taşınır.

Çizelge 4.2. IMU-LSTM Özellik Çıkartıcı Yapılandırması.

Katman	Aktivasyon Fonksiyonu	Giriş	Çıkış
Gömme (Embedding) + LSTM	Sigmoid	$10 \times 10 \times 6$	$Batch \times 400$
Tam Bağlı Katman 1 (FC1)	ReLU	$Batch \times 400$	$Batch \times 200$

4.2.1.3. Full-LSTM Konum Regresörü

Full-LSTM konum regresörünün yapılandırması Çizelge 4.3'te sunulmuştur. Konum tahmini temel ilkesi, ardışık özellikler arasında bağlantılar türetmek için zamanla ilgili bağımlılıkları modelleme talep etmektedir. Bu nedenle, çift yönlü Full-LSTM, gizli önceki durumları h_{t-1} ile birlikte konkatensasyon özelliği $z_t = concat(z_I, z_{IMU})$ 'yu giriş olarak alır. Bu çizelgede konkatensasyon ifadesi $\oplus$ ile gösterilmiştir. LSTM, gizli durumları kullanarak ardışık girişlerin dinamiklerini ve zamansal ilişkisini kullanır. Full-LSTM'den sonra tam bağlantılı bir katman, konum tahmincisi olarak hizmet verir. Bu, x ve q olarak sırasıyla pozisyon ve kuaterniyon rotasyonunun regresyonu için boyutu yedi olan bir vektördür. Genel olarak, tam bağlantılı katman, özelliklerin vektör temsilini bir konum vektörüne dönüştürür (Denklem (4.4)).

$$x_t = FullLSTM(z_t, z_{t-1}^h) \quad (4.4)$$

Çizelge 4.3. Full-LSTM Küresel Konum Regresör Yapılandırması.

Katman	Aktivasyon Fonksiyonu	Giriş	Çıkış
Concat	-	$200 \oplus 200$	$Batch \times 400$
LSTM	Sigmoid	$Batch \times 400$	$Batch \times 100$
Tam Bağlı Katman (FC)	ReLU	$Batch \times 100$	$Batch \times 7$
Eğitim Parametreleri	Optimizer = Adam		
	Learning Rate = 1e-4		
	Decay = 1e-4/200		

4.2.1.4. Hata Fonksiyonu

Konum tahmini için robotun pozisyonu ve rotasyonu, Wang ve ark. (2018) tarafından önerilen bir yaklaşıma dayanarak tahmin edilir. Burada, konum tahmini olasılıksal bir problem olarak kabul edilir. Yani, ardışık konumlar X_t seti, bunların ilgili kamera görüntüleri I_t ve zaman t 'ye kadar olan IMU verileri IMU_t , Denklemler (4.5)-(4.9) gibidir.

$$X_t = (x_1, x_2, \dots, x_t) \quad (4.5)$$

$$I_t = (i_1, i_2, \dots, i_t) \quad (4.6)$$

$$IMU_t = (imu_1, imu_2, \dots, imu_t) \quad (4.7)$$

$$d_t = (i_t, imu_t) \quad (4.8)$$

$$D_t = (d_1, d_2, \dots, d_t) \quad (4.9)$$

Denklemler (4.5)-(4.9) baz alınarak, Denklem (4.10) elde edilir.

$$P(x_t|D_t) = P(x_1, x_2, \dots, x_t|d_1, d_2, \dots, d_t) \quad (4.10)$$

Denklem 4.10'u optimize ederek optimal ağırlıkları θ^* öğrenmek mümkündür.

$$\theta^* = \operatorname{argmax} P(X_t|D_t; \theta) \quad (4.11)$$

Bu nedenle, MSE'ye dayalı olarak zaman t 'de gerçek değer $x_t = (p_t^T, \omega_t^T)^T$ ve tahmini $\hat{x}_t = (\hat{p}_t^T, \hat{\omega}_t^T)^T$ 'nin farkı, Denklem (4.12) ile azaltılabilir:

$$\theta^* = \underset{\theta}{\operatorname{argmin}} \frac{1}{T} \sum_{t=1}^T \|\hat{p}_t - p_t\|_2^2 + \lambda \|\hat{\omega}_t - \omega_t\|_2^2 \quad (4.12)$$

burada p ve ω , $\|\cdot\|$ ve λ ile temsil edilen pozisyonların ve rotasyonların ağırlıklarını dengelemek için bir ölçek faktörüdür. ω , Euler temsiliinde küresel koordinat sisteminde sorunlarla karşılaşabileceğinden kuaterniyon olarak temsil edilir.

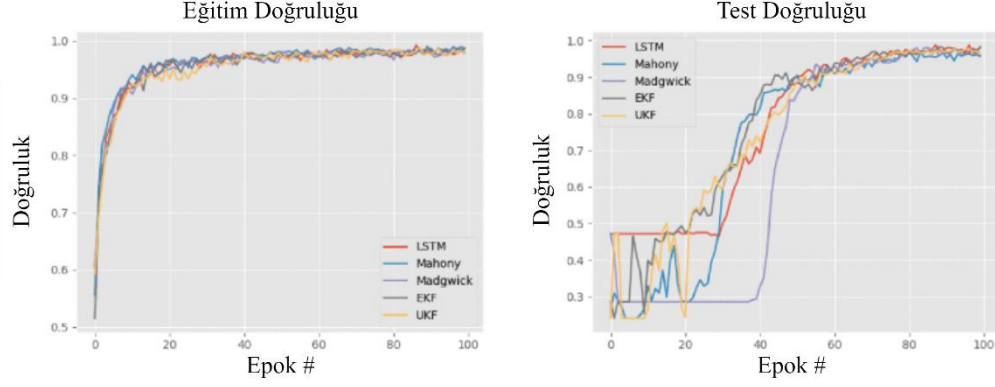
4.2.2. Model Eğitimi

Tüm gösterilen deneyler, 16 GB (Giga Byte) DDR3 RAM (Random Access Memory) ve NVIDIA GeForce GTX 1660Ti GPU ile donatılmış bir Intel(R) Core(TM) i7-9750H CPU (Central Processing Unit) işlemci sistemde gerçekleştirilmiştir. Konum tahmini öğrenme sürecini ve GPU uygulamalarını değerlendirmek için Google Brain ekibi tarafından geliştirilen Tensorflow kullanılmıştır (Abadi ve ark. 2016). Tüm veri ön işleme işlemleri, Tensorflow'un Python bağlantılarıyla uyumlu ilgili kütüphaneler kullanılarak Python'da uygulanmıştır.

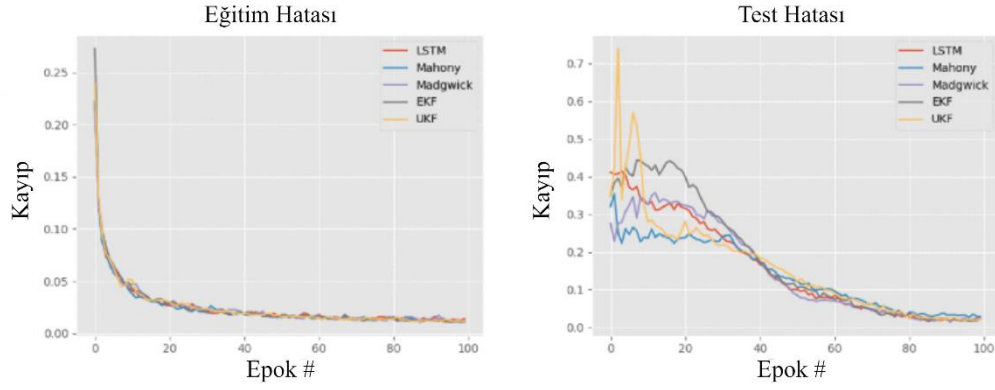
Tek gözlü görüntülerden en iyi özellikleri elde etmek için, önerilen derin konum tahmincisinin CNN özellik çıkartıcı modülünde transfer öğrenme tekniği kullanılmıştır. Burada, ResNet50 modeli, 4.2.1.1 bölümünde açıklandığı gibi son sınıflandırma katmanı çıkarılmış şekilde kullanılmıştır. ResNet50, $224 \times 224 \times 3$ boyutundaki giriş görüntülerini kabul ettiğinden, görüntüler buna göre küçültülmüştür. Tam ağ, ortalama olarak veri seti başına üç saat süren 100 epoch eğitilmiştir. Adam algoritması (Kigma & Ba, 2014), öğrenme hızı $= 1e - 4$, azalma $= \frac{1e-4}{200}$ ve MAE (Mean Absolute Error) hatası işlevi ile optimizasyon fonksiyonu olarak kullanılmıştır.

Şekil 4.5, derin RCNN'nin beş farklı yapılandırmasıyla EuRoC dizisi "V1_02_medium" için eğitim ve doğrulama kaybı ve doğruluğunu göstermektedir: Kırmızı eğri, sadece IMU-LSTM ile eğitilen Derin RCNN konumlandırmanın eğitim doğruluğunu ve kaybını gösterirken, mavi eğri Mahony ile eğitilen doğruluk ve kaybı, mor eğri Madgwick ile eğitilen doğruluk ve kaybı, gri ve sarı eğriler sırasıyla EKF ve UKF ile eğitilen VIO'nun doğruluğunu ve kaybını gösterir. Ağın eğitim setindeki yakınsaması tüm ağ formları için aynı olduğu gözlemlenmiştir. Ancak, doğrulama setinde doğruluğun yakınsaması, diğer ağ formlarına kıyasla EKF ve UKF için daha hızlıdır. Doğrulama kaybı grafiği için UKF ve Mahony'nin daha hızlı bir yakınsaması vardır.

Ayrıca, ağın LSTM formunun hem doğrulama doğruluğu hem de hata grafiklerinde orta performansa sahip olduğu belirtilmelidir.



(a) Eğitim ve Test Doğruluğu Performans Karşılaştırması



(b) Eğitim ve Test Hatası Performans Karşılaştırması

Şekil 4.5. EuRoC veri seti V1_02 orta dizisinde beş farklı ağ formu kullanarak eğitim ve doğrulama doğruluğunun (a) ve kaybın (b) karşılaştırılması.

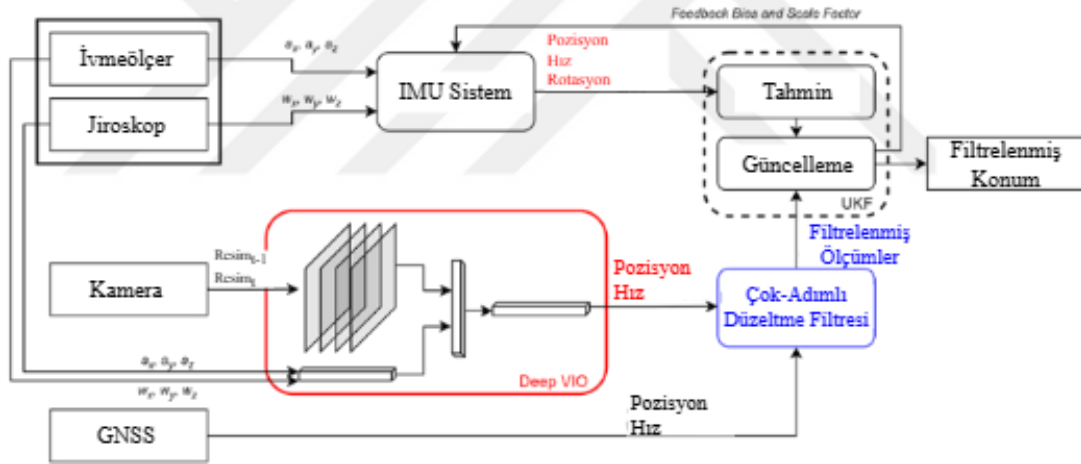
4.5. Uygulama 2: Genelleştirilebilir D-VIO ve GNSS/IMU ile Birleştirilmiş İyileştirilmiş Otonom Araç Konumlandırma Uygulaması

Bu tez uygulaması, özellikle geniş açık hava ortamlarında otonom araç konum belirleme hassasiyetini ve güvenilirliğini artırmaya odaklanmaktadır. GNSS sinyal karışımı veya engellenmesinden kaynaklanan zorlukları hafifletmek için, yanlış GNSS verilerini düzeltmek için sofistike çok aşamalı bir düzeltme filtresi kullanılmaktadır. Önerilen çözümler, farklı sensörlerden gelen verileri entegre ederek bireysel sensör kısıtlamalarını telafi eder ve yeni bir GD-VIO yaklaşımını tanıtır. GNSS kesintileri durumunda, GD-VIO sağlam araç konum belirleme sağlar. Sıkı gerçek dünya deneyleri,

algoritmaların etkinliğini kanıtlamakta olup, doğru ve güvenilir araç duruş tahminleri sağlama yeteneklerini göstermektedir.

4.5.1. Mimari

Bu uygulamada, toplanan hatalı GNSS verilerini düzeltmek için çok aşamalı bir düzeltme filtresi sunulmaktadır. Ardından, şehir ortamlarında otonom araçlar için konum belirleme hassasiyetini ve güvenilirliğini artırmak amacıyla geliştirilebilir bir uçtan uca çoklu model Derin VIO konusu ele alınmıştır. Bu yaklaşım, çeşitli çevresel zorluklara genelleme yapabilir ve GNSS hatası durumunda araç konum belirlemeyi iyileştirebilir. Son olarak, çok aşamalı düzeltme filtresine dayalı gevşek bağlantılı bir GNSS/GD-VIO/IMU konum belirleme yaklaşımının, klasik UKF tabanlı teknikten daha iyi performans sergilediği gösterilmiştir. Şekil 4.6, önerilen konum belirleme tekniğinin bir akış şemasını göstermektedir.



Şekil 4.6. Önerilen GNSS/D-VIO/IMU yönteminin akış şeması.

Şekil 4.6'da Derin VIO modülü kırmızı bir kutu ile belirtilmiştir. IMU sensörü aracılığıyla x , y ve z eksenli doğrusal ivme değerleri (a_x , a_y , a_z) ile x , y ve z eksenli açısal hız değerleri (w_x , w_y , w_z) hesaplanır. Bu hesaplanan değerler UKF'nin tahmin adımına giriş olarak sağlanır. Aynı anda, kamera görüntüleri ile birlikte sunulan veri seti, Pozisyon ve Hızı tahmin etmek için Derin VIO modülüne giriş olarak verilir. GNSS tarafından hesaplanan Pozisyon ve Hız, Multi-Step Düzeltme Filtresi için giriş olarak kullanılır. Bu filtre tarafından türetilen nitelikli ölçümler, UKF'nin güncelleme adımına

yönlendirilir ve süzülen konumun tahmin edilmesine yol açar. Her modülün ayrıntıları, 4.5.1.1-4.5.1.3 bölümlerinde kapsamlı olarak verilmiştir.

Özetle, bu çalışmadaki ana katkılar aşağıdaki gibidir:

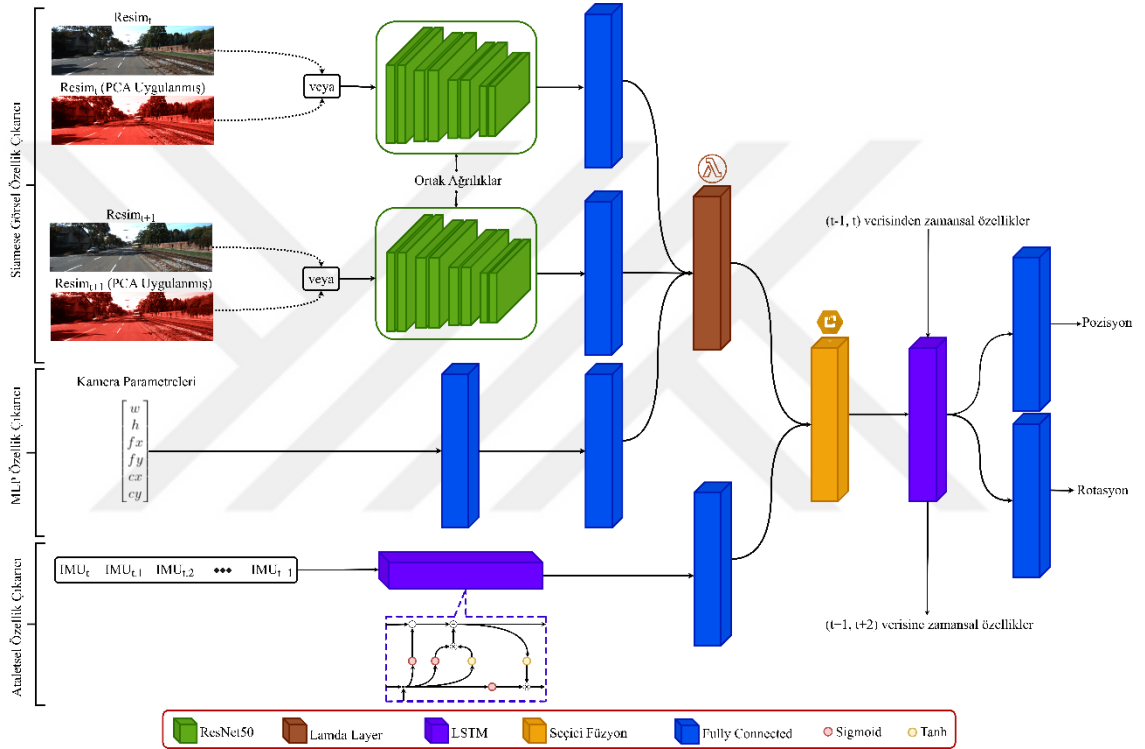
- GNSS sıçramalarıyla başa çıkmak için çok adımlı bir UKF tabanlı filtre sunulmuştur, otonom araç konumlandırma doğruluğunda bir artış olmuştur.
- Geleneksel kısıtlı VIO yöntemlerinden ayrılan çeşitli ortamlara uyumlulukta mükemmel bir performans sergileyen GD-VIO sunulmuştur, bu da konum tahmini için yeni bir derin öğrenme tabanlı VIO yöntemidir.
- Çok adımlı düzeltme filtresini ve GD-VIO'yu birleştirerek IMU füzyon uygulaması oluşturulmaktadır. Bu da GNSS/GD-VIO/IMU füzyonunu ortaya çıkararak, özellikle GNSS çekmeyen alanlarda üstün otonom araç konumlandırması sağlamaktadır.
- Yöntem, gerçek dünya verileriyle test edilmiştir ve önde gelen yaklaşımlarla karşılaştırılmıştır. Sonuçlar, konumlandırma doğruluğunu ve GNSS atlamalarına karşı direnci arttırmaktadır.

Önerilen çok adımlı düzeltme filtresi ve GDVIO yöntemi, genel olarak otonom araç konumlandırma alanına önemli bir katkı sağlamakta olup, şehir ortamları için daha sağlam ve doğru bir çözüm sunmaktadır.

4.5.1.1. Genelleştirilmiş Derin VIO (GD-VIO)

Bu bölümde, VIO'nun tahmin edilmesi için önerilen GD-VIO yöntemi açıklanmaktadır. GD-VIO modelinin genel mimarisi Şekil 4.7'de gösterilmiştir. Bu model, bir Siyam tipi görüntü özellik kodlayıcı, MLP içsel özellik çıkarıcı, lambda birleştirme modülü, küçük Bi-LSTM IMU özellik kodlayıcı, seçici sensör birleştirme modülü, daha büyük bir Bi-LSTM zamansal özellik kodlayıcı ve bir konum tahmincisi içerir. Siyam görüntü kodlayıcısı, transfer öğrenme kullanarak ardışık görüntülerden önemli özellikleri yakalar. Bir MLP, kamera görüntülerinin içsel parametrelerinden küçük özellikler çıkarmak için kullanılır. İki giriş görüntüsünden ve MLP içsel özelliklerinden görsel özellikleri birleştirmek için bir lambda katmanı kullanılır, bu

katman görsel özelliklere L2 norm ve çıkarma birleştirme işlemleri yapar. IMU ölçümleri akışından veri çıkarmak için bir LSTM kullanılır, çünkü LSTM, ardışık verilerde bulunan zamansal bilgileri modelleme konusunda daha uygun bir yapıya sahiptir (Yusefi ve ark., 2021). Sensör birleştirme modülünde, görüntülerden ve IMU verilerinden belirli özellikler seçilerek konum tahmin için kullanılır. Bu yeniden ağırlıklandırılmış özellikler daha sonra daha büyük bir LSTM modülüne ve ardından konum tahmin modülüne girer. Şekil 4.7’de gösterilen her modül için ayrıntılar aşağıda verilmiştir.



Şekil 4.7. Önerilen GD-VIO'nun mimarisi.

Siamese Görsel Özellik Çıkartıcı: Verilen ardışık monoküler görüntü çifti $x_T = (I_t, I_{t+1}) \in R^{2 \times (w \times h \times c)}$, Siamese ağlarının amacı, giriş görüntülerinden görsel olarak anlamlı bilgileri çıkartarak konum tahmin sürecini kolaylaştırmaktır. Bu modül, eşleşen mimari, parametre ve ağırlıklara sahip ikiz CNN alt ağlardan oluşur. Herhangi bir parametre değişikliği, iki alt ağ arasında senkronize edilir, bu nedenle birindeki ağırlıkları güncellemek, diğerinin ağırlıklarını da günceller. Bu mimari, iki görüntünün özelliklerinin ayrılabilir ve aynı özelliklere sahip olmasını sağlar, böylece L2 norm ve çıkartmalı füzyon işlemlerini gerçekleştirebilir. Her bir alt ağ, ResNet50'e

dayanmaktadır. ResNet50, son ortalama havuz katmanından önce kısaltılmış ve geriye kalan katmanlar silinmiştir. Her konvolüsyonel katman, grup normalizasyonu ve ReLU ile takip edilir. Ortalama havuzun yerine küresel ortalama havuz kullanılır, ardından bir FC katmanı gelir. Son konvolüsyonel katmanın sonucu düzleştirilir ve başka bir tamamen bağlı katmana gönderilir ve görsel özellikler z_{I_t} ve $z_{I_{t+1}}$ elde edilir (Denklem (4.13)). İki z_I vektörü daha sonra 4.5.1.1 bölümünde açıklanan lambda katmanına beslenir.

$$\begin{cases} z_{I_t} = \text{AltAğCNN}(I_t), & I_t \in R^{w \times h} \\ z_{I_{t+1}} = \text{AltAğCNN}(I_{t+1}), & I_{t+1} \in R^{w \times h} \end{cases} \quad (4.13)$$

Çizelge 4.4, Siamese alt ağının kurulumunu göstermektedir.

Çizelge 4.4. Alt Ağ CNN özellik çıkartıcısı konfigürasyonu.

Katman	Akt. fonksiyonu	Giriş	Çıkış
Resnet50 + GlobalAvgPool2D	-	224×224	$\text{Batch} \times 2048$
FC1	Leaky ReLU	$\text{Batch} \times 2048$	$\text{Batch} \times 1024$

Güvenilirliği artırmak amacıyla Siamese ağına katılmadan önce görüntülere rastgele PCA veri artırımı eklenir. Bu durum, giriş görüntülerinin RGB kanallarındaki yoğunlukları ayarlayarak modeli daha genelleştirilebilir hale getirir. Bulunan temel bileşenlerin katları, her bir giriş görüntüsüne uygulanır ve büyüklükleri buna göre orantılıdır. Dolayısıyla, her RGB piksel için $I_{xy} = [I_{xy}^R, I_{xy}^G, I_{xy}^B]^T$, Denklem (4.14) gibi bir değer eklenir:

$$\Delta I_{xy} = [pc_1, pc_2, pc_3][a_1 b_1, a_2 b_2, a_3 b_3]^T \quad (4.14)$$

Bu denklemde, ΔI_{xy} , bir görüntünün belirli bir konumundaki RGB piksel değerlerine eklenen renk değişikliğini, pc_i , 3×3 boyutunda bir kovaryans matrisinin i^{inci} özvektörü ve b_i , i^{inci} özdeğeridir ve a_i , rastgele seçilmiş bir değişkendir. Her a_i , belirli bir giriş görüntüsünün tüm pikselleri için bir kez seçilir. Bu düzen, görüntülerde ışık yoğunluğu ve renkteki değişikliklere karşı değişmezlik özelliğini ekler.

MLP Dahili Özellik Çıkarıcı: Bu modül, kamera dahili parametrelerini işlemek için tasarlanmış basit bir MLP'dir ve modelin farklı kamera kurulumlarında iyi

performans sergilemesine olanak tanır. Bu modülün girişi, x ve y eksenlerinde görüntü genişliği ve yüksekliği, kamera odak uzunlukları (f_x, f_y) ve optik merkezler (c_x, c_y) içerir. İki tamamen bağlı yoğun katman ve bunların aktivasyon fonksiyonu olarak ReLU içerir. Çizelge 4.5, MLP dahili katman alt ağı'nın kurulumunu göstermektedir.

Çizelge 4.5. MLP Dahili Özellik Çıkartıcı Konfigürasyonu.

Katman	Akt. fonksiyonu	Giriş	Çıkış
FC1	ReLU	$Batch \times 6$	$Batch \times 8$
FC2	ReLU	$Batch \times 8$	$Batch \times 7$

Lambda Katmanı: İki ardışık giriş görüntüsünden özellik çıkardıktan sonra, bu özellikleri işlemek için ağı bir lambda katmanı eklenir. Bunun için önce özelliklere bir L2 norm ve bir çıkarma işlemi uygulanır, ardından birleştirme katmanı takip eder.

Makine öğrenimi alanında, L2 norm kavramı, bir düzenlilik terimi ekleyerek modellerin aşırı uyumunu azaltmak için kullanılır. L2 norm, aynı zamanda öklit mesafe olarak bilinen matematiksel bir terimdir ve bir veya iki vektöre uygulanarak mesafeleri ifade etmek için kullanışlıdır. Bu kavramdan ilham alarak, Siamese alt ağının çıkış görsel özellikleri üzerinde L2 norm işlemi gerçekleştiriyor ve bunu lambda katmanına ekliyoruz. L2 norm, L1 normdan daha iyi bir kararlılığa sahip olduğu için tercih edilir. İki vektör arasındaki mesafeyi hesaplamak için kullanılan L2 norm Denklem (4.15) ile hesaplanır.

$$L2\ Norm(x, y) = \sqrt{\sum_{i=1}^n |x_i - y_i|^2} \quad (4.15)$$

Füzyon bloğu, ilgili Siamese alt ağlarından elde edilen görsel özellik akışlarını birleştirir ve bunlara L2 norm terimini ekler. Bu işlem, Denklem (4.16)'da gösterilmiştir. Veri füzyonu genellikle iki özellik akışını tek bir özellik uzayına birleştirme veya bunları tekrar ağırlıklandırma yoluyla gerçekleştirilir, her ikisi de eklemeli etkileşimlerdir.

$$f(z_{I_t}, z_{I_{t+1}}, z_{C_t}) = W \times \text{concat}((z_{I_t}, z_{I_{t+1}}), (z_{I_t} \ominus z_{I_{t+1}}), z_{C_t}, L2\ Norm(z_{I_t}, z_{I_{t+1}})) \quad (4.16)$$

Burada, ağırlık matrisi W ve vektör c öğrenilen parametrelerdir, $z_{I_t}, z_{I_{t+1}}$ ve z_{c_i} ise iki Siamese ağı tarafından çıkarılan görsel özellikler ve MLP ağı tarafından çıkarılan dahili özelliklerdir.

LSTM Ataletsel Özellik Çıkarıcı: LSTM ağları, uzun vadeli ilişkileri yakalama konusunda mükemmel bir performans sergiler, geçmiş giriş verilerini mevcut çıkışı etkilemek için kullanır. Sensör tarafından yakalanan aktivite verileri zaman serisi formatında olduğundan, LSTM eğitimi için eğitim verilerini etkili bir şekilde hazırlamak önemlidir. LSTM çözümündeki IMU veri girişi çok boyutludur, üç eksenli bir ivmeölçerden ve üç eksenli bir jiroskoptan gelir. Görev, altı paralel bir boyutlu zaman serisi verisini gereken 3B LSTM yapısına yeniden şekillendirmeyi içerir.

Seçici Sensör Füzyonu: Bu bölüm, kamera ve IMU'dan elde edilen görsel ve ataletsel özellikleri birleştirmek içindir (Chen ve ark., 2019). Burada özellik seçimini belirleyici (deterministic) bir şekilde modelleyen bir füzyon yaklaşımı sunulmaktadır. Bu işlev, görsel ve ataletsel koşullarla özellikleri yeniden kalibre eder ve diğer modüllerle işbirliği yapabilen bir özellik sunar.

Bi-LSTM ve Göreceli Konum Tahmin Edici: Çizelge 4.6, Bi-LSTM ve konum regresörü katmanlarının düzenini açıklar. Burada, VIO'nun temel konsepti, ardışık özellikleri temsil ederek ardışık verilerdeki bağlantıları türetmeyi gerektirir. Bu nedenle, Bi-LSTM, birleşik görsel özellikler $z_t = f(z_{I_t}, z_{I_{t+1}})$ ile birlikte önceki gizli durumları h_{t-1} kabul eder. LSTM, gizli durumları kullanarak girişin dinamiklerini ve zamansal ilişkileri kullanır. Bi-LSTM'den sonra, iki tam bağlı katman, konum tahmincileri olarak hizmet eder. İlk FC, pozisyonu tahmin etmekten sorumludur ve $t = (x, y, z)$ boyutunda bir boyuta sahiptir. İkinci FC, kuaterniyon formatında rotayı tahmin eder ve $r = (x, y, z, w)$ boyutunda bir boyuta sahiptir. Özetle, tam bağlı katmanlar, z_t ve z_{t-1} özellik vektörlerini, Denklemler (4.17)-(4.19) kullanılarak konum vektörlerine dönüştürür:

$$s_t = LSTM(z_t, z_{t-1}^h) \quad (4.17)$$

$$t_t = FC1(s_t), \quad t_t \in R^3 \quad (4.18)$$

$$r_t = FC1(s_t), \quad r_t \in R^4 \quad (4.19)$$

Çizelge 4.6. Bi-LSTM ve Konum Regresörü Konfigürasyonu.

Katman	Akt. fonksiyonu	Giriş	Çıkış	Eğitim Param.
LSTM	Sigmoid	$Batch \times 400$	$Batch \times 500$	Öğrenme Hızı= $1e-4$
Pozisyon FC	Doğrusal	$Batch \times 500$	$Batch \times 3$	Opt. Alg. = Adam
Rotasyon FC	Doğrusal	$Batch \times 500$	$Batch \times 4$	Azalma = $1e-4$

Öğrenme Mekanizması: Eğitim süreci boyunca, çerçeveye iki ardışık görüntü ve bu anlık alınan IMU ölçümleri ile birlikte kamera parametreleri verilir. Görüntülere rasgele bir PCA artırımı uygulanır ve bunlar Siamese alt ağlarına giriş olarak verilir. Bu alt ağlar ağırlıkları paylaşır ve görüntünün görsel özelliklerini çıkarırlar. Siamese ağının her alt ağı, eğitim sürecinde öğrenmeyi hızlandırmak için FC katmanları hariç olmak üzere ConvNet olarak ResNet50'yi kullanır ve dondurulur. Aynı anda, MLP ağına kameranın dahili parametreleri verilir ve iç özellikleri çıkarmak için kullanılır. Bu iç özellikler, görsel özelliklerle birlikte lambda katmanı tarafından işlenir ve ilgili çıkış özelliklerini elde etmek için seçmeli birleştirme katmanına giriş olarak kullanılır. Benzer şekilde, IMU'dan gelen ataletsel özellikler de seçmeli birleştirme katmanına gönderilir. Daha büyük Bi-LSTM katmanı, önceki sekansın gizli özelliklerini seçmeli birleştirme katmanından aldığı gibi önceki sekansın özellikleriyle birlikte alır ve iki özellik vektörü üretir. İlk vektör, nihai konum regresyon katmanına giriş olarak verilir ve ikinci vektör, bir sonraki LSTM sekansına gizli özellik vektörü olarak iletilir. Son katmanda, ağın maliyeti hata fonksiyonu kullanılarak hesaplanır ve ağırlıkların güncellenmesi için geriye doğru yayılım kullanılır. Hata fonksiyonuna L2 normuna dayalı olasılıksal bir yaklaşım kullanılmaktadır. Bu süreç, kaybı en aza indirmeyi ve konum tahminlerinin olasılığını maksimize ederek optimum ağırlık parametrelerini elde etmeyi amaçlar. Bu işlemler, Denklemler (4.20) ve (4.21) ile elde edilir.

$$\theta^* = \operatorname{argmax} P(X_t|I_t; \theta) \quad (4.20)$$

$$\theta^* = \operatorname{argmin} \frac{1}{T} \sum_{t=1}^T \|\hat{x}_t - x_t\|_2^2 + \lambda \|\hat{o}_t - o_t\|_2^2 \quad (4.21)$$

Denklem (4.20) içinde, θ^* optimal ağırlık parametrelerini, $P(X_t|I_t; \theta)$ X_t konumunun I_t giriş verilerine bağlı olarak ağırlık matrisi ile parametrelendirilmiş olasılığını temsil eder. Denklem (4.21) içinde, x ve o , konum pozisyon ve yönlendirmedir. ve λ , Euler açılarının dünya koordinat sisteminde sorunlara yol

açabileceği durumları düzeltmek için konum parametrelerini ayarlamak için kullanılan bir ölçek değişkenidir. o , Euler formatında ifade edilmez, çünkü bu durum dünya koordinat sisteminde sorunlara yol açabilir.

4.5.1.3. GNSS/GD-VIO/IMU Füzyonu

Genel olarak, hassasiyeti ve mutlak konum tahmin yetenekleri nedeniyle GNSS, IMU'yu mükemmel bir şekilde tamamlar. Ne yazık ki, şehir ortamlarında yüksek binaların arasından geçerken araçlarla GNSS pozisyonlarını elde etmek zordur ve ölçümlerin istikrarlı ve güvenilir olduğunu sağlamak zordur. Bu kısıtlamaları aşmak için GD-VIO gibi diğer konumlandırma yaklaşımları, entegre sistemimizde yoğun bir şekilde kullanılır. Bu bölümde, GNSS/GD-VIO/IMU sensör okumalarını entegre eden ve yüksek doğrulukta araç pozisyon bilgisi üreten UKF tabanlı konumlandırma stratejisi sunulmaktadır.

UKF, durum değişkenlerinin sonlu noktalar setini kullanarak durum değişkenlerinin sonlu noktalar setini tahmin etmeye yönelik bir dizi sigma noktası kullanarak bir arka olasılık dağılımını yaklaştıran iyi bilinen bir doğrusal olmayan filtreleme algoritmasıdır. UKF'nin, EKF karşısında bir dizi avantajı vardır, bunlar arasında kovaryans matrisinin daha iyi bir tahmin edilmesi, sayısal kararlılıkta iyileşme ve daha düşük hesaplama maliyeti bulunur.

Bu çalışmada, GNSS/GD-VIO/IMU birleştirme sürecinin son adımında UKF kullanılmaktadır. Burada, filtrelenmiş GNSS ölçümlerini ve GD-VIO sonuçlarını EKF ile IMU verileriyle birleştirilmektedir. UKF, bu girişleri kullanarak durum uzayı modeli oluşturur ve bu modelde IMU'dan gelen yönlendirme verilerini içerir. Bu yaklaşım, aracın konumu, hızı ve duruşunun kesin bir şekilde temsil edilmesini sağlar ve doğru navigasyon tahminleri için hayati öneme sahiptir.

Araç konumlandırma sistemleri için dört koordinat sistemi tanımlanabilir:

1. Jeosantrik (geocentric) koordinat sistemi: Yer'in kütle merkezinden X , Y ve Z ölçümlerini kullanarak yakındaki konumları temsil etmek için kartezyen bir uzamsal referans kullanır.

2. Küresel koordinat sistemi: Kuzey, Doğu ve Aşağı yönler işaret eden X , Y ve Z eksenleri ile bilinen bir küresel koordinat sistemine Kuzey-Doğu-Aşağı koordinat sistemi denir.
3. Araç koordinat sistemi: X , Y ve Z eksenleri, aracın koordinat sisteminde sırasıyla ileri, sola ve yukarıyı göstermektedir.
4. Sensör koordinat sistemi: Kurulu sensörlerin üç dik eksen, her bir sensör için farklı olabilir. Bu nedenle, sensörden gövdeye hizalama kalibrasyonu yapıldıktan sonra, sensörlerin aracın koordinat sistemine hizalanacağı varsayılmaktadır. Son derece doğru konumlandırma için sensörler ile araç koordinat sistemi arasındaki dış parametreleri (yer değiştirme ve dönme farkları) elde etmemiz gerekmektedir.

Denklemler (4.22) ve (4.23)'te durum uzayı modelinin nasıl tanımlandığı açıklanmıştır. Bu denklemlerde, p ve v , sırasıyla aracın konumunu ve hızını ifade eder. q , normu bir olan oryantasyonu yansıtan bir kuaterniyon'dır: $\|q\| = 1$. Sensor kusurları veya kalibrasyon hatalarından kaynaklanan sapma terimlerini içeren vektör b ve içsel öngörülemezlik ve model dışı bozuklukları hesaplayan vektör d olmak üzere iki ek vektörümüz bulunmaktadır. Bu vektörler, aracın konumu, hızı ve dönmeyi daha kapsamlı ve doğru bir şekilde tahmin etmeyi sağlamak için durum uzayı modeline entegre edilmiştir. Tanıtılan belirsizlik ve sapma faktörleri, konumlandırma sürecinin sağlamlığını ve güvenilirliğini artırmak amacıyla UKF ile birlikte ele alınmaktadır (Meng ve ark., 2017).

$$x = [p, v, q, b, d]_{16 \times 1}^T \quad (4.22)$$

$$q = [e^T, q_4]^T = [q_1, q_2, q_3, q_4]^T \quad (4.23)$$

Sensör Modeli: Sensör modeli, aracın bir sonraki durumu ile mevcut durumu arasındaki dinamik etkileşimi tanımlamak için kullanılır. Sensör modeli denklemi, Denklem (4.24)'te gösterilmiştir.

$$x_t = f(x_{t-1}, u_{t-1}) + w_{t-1} \quad (4.24)$$

Denklem (4.24)'te, x_t belirli bir zamanın geçtiği durumu gösterir ki bu, önceki durum x_{t-1} ile ilgilidir, u_{t-1} kontrol girişidir ve w_{t-1} gürültüdür. Bu çalışmada kullanılan süreç modeli Denklemler (4.25)-(4.30)'da gösterilmiştir.

$$\begin{aligned} x_t &= f(x_{t-1}, u_{t-1}) + w_{t-1} \\ &= [p_t, v_t, q_t, b_t, d_t]^T \\ &\quad + [w_{p,t-1}, w_{v,t-1}, w_{q,t-1}, w_{b,t-1}, w_{d,t-1}]^T \end{aligned} \quad (4.25)$$

Ve,

$$p_t = p_{t-1} + v_{t-1}\delta \quad (4.26)$$

$$v_t = v_{t-1} + C(q_{t-1})(y_{a,t-1} - b_{t-1})\delta - g\delta \quad (4.27)$$

$$q_t = \exp\left(\frac{1}{2}\Omega[y_{w,t-1} - d_{t-1}]\delta\right)q_{t-1} \quad (4.28)$$

$$b_t = \left(1 - \frac{1}{\tau_a}\right)b_{t-1} \quad (4.29)$$

$$d_t = \left(1 - \frac{1}{\tau_w}\right)d_{t-1} \quad (4.30)$$

Denklemler (4.25)-(4.30)'da, $y_{a,t-1}$, $y_{w,t-1}$ zaman adımı $t - 1$ 'de ivmeölçer ve jiroskop ölçüm vektörlerini temsil eder ve $C(q_{t-1})$, q_{t-1} 'in ilişkilendirilmiş dönme matrisidir. Ω , çarpıklık simetrisi olan bir matristir, τ_a ve τ_w zaman sabitleridir, g ise yerçekimi ivmesi vektörüdür. Gürültü vektörü $[w_{p,t-1}, w_{v,t-1}, w_{q,t-1}, w_{b,t-1}, w_{d,t-1}]^T$ işlem gürültüsü eklemek için kullanılır.

Gözlem Modeli: Durum vektörü ile sensör ölçümleri arasındaki bağlantıyı ölçüm modeli yönetmektedir. Bu çalışmada, araç duruş tahminlerindeki hataları sınırlamak için GD-VIO ve GNSS tarafından ölçülen veriler kullanılmaktadır. GD-VIO ve GNSS alıcısından gelen ölçümler mevcut olduğunda, UKF içinde eşzamansız güncellemeler yapılır. GD-VIO okumaları için bir filtre güncellemesi, aracın gözlemlenen hızını kullanır. FOIF GNSS'nin konumu, GNSS verileri için bir filtre güncellemesi üretmek için kullanılır.

Kullanılan metot, iki farklı zamandaki iki GNSS konumunu karşılaştırarak dönme açısını tahmin eder, ardından jiroskop yaw hızını zaman içinde entegre ederek göreceli dönme açısını belirler; bu değer daha sonra jiroskop-GNSS karşılaştırmasından elde edilen başlangıç rostasyon açısı kullanılarak daha da hassas hale getirilir. Bu başlangıç dönme açısı, GNSS, GD-VIO ve IMU ölçümleri gibi diğer sensör verileriyle birlikte UKF'ye entegre edilir. UKF, aracın konumunu, hızını ve duruşunu tahmin ederek hassasiyeti artırmak için doğrusal olmayan bir filtreleme algoritması ve sigma noktaları yaklaşımını kullanır. Dönme açısı tahmini UKF içine entegre edilerek, jiroskop ve GNSS bilgileri etkili bir şekilde birleştirilir, ve böylece daha kesin ve güvenilir bir konumlandırma elde edilir.

GNSS Gözlem Modeli: Veri, GNSS alıcısı tarafından düşük bir frekansta ölçülür (kullanılan GNSS frekans 5 Hz'dir). GNSS yeni veri aldıkça, sensör modeli Denklem (4.31) gibi tanımlanır:

$$y_p = p + n_p \quad (4.31)$$

Denklem (4.31)'de, $y_p = C_E^G[p^X, p^Y, p^Z]$ ve p^X, p^Y, p^Z sırasıyla jeosantrik koordinat sisteminde X, Y ve Z eksilerindeki pozisyonları temsil eder ve n_p , GNSS ölçüm gürültüsü için Gaus modelidir. $[p^X, p^Y, p^Z]$, Denklemler (4.32)-(4.34) kullanılarak elde edilir:

$$p^X = (N + h) \cdot \cos \lambda \cdot \cos \varphi \quad (4.32)$$

$$p^Y = (N + h) \cdot \sin \lambda \cdot \cos \varphi \quad (4.33)$$

$$p^Z = [N \cdot (1 - e^2) + h] \cdot \sin \lambda \quad (4.34)$$

burada, $[\lambda, \varphi, h]$ GNSS tarafından sağlanan enlem, boylam ve irtifayı gösterir. Yukarıda kullanılan parametreler şunlardır: Yeryüzünden çekirdeğe olan uzaklık $N = \frac{a}{1 - e^2 \sin^2 \lambda}$;

Dünya eksantrikliği $e = \sqrt{1 - \frac{b^2}{a^2}}$; ve Dünya elipsoid büyük ve küçük yarıçapları sırasıyla $a = 6,378,137$ (m) ve $b = 63,567,523,142.3142$ (m). $R = \Sigma_{GNSS}$ GNSS ölçüm gürültüsünün kovaryans matrisidir.

GD-VIO Gözlem Modeli: GD-VIO'nun ölçüm modeli Denklem (4.35) ile ifade edilmiştir. Bu denklemde, Δp_{GD-VIO} , GD-VIO'nun araç koordinat sisteminde biriken

delta konum vektörüdür, n_v ise GD-VIO ölçüm gürültüsü için Gaus gürültü modelidir. GD-VIO ölçüm gürültüsü, $N(0, \Sigma_{GD-VIO})$, olarak tanımlanır ve GD-VIO ölçüm gürültüsü kovaryans matrisi $R = \Sigma_{GD-VIO}$ şeklindedir.

$$y_v = p + \Delta p_{GD-VIO} + n_v = p + (p_{GD-VIO_t} - p_{GD-VIO_{t-1}}) + n_v \quad (4.35)$$

GNSS/GD-VIO/IMU füzyonunun algoritması aşağıda verilmiştir. Giriş, sensör verileri ve çıkış, filtrelenmiş araç konum durumunu içerir. Algoritma, parametreleri başlatma, filtre durumunu başlatma, sensör verilerini işleme ve konum tahminini güncelleme adımlarını içerir.

Algoritma: GNSS/GD-VIO/IMU için Algoritma

- 1 Giriş: GNSS, IMU, RGB Görüntü
- 2 Çıkış: Filtrelenmiş Araç Konum Durumu
- 3 Parametreleri başlat: ölçüm gürültüsü, Kovaryans matrisi R , işlem gürültüsü Kovaryans matrisi Q , başlangıç durumu tahmini x_{init} .
- 4 Filtre durumunu başlat: $x_k = x_{init}$.
- 5 Filtre kovaryansını başlat: $P_k = P_{init}$.
- 6 Her zaman adımı k için::
- 7 GNSS verilerini düzeltmek için çoklu adımlı düzeltme filtresini uygula.
- 8 RGB görüntüsü ve IMU ölçümleri kullanarak aracın göreceli durumunu tahmin etmek için GD-VIO'yu uygula.
- 9 Durumu tahmin et: $x_{k,k-1} = f(x_{k-1})$.
- 10 Kovaryansı tahmin et: $P_{k,k-1} = F P_{k-1,k-1} F^T + Q$.
- 11 Kalman kazancını hesapla: $K_k = P_{k,k-1} H^T (H P_{k,k-1} H^T + R)^{-1}$.

Not: Bu denklemde H , ölçüm modeli matrisini temsil eder. Amacı, öngörülen durum tahminini ve sensörlerden elde edilen ölçümler arasında bir ilişki kurmaktır. Matris H , ölçüm değerlerine (bizim durumumuzda Pozisyon ve Hız) karşılık gelen elemanlara 1 değeri atayarak ve geri kalan elemanlara 0 değeri atayarak oluşturulur. Bu düzenleme, durum tahminini

ölçüm uzayına dönüştürmeyi sağlar ve sensör ölçümlerini Kalman Filtresi'nin tahmin sürecine entegre etmeyi kolaylaştırır.

- 12 Durum tahminini güncelle: $x_{k,k} = x_{k,k-1} + K_k(z_k - H_k x_{k,k-1})$.
- 13 Kovaryans tahminini güncelle: $P_{k,k} = (I - K_k H)P_{k,k-1}$.
- 14 Tahmin edilen durum vektörünü x ve kovaryans matrisini P 'yi nihai konumlandırma sonucu olarak çıkar.



5. DENEYSEL SONUÇLAR

Tez çalışması kapsamında, kullanılan otonom sistemlerinin konumlandırmasının geliştirilmesi için iki farklı uygulama gerçekleştirilmiştir. Her bir uygulama için sonuçlar ve tartışma ayrı ayrı sunulmuştur. Tezde, sonraki çalışmalara yönelik öneriler de verilmiştir.

5.1. Uygulama 1 Sonuçları

Önerilen küresel konumlandırma yönteminin performansını değerlendirmek için, pozisyon ve dönmelerin RMSE (Root Mean Squared Error) hesaplanır ve (Baldini ve ark., 2020; Kaiser ve ark., 2016; Mourikis & Roumeliotis, 2007; Leutenegger ve ark., 2015; Bloesch ve ark., 2015; Qin ve ark., 2018; Lynen ve ark., 2013; Forster ve ark., 2016; Chen ve ark., 2019; Almalioglu ve ark., 2022; Rosinol ve ark., 2020; Li & Waslander, 2020) gibi popüler standart ve derin yöntemlerle karşılaştırılır. Diğer bir deyişle, tüm yörüngedeki konum ve rotasyon hataları hesaplanır ve gerçek değerle karşılaştırılır. Regresyon problemleri için RMSE, tahmin edilen ve gerçek değerler arasındaki farkın standart sapması, Denklem (5.1) ile hesaplar. Bu denklemde, x gerçek değeri ve $\hat{x}$ tahmini değeri temsil etmektedir.

$$RMSE = \sqrt{\frac{\sum_{i=1}^l (\hat{x}_i - x_i)^2}{n}} \quad (5.1)$$

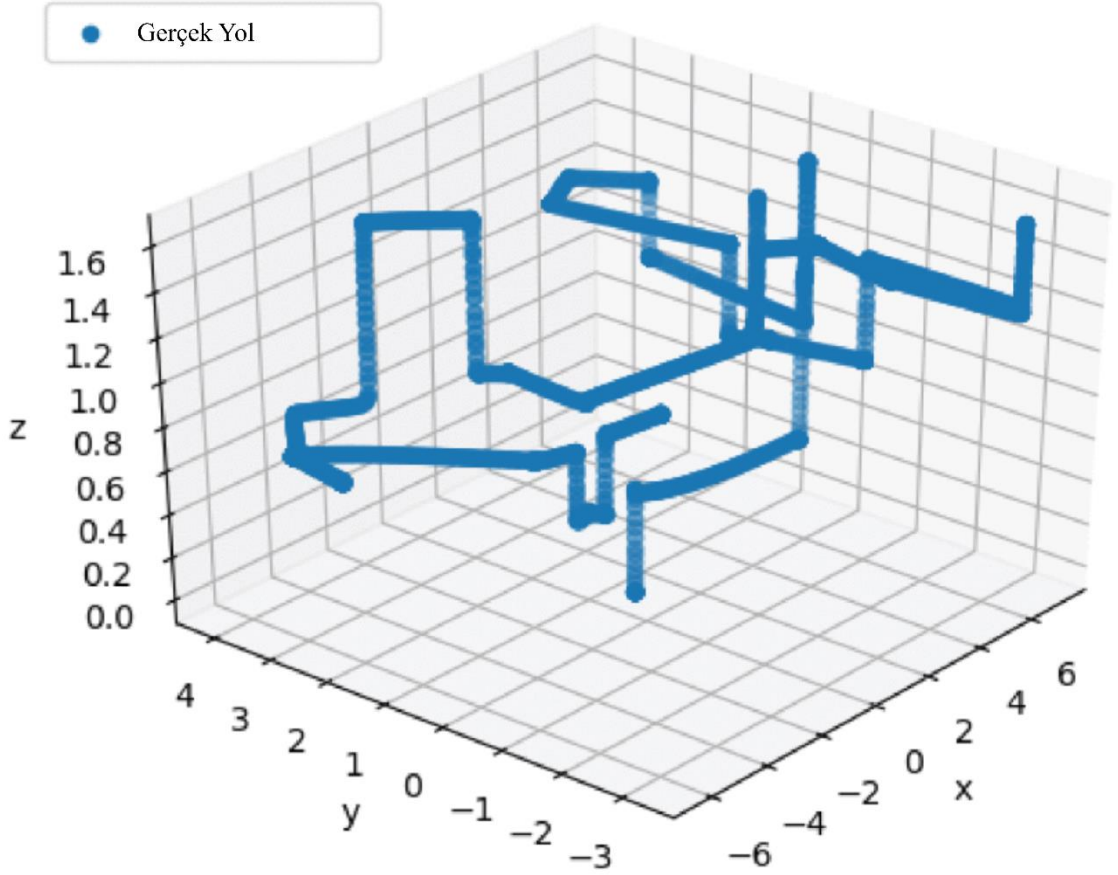
Filtre tabanlı yöntemlerin IMU-LSTM ile karşılaştırıldığında konum tahmini performansına olan etkilerini nicel olarak değerlendirmek için, uçtan uca konum tahmini için her karenin işleme süresi metrik olarak ele alınır.

5.1.1. IMU-LSTM ve IMU-Filtre Karşılaştırması

Derin RCNN tabanlı konum tahmin edicilerin performansını nicel olarak değerlendirmek için, öncelikle IMU-LSTM tabanlı kurulumu, filtre tabanlı (Mahony, Madgwick, EKF, UKF) yöntemlerle karşılaştırılmıştır. Konum tahmini doğruluğu ve zaman verimliliği açısından hem simülasyon hem de genel EuRoC veri setlerinde denemeler yapılmıştır.

5.1.1.1. Simülasyon denemesi

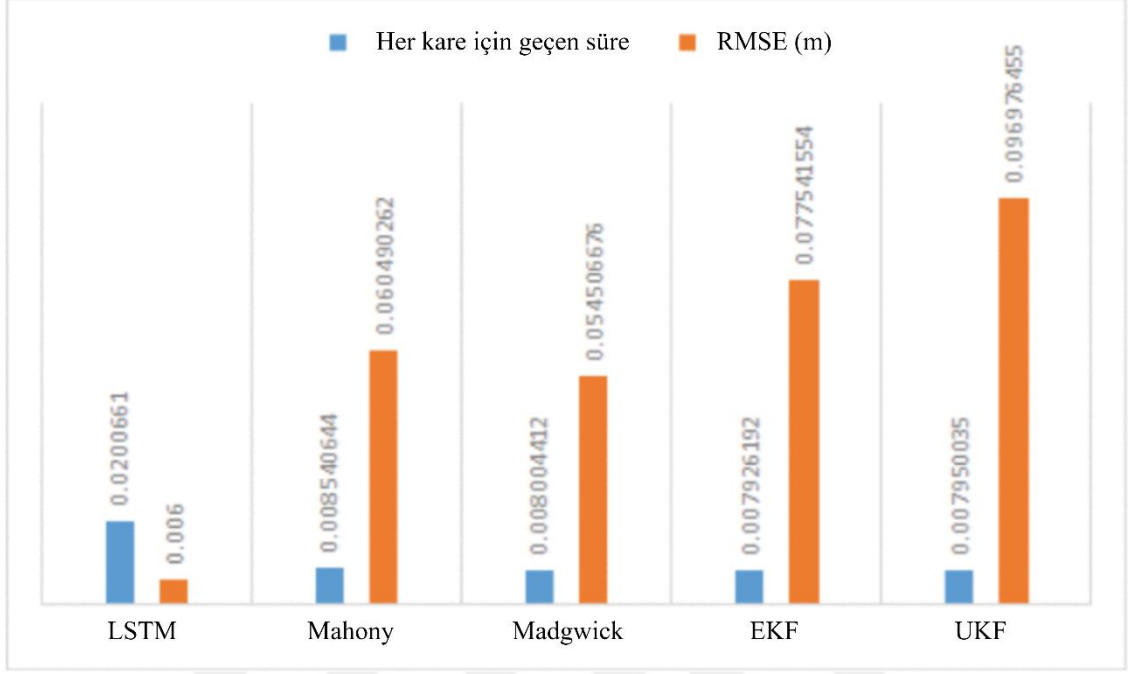
Toplamda 2432 kamera karesi ve 24320 IMU ölçümü bulunan simülasyon ortamında, Tello İHA tarafından taşınan 3B gerçek hareket EuRoC veri seti tarafından gösterilmektedir (Şekil 5.1). Şekil 5.1’de görüldüğü gibi, yönlendirme değişikliği sırasındaki yörünge çok keskin kenarlıdır, bu da ağ için öğrenme durum tahminini bir şekilde zorlaştırmaktadır.



Şekil 5.1. Test edilen simülasyon ortamının Gerçek Yol Yörüngesi.

Şekil 5.2’de, simülasyon veri setinde farklı ağ kurulumları ile Derin RCNN tabanlı konum tahmin yönteminin karşılaştırma sonucu verilmiştir. Burada, IMU-LSTM kurulumunun doğruluk açısından filtre tabanlı alternatiflere göre daha iyi performans gösterdiği gözlemlenmektedir. Ancak, filtre tabanlı yöntemler kadar hızlı performans sergilemez. Zaman verimliliği açısından, filtre tabanlı yöntemler daha iyi bir performansa

sahiptir. Özellikle EKF kurulumu en iyi performansa sahip olup, onu UKF takip etmektedir.



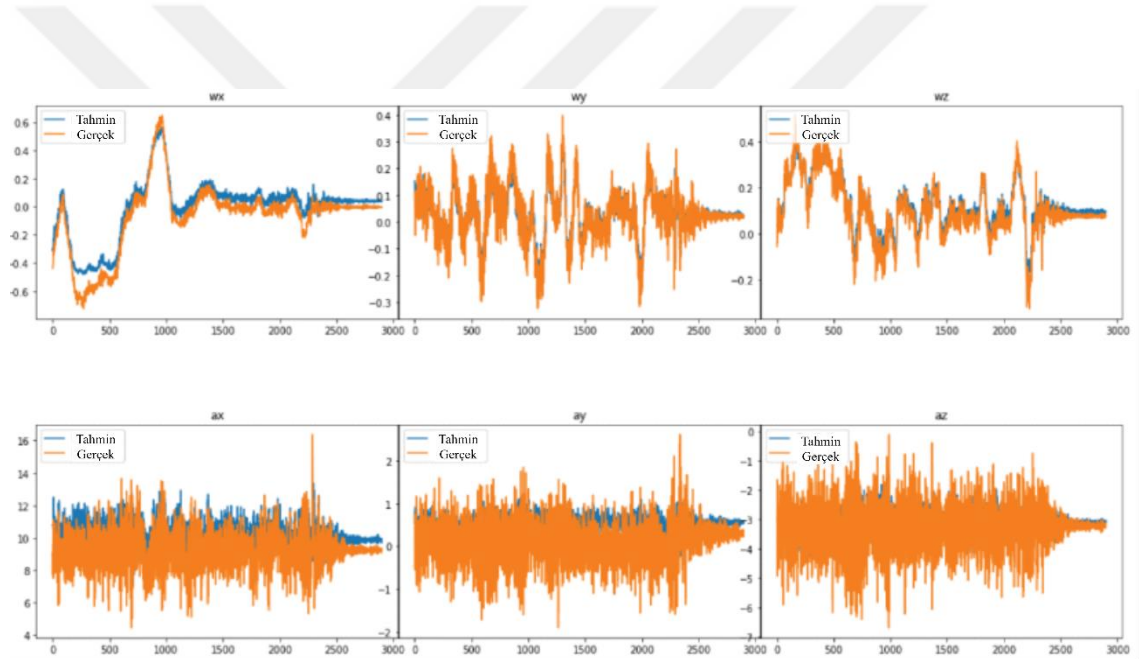
Şekil 5.2. Simülasyon ortamındaki IMU-LSTM tabanlı ve IMU-Filtre tabanlı konum tahmin kurulumunun karşılaştırma sonucu.

5.1.1.2. EuRoC veri seti denemeleri

Bu bölümde, IMU-LSTM kurulumu Derin RCNN tabanlı konum tahmin yönteminin sonuçları, filtre tabanlı Derin RCNN konum tahmin edicileri ile karşılaştırılarak sunulmaktadır. Deneyler, EuRoC very setinde üç zorlu kategoriye (kolay, orta, zor) ait dizilerde gerçekleştirilmiştir. Veri seti, eğitim ve test olmak üzere iki sete ayrılmıştır. Eğitim ve test veri setleri, her bir dizi için sırasıyla %75 ve %25 içermektedir. Model anlayışını kolaylaştırmak için transfer öğrenme ve LSTM modellerimizi ilk olarak sırasıyla görüntüler ve IMU verileri üzerinde test edilmiştir. ResNet50, görüntülerden yeterli özellikleri çıkarmayı başardı. LSTM modeli, IMU ölçümlerini normalleştiren bir şekilde gelecekteki IMU ölçümlerini tahmin edebildi, bu da eğitim sürecini ve konum tahminini daha kararlı hale getirmiştir.

Şekil 5.3, EuRoC veri setinin V1_01 dizisinin IMU-LSTM alt modelinin sonuçlarını göstermektedir. Burada, her biri sırasıyla $(\omega_x, \omega_y, \omega_z)$ ve (a_x, a_y, a_z) ile 3

eksenli olarak temsil edilen gyroskop ve ivmeölçer ölçümleri bulunmaktadır. Turuncu eğri, biashlı IMU ölçümlerini temsil ederken, mavi eğri, IMU-LSTM alt modeli tarafından elde edilen filtrelenmiş IMU ölçümlerini temsil eder. Sonuçlar, IMU-LSTM alt modelinin IMU ölçümleri arasındaki zamansal bağımlılıkları göz önüne alarak normalleştirilmiş IMU veri değerleri sağladığını göstermektedir. Bu normalleştirme süreci, ölçümlerde bulunan IMU gürültüsü ile başa çıkmaya yardımcı olur. En iyi tahmin edilen değerler, tahmin edilen değerlerin daha iyi normalleştirildiği ve tahmin edildiği gyroskop ölçümlerinde görülmektedir. Ivmeölçerin yüksek titreşim gürültüsüne sahip olması nedeniyle ivmeölçerde etki daha azdır. Bu normalleştirilmiş değerler daha sonra Derin RCNN ağının IMU-LSTM kurulumuna, robotun küresel pozisyonunu tahmin etmek için IMU girişi olarak kullanılır.

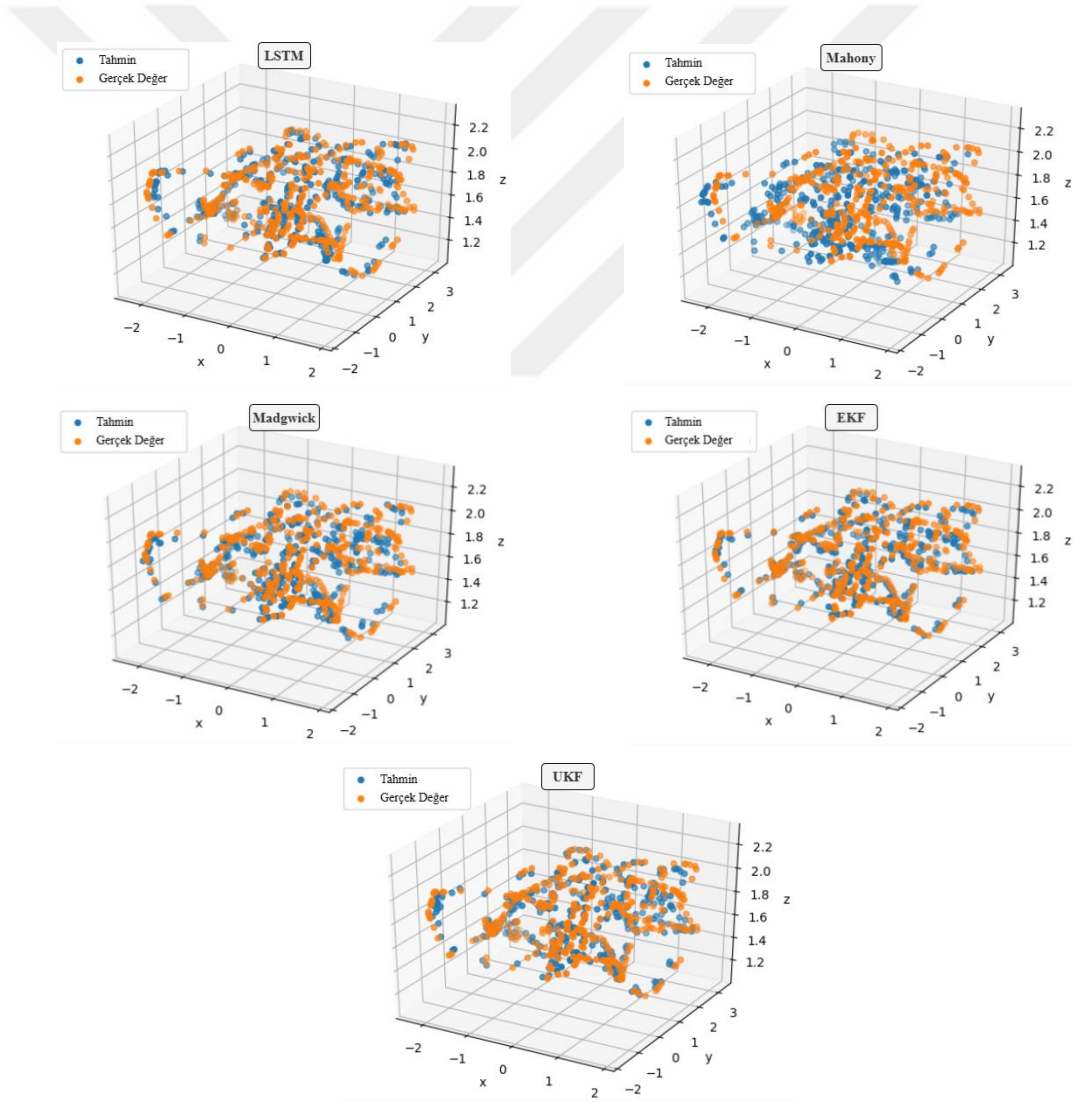


Şekil 5.3. EuRoC veri setinin V1_01 dizisi üzerinde IMU-LSTM alt modelinin sonuçları.

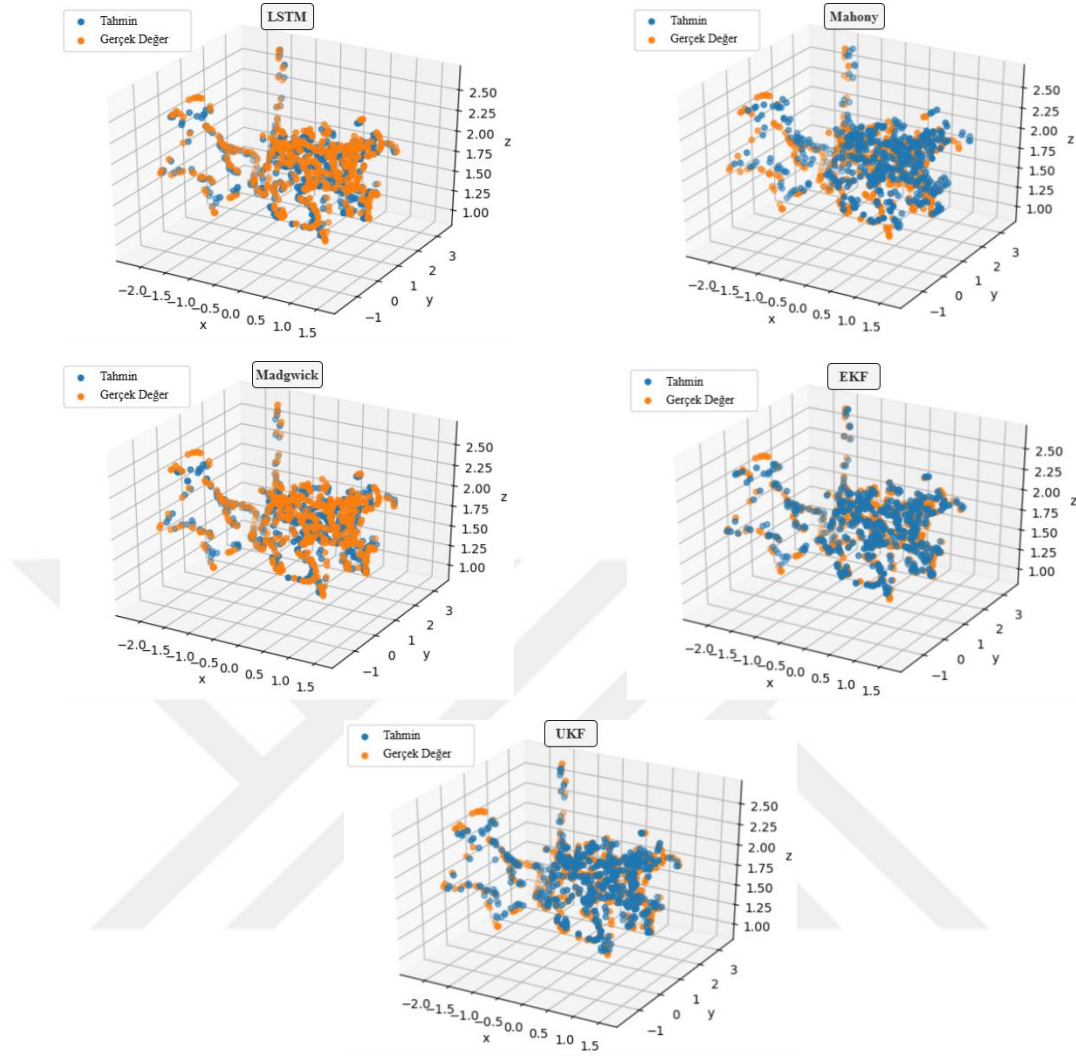
Şekil 5.4 ve Şekil 5.5'de, EuRoC veri setinin V1_02 dizisinde Derin RCNN tabanlı konum tahmin mimarisinin IMU-LSTM, IMU-Mahony, IMU-Madgwick, IMU-EKF ve IMU-UKF kurulumlarından elde edilen 3B yörüngelerin karşılaştırması bulunmaktadır. V1_03_zorlu dizisinin daha fazla aydınlatma tutarsızlıkları ve hareket hızından kaynaklanan bulanık görüntüler içermesi nedeniyle, tüm 5 kurulumun her iki dizide de oldukça iyi sonuçlar sunduğu gözlemlenmektedir.

Çizelge 5.1. EuRoC veri seti üzerinde IMU-LSTM tabanlı ve IMU-Filtre tabanlı pozisyon tahmini kurulumlarının karşılaştırma sonucu.

	V1_01	V1_02	V_03	V2_01	V2_02	V2_03
<i>Time Taken Per Frame (ms)</i>						
LSTM	8.41669147	11.5215215	10.2330735	13.0703139	9.473889344	13.8395459
Mahony	8.317947056	11.07478142	9.892942327	9.815794899	9.500358344	11.86350998
Madgwick	11.38009481	11.06065725	9.847396251	11.81925767	11.68309611	10.96841642
EKF	8.321962038	11.01558874	9.841596087	9.989095457	11.73014831	10.94181507
UKF	8.271849039	10.91511267	9.848773028	9.599138438	9.124278197	10.57100549
<i>RMSE (m)</i>						
LSTM	0.017258338	0.086499608	0.064554869	0.021935895	0.042341781	0.059033888
Mahony	0.171943049	0.254456863	0.093865869	0.036503232	0.05348064	0.165161187
Madgwick	0.03744389	0.090967237	0.068016986	0.110042218	0.068958965	0.085359013
EKF	0.029492422	0.073287377	0.052987854	0.088692036	0.070216253	0.082095109
UKF	0.045078974	0.07922591	0.062388228	0.046098865	0.069659343	0.08512485



Şekil 5.4. IMU-LSTM tabanlı ve IMU-Filtre tabanlı pozisyon tahmini kurulumlarının EuRoC V1_02 dizileri üzerindeki yörünge karşılaştırması.



Şekil 5.5. IMU-LSTM tabanlı ve IMU-Filtre tabanlı pozisyon tahmini kurulumlarının EuRoC V1_03 dizileri üzerindeki yörünge karşılaştırması.

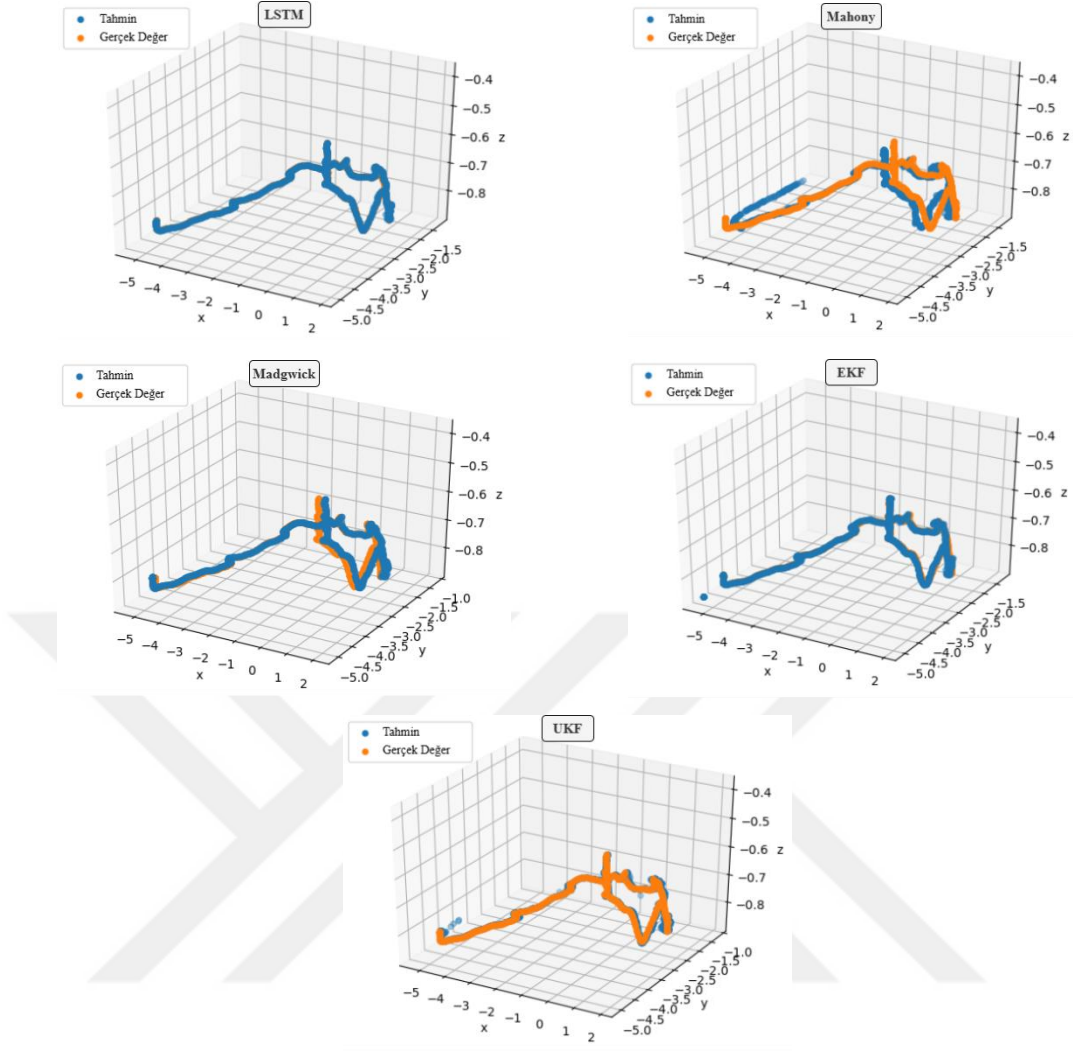
5.1.1.3. Gerçek Dünya Deneyimi

Bu deney, Bölüm 4.1.3.'de bahsedilen bilinen laboratuvar ortamında gerçekleştirilmiştir ve verilerin yüzde 75 ve yüzde 25 oranlarında eğitim ve test dizilerine ayrıldığı diğer yukarıda bahsedilen deneylerle benzer bir şekilde gerçekleşmiştir. Çizelge 5.2'de gösterildiği gibi, zaman verimliliği ve RMSE açısından Derin RCNN'nin beş farklı kurulumu arasında performans karşılaştırması yapılmıştır. Sonuçlar, benzer şekilde simülasyon ve EuRoC deneylerinde olduğu gibi, ağırlıklı olarak IMU-LSTM kurulumunun doğruluk açısından daha iyi performans gösterdiğini, ancak işleme süresinin açısından filtre tabanlı alternatiflerine kıyasla daha yavaş olduğunu göstermektedir.

Çizelge 5.2. IMU-LSTM tabanlı ve IMU-Filtre tabanlı pozisyon tahmini kurulumlarının gerçek zamanlı deneme üzerindeki karşılaştırma sonuçları.

	Görüntü Başına Geçen Süre	RMSE
LSTM	10.38149954	0.014656638
Mahony	10.19134924	0.163732653
Madgwick	10.33280875	0.031682532
EKF	10.18071596	0.048106454
UKF	10.28413951	0.028917899

Çalışılan ortama göre konumlandırma performansını analiz etmek için, beş farklı Derin RCNN kurulumundan elde edilen tahmin edilen konumun zemin gerçeğe karşılaştırıldığı bir yörünge görselleştirmesi sunulmaktadır. Şekil 5.6, bu deneyin sonuçlarını göstermektedir. Tahmin edilen konumlar mavi bir yörünge olarak gösterilirken, zemin gerçekleri turuncu bir yörünge olarak gösterilir. Bu, beş kurulumun performanslarının birbirine benzer olduğunu, hatta Şekil 4.2'de görüldüğü gibi bozulmuş monoküler görüntülerin varlığına rağmen küçük doğruluk farkları ile olduğunu göstermektedir. IMU-LSTM kurulumu minimum hata oranını gösterirken, IMU-Mahony diğer kurulumlara kıyasla daha yüksek bir hata oranına sahiptir.



Şekil 5.6. IMU-LSTM tabanlı ve IMU-Filtre tabanlı pozisyon tahmini kurulumlarının gerçek dünya deneyimi üzerindeki yörünge karşılaştırması.

5.1.2. Literatürle Karşılaştırma

Literatürdeki en yakın ilgili yöntemlere dair empirik bir karşılaştırma sunmak amacıyla, yakın dönemde geliştirilmiş geleneksel ve öğrenmeye dayalı VO, VIO ve V-SLAM yöntemleri (Baldini ve ark., 2020; Kaiser ve ark., 2016; Mourikis & Roumeliotis, 2007; Leutenegger ve ark., 2015; Bloesch ve ark., 2015; Qin ve ark., 2018; Lynen ve ark., 2013; Forster ve ark., 2016; Chen ve ark., 2019; Almalioglu ve ark., 2022; Rosinol ve ark., 2020; Li & Waslander, 2020) ile önerilen yaklaşım karşılaştırılmıştır. Geleneksel yöntemler arasında SVOMSF (Kaiser ve ark., 2016), MSCKF (Mourikis & Roumeliotis, 2007), OKVIS (Leutenegger ve ark., 2015), ROVIO (Bloesch ve ark., 2015), VINSMONO (Qin ve ark., 2018), VINSMONO+LC (Loop Closure) (Lynen ve ark.,

2013), SVOGTSAM (Forster ve ark., 2016), ORB-SLAM2 (Mur-Artal, & Tardós, 2017), STCM-SLAM (Chen ve ark., 2019) bulunurken, öğrenmeye dayalı yöntemler arasında (Almalioglu ve ark., 2022), Kimera (Rosinol ve ark., 2020) ve Baldini ve ark. (2020) ile Li ve Steven (2020) tarafından önerilen yöntemler bulunmaktadır. Bu değerlendirme için kullanılan veri seti, EuRoC'un V1_01-03, V2_01-03 dizileridir.

Çizelge 5.3, bu diziler için karşılaştırmalı sonuçları göstermektedir. Önerilen mimarinin, konum tahmini doğruluğu açısından mevcut yöntemlere göre sürekli olarak daha iyi performans sergilediği gözlemlenmiştir. Özellikle, daha zorlu ve zorlayıcı dizilerde, yani özelliiksiz ve bulanık kamera karelerini içeren dizilerde en yüksek iyileme elde edilmiştir. Görsel kaliteli ve özellikli görüntüleri işleyen geleneksel yöntemler, geometrik algoritma sınırlamaları nedeniyle tam sensör bozulması ve görüntü örtülmeleri ile başa çıkamazlar ve zaman bilgisinden faydalanamazlar. Gözlemlenen sonuçlardan çıkarılacak sonuç, geleneksel görsel tabanlı konum tahmini yöntemlerinin, örtülü veya özelsiz görsel verilerin varlığında ve IMU ölçümleri sıkıca senkronize edilmediğinde konumlandırma için uygun bir yaklaşım olmadığıdır. Ayrıca, önerilen mimarinin tahmin doğruluğunun, Çizelge 5.3'te listelenen öğrenmeye dayalı yöntemlerden üstün olduğunu belirtmek önemlidir. Önerilen yöntemin, belirli bir ölçüde görünmeyen senaryolara genelleme yapmasına yardımcı olan zamansal kısıtlamaları hesaba katabildiğine dikkat edilmelidir.

Çizelge 5.3. EuRoC (V1_01-03, V2_01-03) dizileri için karşılaştırmalı sonuçlar.

Çalışma	Metod	V1_01	V1_02	V1_03	V2_01	V2_02	V2_03
Baldini	Derin Öğrenme	<u>0.03</u>	0.05	0.1	<u>0.05</u>	<u>0.05</u>	<u>0.08</u>
SVOMSF	Semi-Direct & Filter-based VIO	0.4	0.63	-	0.2	0.37	-
msckf	Filter-based VIO	0.34	0.2	0.67	0.1	0.16	1.13
OKVIS	Keyframe-based VI-SLAM	0.09	0.2	0.24	0.13	0.16	0.29
ROVIO	Filter-based VIO	0.1	0.1	0.14	0.12	0.14	0.14
Vins-mono	Optimization-based VIO	0.07	0.1	0.13	0.08	0.08	0.21
Vinsmonolc	Optimization-based VIO	0.04	<u>0.06</u>	0.11	0.06	0.06	0.09
SVOGTSAM	Semi-direct & Graph Optimization based VIO	0.07	0.11	-	0.07	-	-
ORB-SLAM2	Feature-based V-SLAM	0.087	0.065	0.092	0.071	0.061	0.184
STCM-SLAM	Keyframe-based VI-SLAM	0.079	0.113	0.12	0.084	0.06	0.099
SelfVIO	Derin Öğrenme	0.08	0.09	0.1	0.11	0.08	0.11

Kimera	Metric-Semantic VI-SLAM	0.05	0.08	0.07	0.08	0.1	0.21
Li & Waslander	Derin Öğrenme	2.07	2.2	2.83	1.49	1.49	2.22
Önerilen IMU-LSTM	Derin Öğrenme	0.01	0.0864	0.0645	0.02193	0.0423	0.0590

5.1.3. Eksiltme çalışması ve sonuçların değerlendirilmesi

Her bir modülün etkinliğini ve Derin RCNN'nin genel mimarisini göstermek amacıyla, EuRoC veri seti, simülasyon ortamı ve gerçek dünya deneylerinde önerilen yöntem için eksiltme (ablation) çalışması yapılmıştır. Eksiltme çalışması, bir modelin performansını anlamak ve modelin hangi özelliklerinin veya bileşenlerinin katkı sağladığını belirlemek amacıyla yapılan bir tür analizdir. Bu çalışma tipinde, bir modelden sırasıyla bir veya daha fazla özellik (veya katman) çıkarılır, ve modelin performansındaki değişiklikler incelenir. Bu sayede, modelin hangi özelliklere veya bileşenlere ne kadar bağımlı olduğu, sistem üzerindeki etkisi ve önemi anlaşılır. Şekil 5.3 – 5.6 ve Çizelge 5.1 – 5.3, performans sonuçların en iyisini kalın olarak ve ikinci en iyiyi altı çizili olarak göstermektedir.

İlk olarak, önerilen yöntem bir İHA'nın simülasyon ortamında konum tahmini üzerinde değerlendirilmiştir (Bölüm 4.2.4.1). Bu, Derin RCNN ağını simülasyon ortamından elde edilen veri setini kullanarak beş farklı kurulumda eğitmeyi ve eğitilmiş modeli aynı simülasyon ortamında bir drone'un konum tahmini için test etmeyi içerir. Şekil 5.2'den görülebileceği gibi, simülasyon ortamı birçok benzer veya hiç duvar dokusuna sahip olmasına rağmen (geleneksel geometrik yöntemlerin özellik çıkarma ve eşleştirme ile başa çıkma konusunda zorlandığı yerler), ağ kurulumları hala oldukça yüksek doğruluk göstermektedir ve oldukça düşük işleme maliyetine sahiptir. Sonuçlar aynı zamanda, IMU-LSTM kurulumunun IMU'nun zamansal özelliklerinden faydalanabileceğini ve bu nedenle RMSE hatasını azaltabileceğini göstermektedir, bu da filtre tabanlı alternatiflere göre. Diğer taraftan, filtre tabanlı yöntemler, her bir çerçeveyi işlemek için geçen süreyi azaltırken LSTM alternatiflerine göre RMSE'yi artırma maliyetiyle gelmektedir.

İkinci olarak, Derin RCNN mimarisinin etkinliği EuRoC adlı yaygın bir iç mekan referans veri setinde karşılaştırılmıştır. Bu veri seti üç farklı zorluk seviyesini içerdiği için odak noktası, mimarinin bulanık görüntüler ve yüksek hızlı hız değişiklikleri gibi daha

zorlu görsel verilerle başa çıkma yeteneğini test etmek olmuştur. Sonuçlar Çizelge 5.1 ve Şekil 5.4'te gösterilmiştir. Görüleceği gibi, veri seti çok bulanık görüntüler içeriyordu ve görsel veri parlaklığı aşırı pozlanmış veya çok düşük olduğu durumlarda, Derin RCNN ağı RMSE hatasını bir ölçüde azaltabiliyor, bu da çeşitli aydınlatmalarda yüksek hareket hızlarıyla başa çıkma yeteneğine sahip olduğunu göstermektedir. Beş ağ kurulumunu karşılaştırırken, simülasyon ortamının genel sonucu burada da geçerlidir, IMU-LSTM IMU verilerindeki zamansal bağımlılığı kullanır ve bu nedenle filtre tabanlı kurulumlara göre daha iyi RMSE'ye sahiptir. Ancak filtre tabanlı yöntemler, zaman etkinliği durumunda daha iyi performans göstermektedir.

Son olarak, DJI Tello İHA'nın bir iç mekan laboratuvar ortamında gerçekleştirilen bir gerçek dünya deneyiyle, Derin RCNN mimarisinin bir İHA'nın gerçek zamanlı küresel konumlandırma üzerindeki etkisini incelemek için bir deney yapılmıştır. Bu deneyde, drondan alınan verilerin İHA bileşenlerinin kalitesine ve İHA ile uzaktan işlem birimi arasındaki bağlantıya bağlı olduğu gözlemlenmiştir. Örneğin, deneyde DJI Tello İHA'nın verileri oldukça senkronize olmamıştır ve görsel veri bozulmuştur (Şekil 4.2). Ancak, Çizelge 5.2 ve Şekil 5.6'da gösterildiği gibi, sonuçlar göstermektedir ki bu görsel bozulma doğruluğu olumsuz yönde etkilememekte ve Derin RCNN mimarisi bununla başa çıkabilmektedir. Ayrıca, filtre tabanlı kurulumlara kıyasla, IMU-LSTM versiyonu konum tahmini doğruluğuna ek bir katkı sağlamaktadır. Önceki paragrafta belirtildiği gibi, nedeni IMU-LSTM'nin IMU ölçümleri arasındaki pencere dizileri boyunca zamansal korelasyonları ve önceki deneyimleri entegre etmesidir. Ayrıca, filtre tabanlı kurulumların genellikle zaman etkinliği açısından IMU-LSTM'den daha iyi performans gösterme eğiliminde olduğunu belirtmek önemlidir. Filtre tabanlı kurulumlar arasındaki performans farkı çeşitli testler için değişmiştir ve ihmal edilebilmektedir.

5.2. Uygulama 2 Sonuçları

5.2.1. GD-VIO Sonuçları

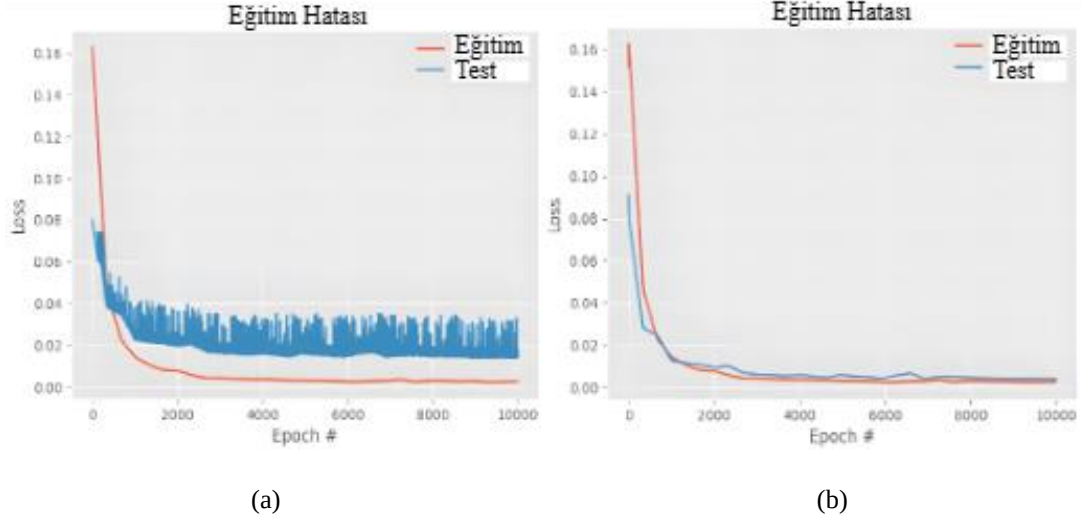
Deneyisel değerlendirme, ilk olarak denemede kullanılan veri setlerinin genel bir değerlendirmesi ile başlanır, ardından sonuçları değerlendirmek için kullanılan ölçütlerin açıklamasını yapılır. Ayrıca, önerilen GD-VIO'nun aşırı öğrenme sorununu nasıl çözebildiği gösterilmektedir. İkinci olarak, önerilen GD-VIO'nun deneyisel sonuçları KITTI veri seti kullanılarak göstermektedir. Son olarak, önerilen GD-VIO'yu mevcut konum tahmini teknolojileriyle karşılaştırarak yöntem etkisi gösterilmiştir.

Değerlendirme Metriği: ATE (Absolute Trajectory Error) ve RPE (Relative Position Error) olarak bilinen ortalama karesi, VO veya görsel SLAM'ın doğruluğunu değerlendirmek için sıkça kullanılan değerlendirme metrikleridir. Bu hesaplamalar, Denklem 5.2 ve Denklem 5.3 kullanılarak gerçekleştirilir. Burada, N toplam kare sayısını temsil eder ve ΔR_i ile Δt_i , tahmini konum x'_i ile gerçek zemin x_i arasındaki dönme ve pozisyon hatasını gösterir. Önerilen GD-VIO yönteminin performansını değerlendirmek için bu değerler hesaplanır, yaygın standartlar ve derin öğrenme temelli tekniklerle karşılaştırılır.

$$ATE_{rot} = \left(\frac{1}{N} \sum_{i=0}^{N-1} \|\Delta R_i\|^2 \right)^{\frac{1}{2}} \quad (5.2)$$

$$ATE_{tra} = \left(\frac{1}{N} \sum_{i=0}^{N-1} \|\Delta t_i\|^2 \right)^{\frac{1}{2}} \quad (5.3)$$

Eğitim Detayları: Bu model, Uygulama 1'e göre farklı özellikli sistemde eğitilmiştir. Bu uygulamada eğitim, Intel(R) Core(TM) i7-9750H CPU @ 2.60GHz işlemci, 32 GB DDR3 RAM ve bir NVIDIA Quadro RTX 4000 GPU içeren bir sistem üzerinde gerçekleştirilmiştir. Ağ, Google Brain ekibi tarafından geliştirilen Tensorflow System kullanılarak uygulandı ve eğitilmiştir. Tüm ön işleme verileri Python ve Tensorflow uyumlu modüller kullanılarak gerçekleştirilmiştir. Bellek sınırlamaları ve hesaplama süresi nedeniyle görüntüler $128 \times 128 \times 3$ boyutuna küçültüldü, bu sayede ağ daha büyük görüntü boyutlarında daha iyi performans sergileyebilmektedir. Toplamda 10000 epoch boyunca, tüm ağlar 128'lik bir grup boyutunu kullanarak eğitildi ve her veri seti için 6 gün sürmüştür. Eğitim parametrelerini optimize etmek için Adam yöntemi kullanılmıştır, azalma ve öğrenme hızı her ikisi de 0.0001 değerine sahiptir ve hata fonksiyonu MAE'dır.



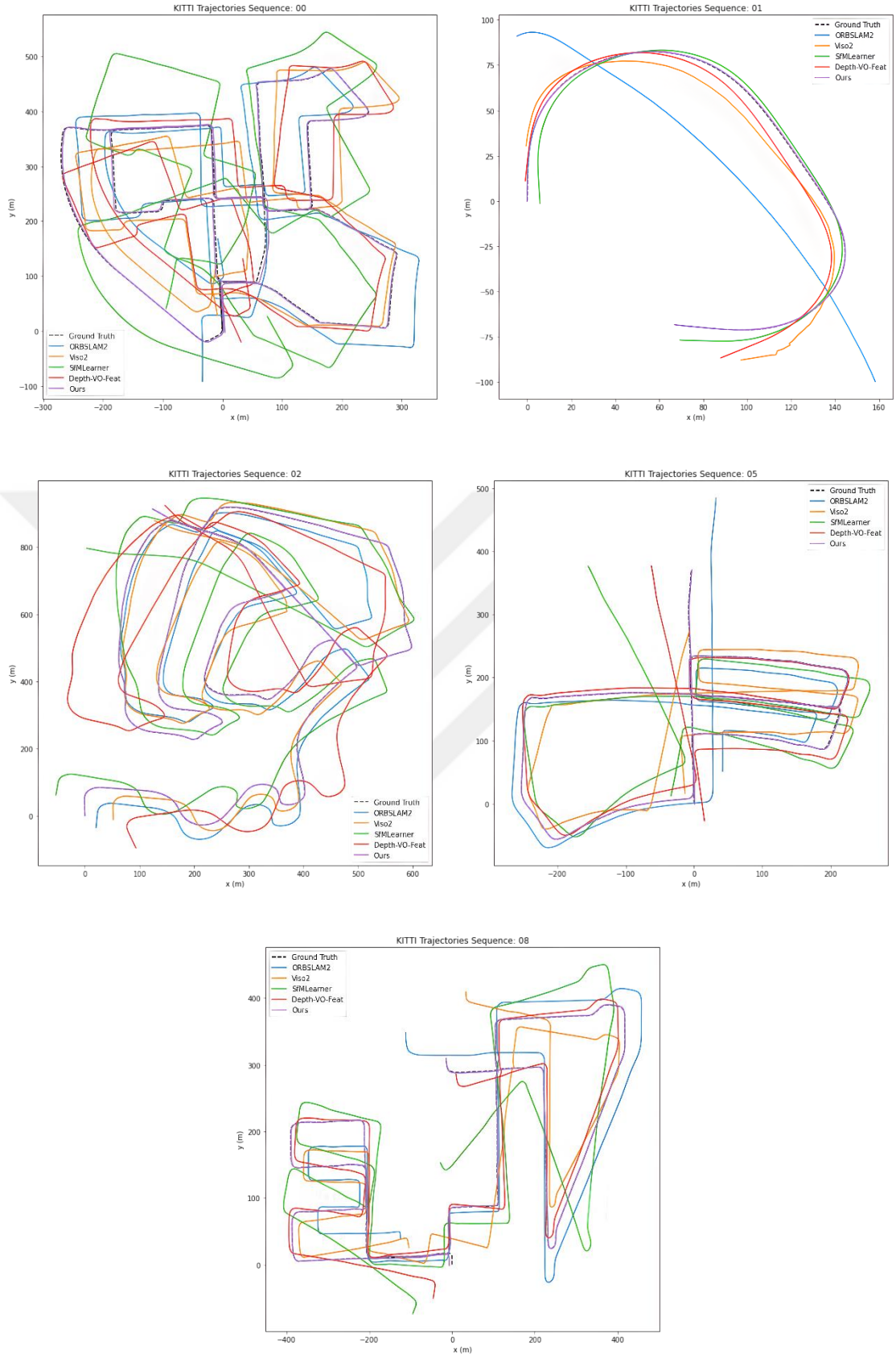
Şekil 5.7. Eğitim ve test kaybını Siamese Tipi CNN + Lambda katmanı ile ve olmadan karşılaştırın. Sol: Aşırı öğrenme ile hata. Sağ: İyi öğrenme hata.

GD-VIO'nun geçerliliğini göstermek için, KITTI veri seti için iki ağ yapısı altında ağ model kaybını karşılaştırılmıştır. İki ağın yapısı Siamese tip CNN ve lambda fusion katmanı dışında aynıdır. İlk ağ, her iki giriş görüntüsü için yalnızca bir CNN kullanır ve lambda fusion katmanını içermez. İkinci ağ, Şekil 4.7'de görüldüğü gibi önerilen çerçevedir. Şekil 5.7 (a)'da görüldüğü gibi, Siamese tip CNN + lambda füzyon katmanı olmadan ağ model kaybının eğitim verilerinde daha az olduğu ancak test verilerinde büyük hata yaşadığı, bu da aşırı uydurmaya yol açtığı görülmektedir. Bu da, Görünmeyen ve yeni ortamlarda, düşük performansa neden olur, ancak önerilen ağ yapısı bu sorunu etkili bir şekilde çözebilir. Şekil 5.7 (b)'de görüldüğü gibi, Siamese tip CNN + lambda füzyon katmanı kullanıldığında test verilerinde ağ model kaybı daha küçüktür.

Sonuç olarak, VO modeli KITTI veri setinde test edilmiştir ve önerilen yöntemin bazı durumlarda, mevcut en iyi yöntemlerden daha iyi olduğu popüler VO yöntemleriyle karşılaştırılarak gösterilmiştir. Geleneksel yaklaşım için ORB-SLAM2 (Mur-Artal & Tardós, 2017) ve VISO2 (Geiger ve ark., 2011)'yi kullanırken, derin öğrenme tabanlı yaklaşımlar için sfmLearner (Zhou ve ark., 2017) ve Depth-VOFeat (Zhan ve ark., 2018) kullanılmaktadır. Eğitim veri seti olarak, KITTI veri setinin (00, 01, 02, 05 ve 08) dizileri kullanılmıştır. Eğitilen model, (03, 04, 06, 07, 09 ve 10) diziler üzerinde test edilmiştir. Önerilen modelin genelleme yeteneğini değerlendirmek için, modelin görünmeyen ortamlarda iyi performans gösterme yeteneği olan, deneysel testler iki bölüme ayrılmıştır. İlk olarak, modelin öğrenme yeteneğini göstermek için eğitilmiş diziler üzerinde

karşılaştırma sonuçları gösterilmiştir. İkinci olarak, modelin genelleme yeteneğini göstermek için modelin görmediği test dizilerinde karşılaştırma sonuçları elde edilmiştir.

Çizelge 5.4 ve Şekil 5.8, RPE ve ATE bakımından eğitilmiş diziler (00, 01, 02, 05, 08) üzerinde sayısal değerlendirme sonuçlarını ve yörüngeleri göstermektedir. En iyi sonuçlar kırmızı ile gösterilmiş ve ikinci en iyi sonuçlar altı çizilmiştir. Bu sonuçlarda, tüm derin öğrenme tabanlı yöntemlerin bu diziler üzerinde VO öğrenmek için eğitildiği gözlemlenmektedir. Çizelge 5.4'te görüldüğü gibi, önerilen yöntem diğer yöntemleri büyük ölçüde geride bırakmaktadır. Bu yöntemde, VO özelliklerini optimal bir şekilde öğrenmiş ve tüm eğitilmiş dizilerde karşılaştırılabilir odometri sonuçları verilmiştir. ATE değerleri, dizideki kare sayısına bağlıdır ve özellikle geometrik yöntemlerde ORB-SLAM2 gibi veri setinin boyutuna bağlı olarak artmaktadır. RPE değerleri bu kümülatif sapma sorunlarına bağlı değildir ve bu durumda daha iyi değerlendirme ölçütleridir. ORB-SLAM2'nin döngü kapanışını gerçekleştirebilme yeteneği, özellikle uzun yörüngelere sahip dizilerde döngüler içeren yörüngelerde hata oranlarını büyük ölçüde azaltır, bu da yörüngelerdeki hata oranlarını özellikle diziler 00 ve 05'de bu sonuçlar görülebilir.



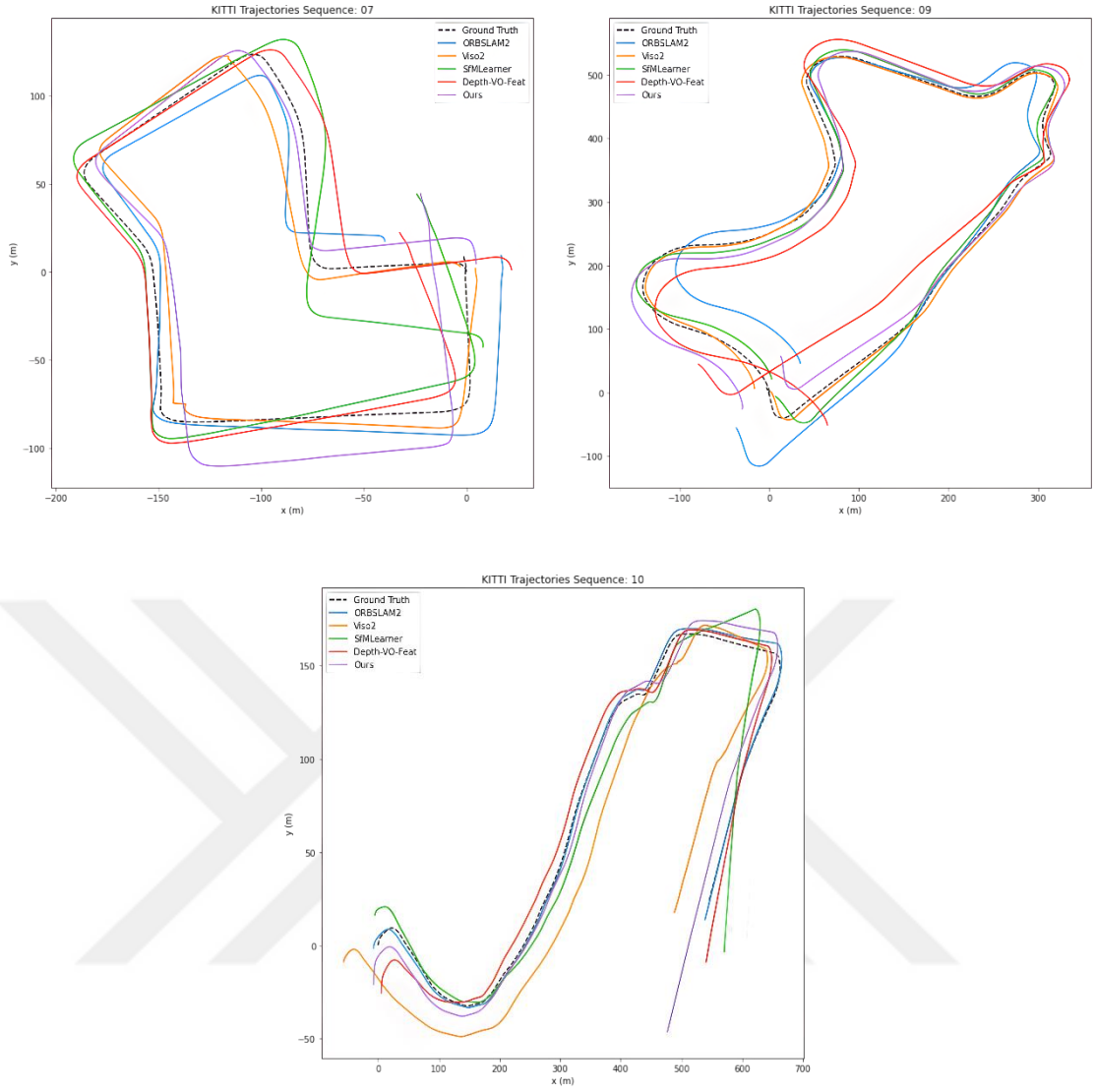
Şekil 5.8. ORB-SLAM2, VISO2, SfMLearner, Depth-Vo-Feat ve önerilen yöntemin eğitilmiş sekeanslarda (00, 01, 02, 05, 08) ölçek hizalaması sonrasında yörüngeleri.

Çizelge 5.4. Eğitilmiş sekanslarda (00, 01, 02, 05, 08) sayısal ATE (metre) ve RPE (derece) sonuçları.

Sek.		ORB-SLAM2	VISO2	SfMLearner	Depth_VO_Feat	GD-VIO
00	ATE	<u>47.273</u>	85.7245	106.081	61.917	2.6165
	RPE-t	0.1073	0.0948	0.0909	<u>0.0688</u>	0.0064
	RPE-r	<u>0.0058</u>	0.0269	0.0335	0.0241	0.0022
01	ATE	42.0219	19.5341	<u>5.0333</u>	8.8063	0.3568
	RPE-t	0.8932	0.2459	<u>0.055</u>	0.1199	0.0043
	RPE-r	0.5723	0.2861	<u>0.021</u>	0.1013	0.001
02	ATE	<u>48.4153</u>	50.6395	72.3044	88.2671	1.026
	RPE-t	0.0933	<u>0.0519</u>	0.06	0.0636	0.0022
	RPE-r	<u>0.0025</u>	0.0118	0.0194	0.0224	0.005
05	ATE	34.3109	71.9748	41.9149	<u>21.8217</u>	1.05
	RPE-t	0.0969	0.1205	<u>0.0663</u>	0.0997	0.0549
	RPE-r	<u>0.0021</u>	0.0353	0.0233	0.0226	0.0192
08	ATE	43.9227	54.6779	57.4341	<u>25.131</u>	1.9511
	RPE-t	0.1336	0.1407	0.0811	<u>0.0562</u>	0.0059
	RPE-r	<u>0.0029</u>	0.0247	0.0259	0.0233	0.0024

İkinci deneyde, önerilen modelin test dizileri (03, 04, 06, 07, 09, 10) üzerinde test edilmiştir ve yöntemin genelleme yeteneği değerlendirilmiştir. Bu testlerde, SfMLearner ve Depth-VO-Feat gibi derin öğrenme yöntemleri, 09 ve 10 dizileri hariç bu dizilerde eğitildiği için iyi performans sergilemelidir. Bununla birlikte, önerilen modelin bu test dizilerinde eğitilmemiştir, bu yüzden burada ilk deneyde eğitilen model kullanılmıştır. Şekil 5.9, ORB-SLAM2, Viso2, SfMLearner, Depth-VO-Feat ve önerilen yöntemin test dizilerindeki yörünge sonuçlarını göstermektedir. Yeni ve görülmemiş dizilere rağmen önerilen modelin karşılaştırılabilir şekilde iyi performans gösterdiği görülmektedir. Örneğin, dönme sapmalarının çok olduğu 04 dizisinde, rekabetçi derin öğrenme tabanlı yöntemlerin ve hatta geometrik yöntem Viso2'nin iyi performans göstermediği açıkça görülmektedir. Ancak önerilen model sapmaları işleyebilmektedir ve sonuçlar ORB-SLAM2 ile karşılaştırılabilmektedir.

Tüm test dizilerinin nicel bulguları Çizelge 5.5'de açıklanmıştır. Tüm modellerin küçük ve kısa dizilerde iyi performans sergilediği, ancak içinde yüksek dönme olan daha büyük dizilerde zorlandığı gösterilmiştir. Önerilen model, test dizilerinin modele yeni olmasına rağmen çok iyi sonuçlar göstermektedir. Sistem sınırlamaları nedeniyle önerilen model, SfMLearner ve Depth-VO-Feat tarafından sırasıyla (128 × 416) ve (608 × 160) çözünürlükte kullanılan RGB resimlerine göre daha küçük bir gri tonlamalı giriş görüntüsü kullanmaktadır.



Şekil 5.9. Test sekanslarında (03, 04, 06, 07, 09, 10) ORB-SLAM2, VISO2, SfMLearner, Depth-Vo-Feat ve önerilen yöntemin ölçek hizlaması (Scale Alignment) sonrasındaki yörüngeleri.

Çizelge 5.5. Test sekanslarında (03, 04, 06, 07, 09, 10) sayısal ATE (metre) ve RPE (derece) sonuçları.

Sek.		ORB-SLAM2	VISO2	SfMLearner	Depth_VO_Feat	GD-VIO
03	ATE	1.5308	2.3346	10.2203	14.24	13.4
	RPE-t	0.016	0.0644	0.0992	0.2014	0.1516
	RPE-r	0.0013	0.0194	0.0478	0.1013	0.1534
04	ATE	1.4075	5.5588	2.9668	1.5141	1.4919
	RPE-t	0.0136	0.0282	0.0462	0.0253	0.0272
	RPE-r	0.0019	0.0202	0.0197	0.0189	0.0176
06	ATE	55.6568	21.446	12.5757	14.3435	14.032
	RPE-t	0.1708	0.0692	0.0549	0.0582	0.0816
	RPE-r	0.0023	0.0178	0.015	0.0189	0.0011
07	ATE	16.2925	18.15	21.4758	15.208	16.793
	RPE-t	0.0924	0.1182	0.1082	0.1858	0.3801
	RPE-r	0.003	0.041	0.0452	0.0304	0.0743
09	ATE	39.3475	10.1583	15.0372	41.7685	23.619
	RPE-t	0.0907	0.0366	0.0761	0.0917	0.0452
	RPE-r	0.0025	0.0114	0.0208	0.0345	0.0272
10	ATE	5.4202	47.8139	20.2817	12.9822	24.293
	RPE-t	0.0336	0.2241	0.1068	0.2275	0.4173
	RPE-r	0.0028	0.0305	0.0447	0.027	0.0562

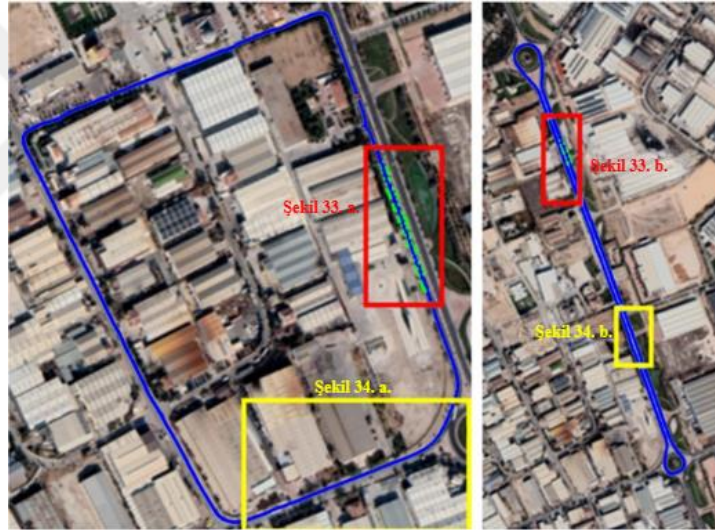
5.2.2. GNSS/GD-VIO/IMU Füzyonu Sonuçları

Bu alt bölümde, gerçek bir otonom araç kullanarak GNSS/GD-VIO/IMU tabanlı konumlandırmanın performansı analiz edilmiştir. Önerilen yöntem, rota üzerinde yaklaşık 4960 metrelik bir mesafeyi kapsayan ve dört ani viraj ile iki yuvarlak dönerde iki U dönüşü içeren iki deneysel kentsel ortamda (sırasıyla 8.467 ve 8.967 dakika) değerlendirilmiştir. Bu tezde, iki UKF tabanlı yaklaşımın konum tahminleri referansla karşılaştırılmaktadır: biri çoklu adım düzeltme filtresi stratejisi içeren UKF yöntemi, 'UKF' olarak adlandırılırken, diğeri çoklu adım düzeltme filtresi olmadan UKF yöntemi, 'UKF w/o' olarak adlandırılır. Bu tezde önerilen çoklu adım düzeltme filtresi ve GNSS/GD-VIO/IMU birleştirme tekniği, GNSS, IMU, Stereo Kamera ve 3D/2D LiDAR ile donatılmış otonom bir golf aracı kullanılarak test edilmiştir.

Yapılan deneyde konumlandırma için birkaç önemli parametreyi içeren UKF kullanılmıştır. İşlem Gürültü Kovaryans Matrisi (Q), sistem dinamiklerindeki belirsizliği hesaba katmak için kullanılır, bu da pozisyon, hız ve yönü içerir. Pozisyon için 0.1 ila 1 metre aralığında bir varyans aralığı belirlenmiştir, hız için ise 0.1 ila 1 metre/saniye aralığı düşünülmüştür. Yönlendirme ile ilgili olarak ise 0.01 ila 0.1 radyan aralığında bir varyans aralığı seçilmiştir. Öte yandan, Ölçüm Gürültü Kovaryans Matrisi (R), sensör

ölçümlerindeki belirsizliği ele alır, özellikle de GNSS pozisyonları. GNSS sistemimizin doğruluk özelliklerine dayanarak bir pozisyon ölçümü varyans aralığı olarak 1 ila 10 metre atanmıştır. Bu parametre değerleri başlangıç tahminleridir ve aracın ve çevresel dinamiklerin özel karakteristiklerine dayanarak daha fazla ayarlanabilirler.

Şekil 5.10, iki yaklaşımın iki metropol alanındaki tahmin edilen yolunu referansla karşılaştırmaktadır. Grafikte, katı kırmızı çizgi ile gösterilen referans yolu; mavi noktalı çizgi 'UKF' tarafından tahmin edilen yolu gösterirken; yeşil noktalı çizgi 'UKF w/o' tarafından hesaplanan yolu gösterir. Grafikten de görülebileceği gibi, 'UKF' yaklaşımından tahmin edilen yol referansı yakından takip etmektedir, ancak 'UKF w/o' yöntemleri, GNSS atlamaları meydana geldiğinde (Şekil 5.11) veya GNSS engellendiğinde (Şekil 5.12) bunu başaramamaktadır.



Şekil 5.10. İki kentsel ortamda 'Ref', 'UKF', 'UKF w/o' yolculukları.

Bu testler sırasında RTK-GNSS verilerinde herhangi bir dalgalanma yoktur. Bazı rastgele gürültüler, Çoklu Adım Düzeltme Filtresi'nin performansını test etmek amacıyla bilinçli olarak oluşturulmuştur. Gürültülü GNSS atlama durumundaki hesaplanan yol, Şekil 5.11. (a) ve 5.11. (b)'de referans ile karşılaştırılmıştır. 'UKF w/o' ve 'UKF' teknikleri sonuçları verir. 'UKF w/o' ve 'UKF' sonuçları sırasıyla yeşil ve mavi noktalı çizgilerle gösterilmiştir. Resimde görüldüğü gibi, atlama nedeniyle oluşan GNSS hatalarına rağmen, 'UKF' tekniği tarafından tahmin edilen yol, referans ile yakından eşleşmektedir; ancak GNSS atlama durumunda, 'UKF w/o' yaklaşımı tarafından tahmin edilen yörünge, referans ile büyük ölçüde farklılık göstermektedir.

Çoklu Adım Düzeltme Filtresi, GNSS atlamalarını yönetmek ve otonom aracın konumlandırma doğruluğunu artırmak için önerilmiştir. Filtre, konum tahmincisini çok yollu ve kısa süreli GNSS kesintilerinden kaynaklanan hatalara karşı dirençli olmasını sağlamak için bir dizi test içerir. Bu testler arasında No-Movement Filtresi, Uydu Görünürlük Filtresi, Yatay/Dikey Seyrelme Filtresi, Pozisyon Değişiklik Doğrulama Filtresi ve bir EKF bulunmaktadır. Her filtre belirli bir amaca hizmet eder ve yanıltıcı ölçümleri filtrelemek ve genel konumlandırma doğruluğunu artırmak için belirli parametreler ve eşikler kullanır.

No-Movement Filtresi, araç hareket etmiyorken GNSS ölçümlerinin güncelleme için kullanılmamasını sağlar, çünkü bu durumda GNSS alıcıları sabit bir konum üretemez. Uydu Görünürlük Filtresi, alıcıya dört veya daha fazla uydu görünüyorsa ölçümleri kabul eder. Yatay/Dikey Seyrelme Filtresi, yatay veya dikey doğruluk seyrelmesi belirli bir eşikten büyükse ölçümleri siler. Pozisyon Değişiklik Doğrulama Filtresi, konum değişikliği belirli bir eşikten büyükse GNSS ölçümlerini reddeder ve EKF, hem GNSS ölçümlerini hem de GD-VIO çıktılarını daha fazla filtrelemek için kullanılır. Bu çok adımlı düzeltme filtresinin sonuçları, genellikle VO yöntemlerinin değerlendirilmesi için kullanılan KITTI veri seti kullanılarak gösterilmiştir. ATE ve RPE olarak adlandırılan metrikler, değerlendirme ölçütleri olarak kullanılmış ve yaygın standartlar ile derin öğrenme temelli tekniklerle karşılaştırılmıştır. Çoklu adımlı düzeltme filtresi, konumlandırma doğruluğunu artırır ve zorlu koşullarda GNSS kesintilerini etkili bir şekilde yönetir. Filtre parametreleri ve eşikleri spesifik uygulamaya ve ortama uyacak şekilde ayarlanabilmektedir.

Çizelge 5.6. Test 1'deki Sayısal ATE ve RPE (pozisyon ve dönme) sonuçları.

	GNSS	GNSS/IMU	GD-VIO/GNSS/IMU
Süre (dakika)		8.967	
ATE (metre)	2.78	2.05	1.196
RPE-t (metre)	0.8	0.6	0.31
RPE-r (derece)	1.2	0.5	0.03

Çizelge 5.7. Test 2'deki Sayısal ATE ve RPE (pozisyon ve dönme) sonuçları.

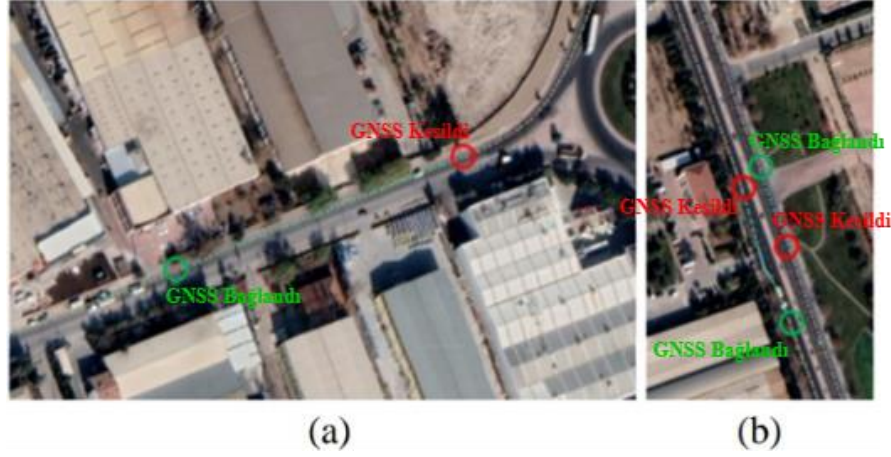
	GNSS	GNSS/IMU	GD-VIO/GNSS/IMU
Süre (dakika)		8.467	
ATE (metre)	3.4	2.35	1.516
RPE-t (metre)	1.01	0.81	0.18
RPE-r (derece)	0.98	0.76	0.05

Çizelgeler 5.6 ve 5.7, süre (dakika), ATE (metre), RPE- pozisyon (metre) ve RPE- dönme (derece) açısından üç yöntemi (GNSS, GNSS/IMU ve Bizimkisi) karşılaştırmaktadır. GNSS için ATE, Çizelge 5.6'da 2.78 metre, RPE- pozisyon 0.8 metre ve RPE-dönme 1.2 derecedir, Çizelge 5.7'de ATE 3.4 metre, RPE- pozisyon 1.01 metre ve RPE-rotasyon 0.98 derecedir. GNSS/IMU için ATE, Çizelge 5.6'da 2.05 metre, RPE- pozisyon 0.6 metre ve RPE-dönme 0.5 derecedir, Çizelge 5.7'de ATE 2.35 metre, RPE- pozisyon 0.81 metre ve RPE-dönme 0.76 derecedir. Çizelge 5.6'da ATE 1.196 metre, RPE- pozisyon 0.31 metre ve RPE-dönme 0.03 derece gösterirken, Çizelge 5.7'de ATE 1.516 metre, RPE-pozisyon 0.18 metre ve RPE-dönme 0.05 derecedir. Bu sonuçlar, önerilen yöntemin doğruluk ve hassasiyet açısından GNSS ve GNSS/IMU'yu aşarak daha düşük ATE ve RPE-pozisyon değerlerine sahip olduğunu göstermektedir. Test 1'de, önerilen yöntemin ayrıca belirli bağlamlarda GNSS ve GNSS/IMU'ya kıyasla önemli ölçüde daha düşük RPE-dönme değerleri sergiledi, bu da üstünlüğünü işaret etmektedir.



Şekil 5.11. Multi-step düzeltme filtresi şemasıyla ve şemasız UKF tabanlı yaklaşımlar kullanılarak, gürültülü RTK-GNSS verilerinde öngörülen yörüngeler ile referans arasındaki karşılaştırma.

GNSS/GD-VIO/IMU füzyon yönteminin GNSS reddedilen senaryolardaki değerlendirmesi, RTK GNSS ölçümlerinin 245 saniye (330 metre) süreyle kesilmesini içermiştir. Şekil 5.12, GNSS reddedilen iki senaryoda GNSS/GD-VIO/IMU'nun sonuçlarını göstermektedir. Kırmızı çizgi referans yolu temsil ederken, yeşil ve mavi kesikli çizgiler sırasıyla 'UKF w/o' ve 'UKF' sonuçlarını göstermektedir. GNSS verilerinin eksikliğine rağmen, önerilen yöntemin GD-VIO/IMU füzyonu aracılığıyla konumlandırmayı sürdürebildiği açıktır. Algoritma, GNSS ölçümleri tekrar kullanılabilir hale geldiğinde konumlandırmayı düzeltir ve devam ettirir.



Şekil 5.12. UKF tabanlı GD-VIO/IMU füzyonu kullanılarak GNSS reddedilen senaryolarda öngörülen yörüngeler.

Bu tez uygulaması, otonom bir araç için GNSS/GDVIO/IMU tabanlı konumlandırma sistemini değerlendirmektedir. Sistem, GNSS, IMU, stereo kamera ve 3D/2D LiDAR ile donatılmış otonom bir golf aracı kullanılarak iki kentsel ortamda değerlendirilmiştir. Konumlandırma sistemi, referansla karşılaştırılarak değerlendirilmiştir ve iki UKF tabanlı yaklaşım kullanılmıştır: 'UKF', çoklu adım düzeltme filtresini kullanan ve 'UKF w/o', kullanmayan. Sonuçlar, 'UKF' yaklaşımının çoklu adım düzeltme filtresi ile referansı doğru bir şekilde takip edebildiğini göstermiştir, hatta GNSS sıçramaları veya blokları durumunda, 'UKF w/o' yaklaşımının bu senaryolarda kötü bir performans sergilediğini göstermiştir. Çoklu adım düzeltme filtresi, No-Hareket Filtresi, Uydu Görünürlük Filtresi, Yatay/Dikey Seyrelme Filtresi, Konum Değişikliği Onay Filtresi ve EKF gibi çeşitli filtreleri içerir ve genel konumlandırma doğruluğunu artırmak için GNSS ölçümlerindeki hataları filtreler. Değerlendirmenin sonuçları, önerilen çoklu adım düzeltme filtresinin, yaygın standartlar ve derin öğrenmeye dayalı tekniklere kıyasla konumlandırma doğruluğunu artırdığını göstermiştir. Ayrıca, GNSS reddedilen senaryolarda GD-VIO/IMU verilerini kullanarak konumlandırmayı sürdürebilen GNSS/GD-VIO/IMU füzyon yöntemi başarıyla uygulanmıştır.

6. SONUÇ VE ÖNERİLER

Bu tez çalışması, İnsansız Hava Araçları (İHA'lar) ve otonom araçlar için sensör füzyon tabanlı konumlandırma sorunlarına yönelik iki ana uygulama üzerinde gerçekleştirilen kapsamlı araştırmaların sonuçlarını sunmaktadır.

İlk uygulamada, İHA'lar için küresel konumlandırma sorunu, sıralı, uçtan uca ve çoklu modele dayalı derin öğrenme tabanlı monoküler görsel-ataletsel konumlandırma yöntemi önerilerek analiz edilmektedir. Daha spesifik olarak, önerilen sinir ağı mimarisi üç katmandan oluşmaktadır: görsel özellik çıkarıcı CNN ağı, küçük IMU entegratörü LSTM ağı, ve konum tahmini için çift yönlü LSTM ağı. Ayrıca, LSTM yerine geleneksel IMU filtreleme yöntemlerini convNet ile birleştirerek, daha zaman verimli bir konum tahmini yöntemi sunulmaktadır. Bu uygulamadaki odak, özellikle kapalı alan senaryolarında görsel-ataletsel tabanlı konumlandırma yaklaşımlarının hassasiyetini ve verimliliğini değerlendirmektir. Önerilen derin küresel konumlandırma, iç mekan İHA veri setleri, simülasyon ortamları ve gerçek dünya İHA deneyleri üzerinde çeşitli öncü algoritmalarla karşılaştırılarak doğruluk ve zaman verimliliği açısından incelenmiştir. Ayrıca, IMU-LSTM ve IMU-Filtre tabanlı konum tahmin algoritmalarının ayrıntılı bir analizi de sunularak karşılaştırma yapılmıştır. Deneysel sonuçlar, önerilen filtre tabanlı yaklaşımın, derin öğrenme yaklaşımı ile birleştirildiğinde iç mekan İHA konumlandırmasında doğruluk ve zaman verimliliği açısından umut vaat ettiğini göstermektedir.

İkinci uygulama, bir otonom aracın GNSS/kamera/IMU sensör füzyon tabanlı konumlandırmasını geliştirmek amacıyla yapılmıştır. Özellikle, GNSS sinyallerinin yansımalar nedeniyle engellenmesi veya müdahale görmesi durumunda, doğru olmayan GNSS verilerini düzeltmek için çok aşamalı bir düzeltme filtresi kullanılmaktadır. Önerilen çözümler, çeşitli sensörlerden gelen yüksek miktarda veriyi entegre ederek sensörlerin bireysel zayıflıklarını telafi etmeyi amaçlamaktadır. Ayrıca, bu çalışma, GNSS kesintileri durumunda aracın konumunu daha iyi belirlemek için GD-VIO'yu önermektedir. Bu araştırmada önerilen algoritmalar, gerçek dünya deneyimleri ile test edilmiş ve doğru ve güvenilir araç duruş tahminleri sağlama yeteneklerini göstermiştir.

Özet olarak, her iki uygulama, gerçek dünya deneyleri ve titiz testlerle değerlendirilmiş, önerilen çözümlerin etkinliği kanıtlanmıştır. Bu çalışmalar, otonom sistemlerin konumlandırma alanındaki mevcut zorlukları aşma ve endüstriye daha güvenilir ve etkin çözümler sunma yolunda önemli adımları temsil etmektedir. Gelecekteki çalışmaların, bu yöntemlerin daha da optimize edilmesi, genişletilmesi ve endüstriyel uygulamalara daha etkin bir şekilde entegre edilmesi yönünde odaklanması gerekmektedir.

6.1. Öneriler

Önerilen tez yöntemlerinin gözetimli öğrenme (supervised learning) tabanlı olduğundan, modelin bir ortam hakkında ne kadar bilgi sahibi olursa, konum tahminini o kadar iyi yapabileceği gösterilmiştir. Ancak bu durum, aşırı öğrenme sorununa ve eğitilen modelin genelleme yeteneğinin eksik olmasına yol açabilmektedir. Ayrıca gözetimli derin öğrenme yöntemlerinin ek dezavantajı, gerçek değer etiketlerine olan ihtiyaç ve verilerin zorluk senaryolarının (örneğin, farklı hava durumu ve çevresel dinamiklik) tümü için eğitilmesi gerekliliğidir. Eğitim verileri bu son koşulu sağlamıyorsa, elde edilen eğitilmiş model genelleme özelliğine sahip olmayacak ve gerçek dünya uygulamalarında kullanılamayacaktır. Bu sorunları ele almak ve sağlam ve güçlü bir derin öğrenme tabanlı durum tahmini veya SLAM sistemleri geliştirmek için bazı yaklaşımlar şunlar olabilir:

Daha fazla sensör verisi ve daha iyi füzyon algoritmaları: Kamera ve IMU sensörlerinin ötesinde, LiDAR, termal kamera, ve radar gibi diğer sensörlerden elde edilen bilgiler, derin öğrenme ve sensör füzyonu tabanlı konumlandırma sistemlerinin özellikle zor hava koşullarında veya düşük ışık koşullarında daha doğru ve sağlam tahminler yapmasına olanak tanıyabilir. Ayrıca, bu sensör verilerinin füzyonu bir araştırma yolu olarak kullanılabilir. Derin öğrenme tabanlı sistemlerde, sensör füzyon algoritmalarına odaklanan pek çok çalışma bulunmamaktadır. Bu nedenle, multimodal sensörlerin kullanımı ve geliştirilmiş füzyon algoritmalarının potansiyeli, daha sağlam sistemler üretme olasılığına sahiptir.

Gözetimsiz/Uyarlamalı Öğrenme: Gözetimsiz/uyarlamalı (unsupervised/adaptive) öğrenme yöntemlerinin kullanılması, büyük miktarda eğitim verisine olan ihtiyacı ortadan kaldırarak derin öğrenme tabanlı yöntemlerin performansını artırabilir. Bu yöntemler, derin öğrenme yöntemlerini, görünmemiş ve dinamik ortamlara poz veya sahneleri tahmin ederek genelleme ve adapte olma yeteneğine sahip kılabilir.

Semantik bilgi: Kamera görüntülerinde bulunan semantik bilgi, derin öğrenme tabanlı yaklaşımları çevre anlayışını ve semantik akıl yürütme yeteneğini artırarak geliştirebilir. Bu bilgi, cihazın robotu çevreleyen nesneleri algılamasına ve bu bilgiye dayanarak konumlandırma kararları vermesine yol açabilir.

Hibrid sistemler: Derin öğrenme tabanlı yöntemlerin avantajlarını ve geleneksel yöntemlerin döngü kapatma ve verimli optimizasyon yeteneklerini birleştirmek, hibrid

bir sistem oluşturabilir. Derin öğrenmenin sağlam özellik tespiti yeteneđi, geleneksel yöntemlerin döngü kapatma (loop closure) ve etkili optimizasyon yetenekleriyle birleřtirildiđinde, konum tahmini ve SLAM sistemlerini geliřtirebilir.



KAYNAKLAR

- Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., ... & Zheng, X. (2016). Tensorflow: Large-scale machine learning on heterogeneous distributed systems. arXiv preprint arXiv:1603.04467.
- Almalioglu, Y., Saputra, M. R. U., De Gusmao, P. P., Markham, A., & Trigoni, N. (2019, May). GANVO: Unsupervised deep monocular visual odometry and depth estimation with generative adversarial networks. In 2019 International conference on robotics and automation (ICRA) (pp. 5474-5480). IEEE.
- Almalioglu, Y., Turan, M., Saputra, M. R. U., de Gusmão, P. P., Markham, A., & Trigoni, N. (2022). SelfVIO: Self-supervised deep monocular Visual-Inertial Odometry and depth estimation. *Neural Networks*, 150, 119-136.
- Aslan, M. F., Durdu, A., Yusefi, A., & Yilmaz, A. (2022). HVIONet: A deep learning based hybrid visual-inertial odometry approach for unmanned aerial system position estimation. *Neural Networks*, 155, 461-474.
- Bacha, A. R. A., Gruyer, D., & Lambert, A. (2016). OKPS: A reactive/cooperative multi-sensors data fusion approach designed for robust vehicle localization. *Positioning*, 7(1), pp-1.
- Baldini, F., Anandkumar, A., & Murray, R. M. (2020, July). Learning pose estimation for UAV autonomous navigation and landing using visual-inertial sensor data. In 2020 American Control Conference (ACC) (pp. 2961-2966). IEEE.
- Bay, H., Tuytelaars, T., & Van Gool, L. (2006). Surf: Speeded up robust features. In *Computer Vision—ECCV 2006: 9th European Conference on Computer Vision, Graz, Austria, May 7-13, 2006. Proceedings, Part I 9* (pp. 404-417). Springer Berlin Heidelberg.
- Bloesch, M., Omari, S., Hutter, M., & Siegwart, R. (2015, September). Robust visual inertial odometry using a direct EKF-based approach. In 2015 IEEE/RSJ international conference on intelligent robots and systems (IROS) (pp. 298-304). IEEE.
- Brahmbhatt, S., Gu, J., Kim, K., Hays, J., & Kautz, J. (2018). Geometry-aware learning of maps for camera localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2616-2625).
- Burri, M., Nikolic, J., Gohl, P., Schneider, T., Rehder, J., Omari, S., ... & Siegwart, R. (2016). The EuRoC micro aerial vehicle datasets. *The International Journal of Robotics Research*, 35(10), 1157-1163.
- Caron, F., Duflos, E., Pomorski, D., & Vanheeghe, P. (2006). GPS/IMU data fusion using multisensor Kalman filtering: introduction of contextual aspects. *Information fusion*, 7(2), 221-230.

- Chang, P., & Hebert, M. (2000, June). Omni-directional structure from motion. In Proceedings IEEE Workshop on Omnidirectional Vision (Cat. No. PR00704) (pp. 127-133). IEEE.
- Chen, C., Lu, X., Markham, A., & Trigoni, N. (2018, April). Ionet: Learning to cure the curse of drift in inertial odometry. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 32, No. 1).
- Chen, C., Rosa, S., Lu, C. X., Wang, B., Trigoni, N., & Markham, A. (2022). Learning selective sensor fusion for state estimation. *IEEE Transactions on Neural Networks and Learning Systems*.
- Chen, C., Rosa, S., Miao, Y., Lu, C. X., Wu, W., Markham, A., & Trigoni, N. (2019). Selective sensor fusion for neural visual-inertial odometry. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 10542-10551).
- Chen, C., Zhu, H., Li, M., & You, S. (2018). A review of visual-inertial simultaneous localization and mapping from filtering-based and optimization-based perspectives. *Robotics*, 7(3), 45.
- Chen, C., Zhu, H., Wang, L., & Liu, Y. (2019). A stereo visual-inertial SLAM approach for indoor mobile robots in unknown environments without occlusions. *IEEE Access*, 7, 185408-185421.
- Clark, R., Wang, S., Markham, A., Trigoni, N., & Wen, H. (2017). Vidloc: A deep spatio-temporal model for 6-dof video-clip relocalization. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 6856-6864).
- Clark, R., Wang, S., Wen, H., Markham, A., & Trigoni, N. (2017, February). Vinet: Visual-inertial odometry as a sequence-to-sequence learning problem. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 31, No. 1).
- Costante, G., Mancini, M., Valigi, P., & Ciarfuglia, T. A. (2015). Exploring representation learning with cnns for frame-to-frame ego-motion estimation. *IEEE robotics and automation letters*, 1(1), 18-25.
- Dai, H. F., Bian, H. W., Wang, R. Y., & Ma, H. (2020). An INS/GNSS integrated navigation in GNSS denied environment using recurrent neural network. *Defence technology*, 16(2), 334-340.
- Dosovitskiy, A., Fischer, P., Ilg, E., Hausser, P., Hazirbas, C., Golkov, V., ... & Brox, T. (2015). FlowNet: Learning optical flow with convolutional networks. In Proceedings of the IEEE international conference on computer vision (pp. 2758-2766).
- Engel, J., Koltun, V., & Cremers, D. (2017). Direct sparse odometry. *IEEE transactions on pattern analysis and machine intelligence*, 40(3), 611-625.

- Engel, J., Schöps, T., & Cremers, D. (2014, September). LSD-SLAM: Large-scale direct monocular SLAM. In European conference on computer vision (pp. 834-849). Cham: Springer International Publishing.
- Engel, J., Sturm, J., & Cremers, D. (2013). Semi-dense visual odometry for a monocular camera. In Proceedings of the IEEE international conference on computer vision (pp. 1449-1456).
- Forster, C., Carlone, L., Dellaert, F., & Scaramuzza, D. (2016). On-manifold preintegration for real-time visual--inertial odometry. *IEEE Transactions on Robotics*, 33(1), 1-21.
- Forster, C., Pizzoli, M., & Scaramuzza, D. (2014, May). SVO: Fast semi-direct monocular visual odometry. In 2014 IEEE international conference on robotics and automation (ICRA) (pp. 15-22). IEEE.
- Geiger, A., Lenz, P., & Urtasun, R. (2012, June). Are we ready for autonomous driving? the kitti vision benchmark suite. In 2012 IEEE conference on computer vision and pattern recognition (pp. 3354-3361). IEEE.
- Geiger, A., Ziegler, J., & Stiller, C. (2011, June). Stereoscan: Dense 3d reconstruction in real-time. In 2011 IEEE intelligent vehicles symposium (IV) (pp. 963-968). Ieee.
- Gers, F. A., Schraudolph, N. N. ve Schmidhuber, J., 2002, Learning precise timing with LSTM recurrent networks, *Journal of machine learning research*, 3 (Aug), 115-143.
- Guo, Y., Kong, H., & Gu, S. (2023). Unsupervised Multi-Spectrum Stereo Depth Estimation for All-Day Vision. *IEEE Transactions on Intelligent Vehicles*.
- Han, L., Lin, Y., Du, G., & Lian, S. (2019, November). Deepvio: Self-supervised deep learning of monocular visual inertial odometry using 3d geometric constraints. In 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (pp. 6906-6913). IEEE.
- Harris, C., & Stephens, M. (1988, August). A combined corner and edge detector. In *Alvey vision conference* (Vol. 15, No. 50, pp. 10-5244).
- Hartikainen, J., Solin, A., & Särkkä, S. (2011). Optimal filtering with Kalman filters and smoothers. Department of biomedical engineering and computational sciences, Aalto University School of Science, 16th August.
- Haykin, S. (1994). *Neural networks: a comprehensive foundation*. Prentice Hall PTR.
- Huang, Z., Xu, Y., Shi, J., Zhou, X., Bao, H., & Zhang, G. (2019). Prior guided dropout for robust visual localization in dynamic environments. In Proceedings of the IEEE/CVF international conference on computer vision (pp. 2791-2800).
- Janocha, K., & Czarnecki, W. M. (2017). On loss functions for deep neural networks in classification. *arXiv preprint arXiv:1702.05659*.

- Jiang, C., Chen, Y., Chen, S., Bo, Y., Li, W., Tian, W., & Guo, J. (2019). A mixed deep recurrent neural network for MEMS gyroscope noise suppressing. *Electronics*, 8(2), 181.
- Jin, H., Favaro, P., & Soatto, S. (2003). A semi-direct approach to structure from motion. *The Visual Computer*, 19, 377-394.
- Kaiser, J., Martinelli, A., Fontana, F., & Scaramuzza, D. (2016). Simultaneous state initialization and gyroscope bias calibration in visual inertial aided navigation. *IEEE Robotics and Automation Letters*, 2(1), 18-25.
- Kang, R., Shi, J., Li, X., Liu, Y., & Liu, X. (2019). DF-SLAM: A deep-learning enhanced visual SLAM system based on deep local features. *arXiv preprint arXiv:1901.07223*.
- Karl Pearson F.R.S. . (1901). LIII. On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11), 559–572. <https://doi.org/10.1080/14786440109462720>.
- Kazik, T., Kneip, L., Nikolic, J., Pollefeys, M., & Siegwart, R. (2012, June). Real-time 6d stereo visual odometry with non-overlapping fields of view. In *2012 IEEE conference on computer vision and pattern recognition* (pp. 1529-1536). IEEE.
- Kim, H. U., & Bae, T. S. (2019). Deep learning-based GNSS network-based real-time kinematic improvement for autonomous ground vehicle navigation. *Journal of Sensors*, 2019.
- Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- Kitt, B., Geiger, A., & Lategahn, H. (2010, June). Visual odometry based on stereo image sequences with RANSAC-based outlier rejection scheme. In *2010 IEEE intelligent vehicles symposium* (pp. 486-492). IEEE.
- Kendall, A., & Cipolla, R. (2017). Geometric loss functions for camera pose regression with deep learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 5974-5983).
- Kendall, A., Grimes, M., & Cipolla, R. (2015). PoseNet: A convolutional network for real-time 6-dof camera relocalization. In *Proceedings of the IEEE international conference on computer vision* (pp. 2938-2946).
- Kneip, L., Chli, M., & Siegwart, R. (2011). Robust real-time visual odometry with a single camera and an IMU. In *Proceedings of the British Machine Vision Conference 2011*. British Machine Vision Association.
- Koch, G., Zemel, R., & Salakhutdinov, R. (2015, July). Siamese neural networks for one-shot image recognition. In *ICML deep learning workshop* (Vol. 2, No. 1).

- Koenig, N., & Howard, A. (2004, September). Design and use paradigms for gazebo, an open-source multi-robot simulator. In 2004 IEEE/RSJ international conference on intelligent robots and systems (IROS)(IEEE Cat. No. 04CH37566) (Vol. 3, pp. 2149-2154). IEEE.
- Koenig, N., & Hsu, J. (2013). The many faces of simulation: Use cases for a general purpose simulator. In Proc. of the ICRA (Vol. 13, pp. 10-11).
- Konda, K. R., & Memisevic, R. (2015). Learning visual odometry with a convolutional network. VISAPP (1), 2015, 486-490.
- Larsson, M., Stenborg, E., Toft, C., Hammarstrand, L., Sattler, T., & Kahl, F. (2019). Fine-grained segmentation networks: Self-supervised segmentation for improved long-term visual localization. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 31-41).
- Leutenegger, S., Lynen, S., Bosse, M., Siegwart, R., & Furgale, P. (2015). Keyframe-based visual-inertial odometry using nonlinear optimization. The International Journal of Robotics Research, 34(3), 314-334.
- Li, C., & Waslander, S. L. (2020, May). Towards end-to-end learning of visual inertial odometry with an EKF. In 2020 17th Conference on Computer and Robot Vision (CRV) (pp. 190-197). IEEE.
- Li, R., Wang, S., Long, Z., & Gu, D. (2018, May). Undeepvo: Monocular visual odometry through unsupervised deep learning. In 2018 IEEE international conference on robotics and automation (ICRA) (pp. 7286-7291). IEEE.
- Li, S., Xue, F., Wang, X., Yan, Z., & Zha, H. (2019). Sequential adversarial learning for self-supervised deep visual odometry. In Proceedings of the IEEE/CVF international conference on computer vision (pp. 2851-2860).
- Li, Y., Ushiku, Y., & Harada, T. (2019, May). Pose graph optimization for unsupervised monocular visual odometry. In 2019 International Conference on Robotics and Automation (ICRA) (pp. 5439-5445). IEEE.
- Liu, J., Cao, Z., Liu, X., Wang, S., & Yu, J. (2022). Self-Supervised Monocular Depth Estimation With Geometric Prior and Pixel-Level Sensitivity. IEEE Transactions on Intelligent Vehicles, 8(3), 2244-2256.
- Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. International journal of computer vision, 60, 91-110.
- Lynen, S., Achtelik, M. W., Weiss, S., Chli, M., & Siegwart, R. (2013, November). A robust and modular multi-sensor fusion approach applied to mav navigation. In 2013 IEEE/RSJ international conference on intelligent robots and systems (pp. 3923-3929). IEEE.
- Madgwick, S. (2010). An efficient orientation filter for inertial and inertial/magnetic sensor arrays. Report x-io and University of Bristol (UK), 25, 113-118.

- Mahony, R., Hamel, T., & Pflimlin, J. M. (2008). Nonlinear complementary filters on the special orthogonal group. *IEEE Transactions on automatic control*, 53(5), 1203-1218.
- Melekhov, I., Ylioinas, J., Kannala, J., & Rahtu, E. (2017). Image-based localization using hourglass networks. In *Proceedings of the IEEE international conference on computer vision workshops* (pp. 879-886).
- Melekhov, I., Ylioinas, J., Kannala, J., & Rahtu, E. (2017). Relative camera pose estimation using convolutional neural networks. In *Advanced Concepts for Intelligent Vision Systems: 18th International Conference, ACIVS 2017, Antwerp, Belgium, September 18-21, 2017, Proceedings 18* (pp. 675-687). Springer International Publishing.
- Meng, X., Wang, H., & Liu, B. (2017). A robust vehicle localization approach based on gnss/imu/dmi/lidar sensor fusion for autonomous vehicles. *Sensors*, 17(9), 2140.
- Mohanty, V., Agrawal, S., Datta, S., Ghosh, A., Sharma, V. D., & Chakravarty, D. (2016). Deepvo: A deep learning approach for monocular visual odometry. *arXiv preprint arXiv:1611.06069*.
- Molton, N., Davison, A. J., & Reid, I. (2004, September). Locally planar patch features for real-time structure from motion. In *Bmvc* (pp. 1-10).
- Mourikis, A. I., & Roumeliotis, S. I. (2007, April). A multi-state constraint Kalman filter for vision-aided inertial navigation. In *Proceedings 2007 IEEE international conference on robotics and automation* (pp. 3565-3572). IEEE.
- Mur-Artal, R., & Tardós, J. D. (2017). Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE transactions on robotics*, 33(5), 1255-1262.
- Naseer, T., & Burgard, W. (2017, September). Deep regression for monocular camera-based 6-dof global localization in outdoor environments. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 1525-1530). IEEE.
- Ndjeng, A. N., Gruyer, D., Glaser, S., & Lambert, A. (2011). Low cost IMU–Odometer–GPS ego localization for unusual maneuvers. *Information Fusion*, 12(4), 264-274.
- Newcombe, R. A., Lovegrove, S. J., & Davison, A. J. (2011, November). DTAM: Dense tracking and mapping in real-time. In *2011 international conference on computer vision* (pp. 2320-2327). IEEE.
- Nicolai, A., Skeelee, R., Eriksen, C., & Hollinger, G. A. (2016, June). Deep learning for laser based odometry estimation. In *RSS workshop Limits and Potentials of Deep Learning in Robotics* (Vol. 184, p. 1).

- Nistér, D. (2004). An efficient solution to the five-point relative pose problem. *IEEE transactions on pattern analysis and machine intelligence*, 26(6), 756-770.
- Noureldin, A., El-Shafie, A., & Bayoumi, M. (2011). GPS/INS integration utilizing dynamic neural networks for vehicular navigation. *Information fusion*, 12(1), 48-57.
- Pretto, A., Menegatti, E., & Pagello, E. (2011, May). Omnidirectional dense large-scale mapping and navigation based on meaningful triangulation. In *2011 IEEE International Conference on Robotics and Automation* (pp. 3289-3296). IEEE.
- Purkait, P., Zhao, C., & Zach, C. (2018). Synthetic View Generation for Absolute Pose Regression and Image Synthesis. In *BMVC* (p. 69).
- Qin, T., Li, P., & Shen, S. (2018). Vins-mono: A robust and versatile monocular visual-inertial state estimator. *IEEE Transactions on Robotics*, 34(4), 1004-1020.
- Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., ... & Ng, A. Y. (2009, May). ROS: an open-source Robot Operating System. In *ICRA workshop on open source software* (Vol. 3, No. 3.2, p. 5).
- Radwan, N., Valada, A., & Burgard, W. (2018). Vlocnet++: Deep multitask learning for semantic visual localization and odometry. *IEEE Robotics and Automation Letters*, 3(4), 4407-4414.
- Rosinol, A., Abate, M., Chang, Y., & Carlone, L. (2020, May). Kimera: an open-source library for real-time metric-semantic localization and mapping. In *2020 IEEE International Conference on Robotics and Automation (ICRA)* (pp. 1689-1696). IEEE.
- Rublee, E., Rabaud, V., Konolige, K., & Bradski, G. (2011, November). ORB: An efficient alternative to SIFT or SURF. In *2011 International conference on computer vision* (pp. 2564-2571). Ieee.
- Saputra, M. R. U., De Gusmao, P. P., Wang, S., Markham, A., & Trigoni, N. (2019, May). Learning monocular visual odometry through geometry-aware curriculum learning. In *2019 international conference on robotics and automation (ICRA)* (pp. 3549-3555). IEEE.
- Scaramuzza, D., Fraundorfer, F., Pollefeys, M., & Siegwart, R. (2008, October). Closing the loop in appearance-guided structure-from-motion for omnidirectional cameras. In *The 8th Workshop on Omnidirectional Vision, Camera Networks and Non-classical Cameras-OMNIVIS*.
- Scaramuzza, D., Martinelli, A., & Siegwart, R. (2006, October). A toolbox for easily calibrating omnidirectional cameras. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 5695-5701). IEEE.

- Scaramuzza, D. (2011). 1-point-ransac structure from motion for vehicle-mounted cameras by exploiting non-holonomic constraints. *International journal of computer vision*, 95, 74-85.
- Scaramuzza, D., & Zhang, Z. (2019). Visual-inertial odometry of aerial robots. *arXiv preprint arXiv:1906.03289*.
- Shamwell, E. J., Lindgren, K., Leung, S., & Nothwang, W. D. (2019). Unsupervised deep visual-inertial odometry with online error correction for RGB-D imagery. *IEEE transactions on pattern analysis and machine intelligence*, 42(10), 2478-2493.
- Shao, S., Pei, Z., Chen, W., Liu, Q., Yue, H., & Li, Z. (2023). Sparse Pseudo-LiDAR Depth Assisted Monocular Depth Estimation. *IEEE Transactions on Intelligent Vehicles*.
- Sheng, L., Xu, D., Ouyang, W., & Wang, X. (2019). Unsupervised collaborative learning of keyframe detection and visual odometry towards monocular deep slam. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (pp. 4302-4311).
- Shi, J. (1994, June). Good features to track. In *1994 Proceedings of IEEE conference on computer vision and pattern recognition* (pp. 593-600). IEEE.
- Silva, H., Bernardino, A., & Silva, E. (2017). A voting method for stereo egomotion estimation. *International Journal of Advanced Robotic Systems*, 14(3), 1729881417710795.
- Silva, H., Bernardino, A., & Silva, E. (2015). Probabilistic egomotion for stereo visual odometry. *Journal of Intelligent & Robotic Systems*, 77, 265-280.
- Silveira, G., Malis, E., & Rives, P. (2008). An efficient direct approach to visual SLAM. *IEEE transactions on robotics*, 24(5), 969-979.
- Strasdat, H., Montiel, J. M., & Davison, A. J. (2012). Visual SLAM: why filter?. *Image and Vision Computing*, 30(2), 65-77.
- Sun, R., Wang, J., Cheng, Q., Mao, Y., & Ochieng, W. Y. (2021). A new IMU-aided multiple GNSS fault detection and exclusion algorithm for integrated navigation in urban environments. *GPS Solutions*, 25, 1-17.
- Sun, R., Yang, Y., Chiang, K. W., Duong, T. T., Lin, K. Y., & Tsai, G. J. (2020). Robust IMU/GPS/VO integration for vehicle navigation in GNSS degraded urban areas. *IEEE Sensors Journal*, 20(17), 10110-10122.
- Tech, Ryze. "Tello." (2020).
- Teuwen, J., & Moriakov. (2020). Chapter 20 - Convolutional neural networks. In *The Elsevier and MICCAI Society Book Series, Handbook of Medical Image Computing and Computer Assisted Intervention*, Academic Press, 2020, Pages 481-501, ISBN 9780128161760.

- Tieleman, T., & Hinton, G. (2012). Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. COURSERA: Neural networks for machine learning, 4(2), 26-31.
- Tomasi, C., & Kanade, T. (1992). Shape and motion from image streams under orthography: a factorization method. *International journal of computer vision*, 9, 137-154.
- Trajković, M., & Hedley, M. (1998). Fast corner detection. *Image and vision computing*, 16(2), 75-87.
- Triggs, B., McLauchlan, P. F., Hartley, R. I., & Fitzgibbon, A. W. (2000). Bundle adjustment—a modern synthesis. In *Vision Algorithms: Theory and Practice: International Workshop on Vision Algorithms Corfu, Greece, September 21–22, 1999 Proceedings* (pp. 298-372). Springer Berlin Heidelberg.
- Valada, A., Radwan, N., & Burgard, W. (2018, May). Deep auxiliary learning for visual localization and odometry. In *2018 IEEE international conference on robotics and automation (ICRA)* (pp. 6939-6946). IEEE.
- Walch, F., Hazirbas, C., Leal-Taixe, L., Sattler, T., Hilsenbeck, S., & Cremers, D. (2017). Image-based localization using lstms for structured feature correlation. In *Proceedings of the IEEE international conference on computer vision* (pp. 627-637).
- Wang, B., Chen, C., Lu, C. X., Zhao, P., Trigoni, N., & Markham, A. (2020, April). Atloc: Attention guided camera localization. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 34, No. 06, pp. 10393-10401).
- Wang, S., Clark, R., Wen, H., & Trigoni, N. (2018). End-to-end, sequence-to-sequence probabilistic visual odometry through deep neural networks. *The International Journal of Robotics Research*, 37(4-5), 513-542.
- Wan, E. A., & Van Der Merwe, R. (2000, October). The unscented Kalman filter for nonlinear estimation. In *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium* (Cat. No. 00EX373) (pp. 153-158). Ieee.
- Wei, P., Hua, G., Huang, W., Meng, F., & Liu, H. (2021, January). Unsupervised monocular visual-inertial odometry network. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence* (pp. 2347-2354).
- Wu, J., Ma, L., & Hu, X. (2017, May). Delving deeper into convolutional neural networks for camera relocation. In *2017 IEEE International Conference on Robotics and Automation (ICRA)* (pp. 5644-5651). IEEE.
- Xue, F., Wang, X., Li, S., Wang, Q., Wang, J., & Zha, H. (2019). Beyond tracking: Selecting memory and refining poses for deep visual odometry. In *Proceedings of*

- the IEEE/CVF conference on computer vision and pattern recognition (pp. 8575-8583).
- Xue, F., Wang, X., Yan, Z., Wang, Q., Wang, J., & Zha, H. (2019). Local supports global: Deep camera relocalization with sequence enhancement. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (pp. 2841-2850).
- Yamashita, R., Nishio, M., Do, R. K. G. ve Togashi, K., 2018, Convolutional neural networks: an overview and application in radiology, *Insights into imaging*, 9 (4), 611-629.
- Yang, N., Wang, R., Stuckler, J., & Cremers, D. (2018). Deep virtual stereo odometry: Leveraging deep depth prediction for monocular direct sparse odometry. In *Proceedings of the European conference on computer vision (ECCV)* (pp. 817-833).5204.
- Yin, Z., & Shi, J. (2018). Geonet: Unsupervised learning of dense depth, optical flow and camera pose. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1983-1992).
- YTU Dataset, in: <https://drive.google.com/drive/folders/1Uk38VwqiD4ntBiw3gB3XTVwtiv2zy70b?usp=sharing> (Ed.), 2021.
- Yusefi, A., Durdu, A., Aslan, M. F., & Sungur, C. (2021). LSTM and filter based comparison analysis for indoor global localization in UAVs. *IEEE Access*, 9, 10054-10069.
- Zhan, H., Garg, R., Weerasekera, C. S., Li, K., Agarwal, H., & Reid, I. (2018). Unsupervised learning of monocular depth estimation and visual odometry with deep feature reconstruction. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 340-349).
- Zhang, Y., Zhang, T., & Huang, S. (2018, May). Comparison of EKF based SLAM and optimization based SLAM algorithms. In *2018 13th IEEE Conference on Industrial Electronics and Applications (ICIEA)* (pp. 1308-1313). IEEE.
- Zhang, Z., & Scaramuzza, D. (2018, October). A tutorial on quantitative trajectory evaluation for visual (-inertial) odometry. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 7244-7251). IEEE.
- Zhao, C., Sun, L., Purkait, P., Duckett, T., & Stolkin, R. (2018, October). Learning monocular visual odometry with dense 3D mapping from dense 3D flow. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 6864-6871). IEEE.
- Zhao, J., Du, J., Zhu, B., Wang, Z., Chen, Z., & Tao, X. (2022). Intelligent vehicle longitudinal cruise control based on adaptive dynamic sliding mode control. *Automotive Engineering*, 44(1), 8-16.

- Zhou, T., Brown, M., Snavely, N., & Lowe, D. G. (2017). Unsupervised learning of depth and ego-motion from video. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1851-1858).
- Zhu, B., Wang, Z., Zhao, J., & Tao, X. (2021, October). Automatic parking space recognition method based on laser radar boundingboxes. In Proceedings of the 2021 Annual Conference of the China Society of Automotive Engineers (Vol. 1).
- Zhuoins, G., Lu, S., Xiong, L., Zhouins, H., Zheng, L., & Zhou, M. (2023). 4DRVO-Net: Deep 4D Radar–Visual Odometry Using Multi-Modal and Multi-Scale Adaptive Fusion. IEEE Transactions on Intelligent Vehicles.

