



T.C.
BANDIRMA ONYEDİ EYLÜL ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

DOKTORA TEZİ

**OTONOM ARAÇLAR İÇİN TRAFİKTEKİ HAREKETLİ NESNELERİN ROTASININ
DERİN ÖĞRENME TABANLI UÇTAN UCA SİSTEMLE TAHMİN EDİLMESİ**

Sevcan TURAN

Akıllı Ulaşım Sistemleri ve Teknolojileri Anabilim Dalı

Akıllı Ulaşım Sistemleri ve Teknolojileri Doktora Programı

MAYIS 2025



T.C.
BANDIRMA ONYEDİ EYLÜL ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

DOKTORA TEZİ

**OTONOM ARAÇLAR İÇİN TRAFİKTEKİ HAREKETLİ NESNELERİN ROTASININ
DERİN ÖĞRENME TABANLI UÇTAN UCA SİSTEMLE TAHMİN EDİLMESİ**

Sevcan TURAN

DANIŞMAN

Prof. Dr. Feyzullah TEMURTAŞ

II. DANIŞMAN

Prof. Dr. Mehmet TEKTAŞ

Akıllı Ulaşım Sistemleri ve Teknolojileri Anabilim Dalı

Akıllı Ulaşım Sistemleri ve Teknolojileri Doktora Programı

MAYIS 2025

ONAY

Sevcan TURAN tarafından hazırlanan “**Otonom Araçlar için Trafikteki Hareketli Nesnelerin Rotasının Derin Öğrenme Tabanlı Uçtan Uca Sistemle Tahmin Edilmesi**” adlı tez çalışması 22/05/2025 tarihinde yapılan sınavla aşağıdaki jüri tarafından oybirliği ile Bandırma Onyedİ Eylül Üniversitesi Lisansüstü Eğitim Enstitüsü Akıllı Ulaşım Sistemleri ve Teknolojileri Anabilim Dalı’nda DOKTORA TEZİ olarak kabul edilmiştir.

Jüri Üyeleri

İmza

Prof. Dr., Feyzullah Temurtaş

(Danışman)

Prof. Dr., Nejat Yumuşak

(Üye)

Prof. Dr., Ahmet Fevzi Baba

(Üye)

Doç. Dr., İlyas Özer

(Üye)

Doç. Dr., Abdullah Elen

(Üye)

Bu tezin kabulü, Lisansüstü Eğitim Enstitüsü Yönetim Kurulu’nun .../.../...gün ve .../... sayılı kararı ile onaylanmıştır.

Prof. Dr. Ahmet AYDIN

Enstitü Müdürü

BANDIRMA ONYEDİ EYLÜL ÜNİVERSİTESİ LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ
DOKTORA TEZ ÇALIŞMASI
ETİK BEYANI

Bandırma Onyedi Eylül Üniversitesi Lisansüstü Eğitim Enstitüsü tez yazım kılavuzuna göre hazırlamış olduğum “**Otonom Araçlar için Trafikteki Hareketli Nesnelerin Rotasının Derin Öğrenme Tabanlı Uçtan Uca Sistemle Tahmin Edilmesi**” adlı tezimin özgün bir çalışma olduğunu, tez hazırlanırken tüm aşamalarda bilimsel etik ilkelerine uygun davrandığımı, tez kapsamında sunulan tüm verileri bilimsel etik ilkelerine uygun elde ettiğimi, tezde faydalandığım tüm eserlere atıf yaptığımı ve kaynaklar kısmında bu eserleri gösterdiğimi beyan ederim.

22/05/2025

Sevcan TURAN
İmza

ÖNSÖZ

Son yıllarda bireysel ve toplu ulaşım alanında otonom araçlar ve onunla ilgili gelişmeler günlük hayatımızın önemli bir konusu haline gelmektedir. Bu durum beraberinde otonom sistemlerin sosyal hayatımıza adaptasyonu problemini ön plana çıkarmaktadır. Özellikle, otonom araçların canlı yaşamına zarar vermeden uyumlu bir şekilde hayatımıza girmesi bu sistemlerin daha hızlı ve kolay özümsemesini sağlayacaktır.

Sağlıktan turizme, trafikten üretim sahalarına kadar birçok alanda hayatımızın parçası olan otonom araçların potansiyel engel ve canlıları tespit ederek kendi rotasını anlık olarak belirlemesi ve buna göre hangi aksiyonu alması gerektiğine karar vermesi gerekmektedir. En doğru kararın verilebilmesi için Derin Öğrenme tabanlı modellerin tasarlanmasına ihtiyaç duyulmaktadır. Dünyada Derin Öğrenme tabanlı algoritmalar ve uygulamalar genelde Akıllı Ulaşım Sistemlerinde özelde otonom araçlar ve teknolojileri alanında yaygın olarak kullanılmaktadır.

Bu anlamda, tez çalışmamızda trafikteki hareketli nesnelerin (yaya, araç, bisikletli) tespit edilmesi, zaman içerisinde izinin takip edilmesi ve rotasının tahmin edilmesi işlemlerini uçtan uca gerçekleştiren yeni bir yöntem ve rota tahmini aşaması için derin öğrenme tabanlı yeni bir model geliştirilmiştir.

Bu tez çalışması sırasında, tez konusunun belirlenmesinden başlayarak son aşamaya kadar her konuda benden yardımlarını esirgemeyen danışman hocam Sayın Prof. Dr. Feyzullah TEMURTAŞ'a ve ikinci danışman hocam Prof. Dr. Mehmet TEKTAŞ'a şükranlarımı sunarım.

Bu günlere gelmemde büyük emekleri olan annem Nuray, babam Ali'ye, yoğun çalışma temposunda benden maddi ve manevi desteğini esirgemeyen eşim Recep Yavuz'a, varlıkları ile birer motivasyon kaynağı olan çocuklarım Sıtkı Enes ve Duru Ülkü'ye ve kardeşlerim Ayşe ve Uğur'a teşekkür ederim.

ÖZET

DOKTORA TEZİ

OTONOM ARAÇLAR İÇİN TRAFİKTEKİ HAREKETLİ NESNELERİN ROTASININ DERİN ÖĞRENME TABANLI UÇTAN UCA SİSTEMLE TAHMİN EDİLMESİ

Sevcan TURAN

Bandırma Onyedli Eylül Üniversitesi

Lisansüstü Eğitim Enstitüsü

Akıllı Ulaşım Sistemleri ve Teknolojileri Anabilim Dalı

Danışman: Prof. Dr. Feyzullah TEMURTAŞ

İkinci Danışman: Prof. Dr. Mehmet TEKTAŞ

Mayıs 2025, 100 sayfa

Otonom araç teknolojileri, trafik güvenliğinin artırılması ve ulaşımın daha verimli hale getirilmesi açısından kritik bir rol üstlenmektedir. Bu tez çalışmasında, otonom araçlar için trafikteki hareketli nesnelerin rotalarını, nesne algılama ve takip süreçlerini de içerecek şekilde uçtan uca tahmin eden Uçtan-Uca Rota adlı yeni bir yöntem ile rota tahmin aşaması için geliştirilen derin öğrenme tabanlı ROTA adlı model önerilmiştir.

Çalışmada, LiDAR sensörlerinden elde edilen veriler kullanılarak trafikteki hareketli nesneler algılanmış; bu nesneler ardışık nokta bulutu çerçeveleri arasında takip edilerek kimlik numaraları atanmış ve izleri oluşturulmuştur. Nesne algılama ve takip aşamalarında, açık kaynak kodlu ve önceden eğitilmiş derin öğrenme modelleri kullanılmıştır. Her bir hareketli nesne için algılama süreci sonucunda elde edilen üç boyutlu çevreleyen kutular aracılığıyla konum, hız, ivmelenme ve yönelim bilgileri çıkarılmış; bu bilgiler doğrultusunda zaman içindeki izleri ve kimlikleri belirlenmiştir.

Rota tahmini aşaması için heterojen çizge yapısını zamansal-uzamsal derin öğrenme modelleri ile birleştiren ROTA modeli geliştirilmiştir. Bu model, trafikteki nesneler arasındaki

etkileşimleri ve hareket dinamiklerini öğrenerek, nesnelerin gelecekte izleyebilecekleri olası rotaları yüksek doğrulukla tahmin etmeyi hedeflemektedir.

Önerilen yöntem ve model, araçlar, yayalar ve bisikletliler gibi farklı türdeki hareketli nesnelerin rotalarını tahmin etmede başarılı sonuçlar ortaya koymuştur. Gelecek 8 saniyeye yönelik rota tahminlerinde, ortalama 1,64 m Ortalama Yer Değiştirme Hatası (ADE) ve 3,91 m Son Yer Değiştirme Hatası (FDE) değerleri elde edilmiş; bu sonuçlar mevcut yöntemlere kıyasla sırasıyla %12,5 ve %18 oranında iyileşme sağlamıştır. Elde edilen bulgular, geliştirilen yöntemin ve modelin doğruluk ve etkinlik açısından güçlü bir alternatif sunduğunu ortaya koymaktadır.

Anahtar kelimeler: Derin Öğrenme, Heterojen Çizgeler, LiDAR Sensörü, Otonom Araçlar, Rota Tahmini.



ABSTRACT

Ph.D. THESIS

TRAJECTORY PREDICTION OF MOVING OBJECTS IN TRAFFIC FOR AUTONOMOUS VEHICLES USING A DEEP LEARNING-BASED END-TO-END SYSTEM

Sevcan TURAN

Bandırma Onyedi Eylül University

School of Graduate Studies

Department of Intelligent Transportation Systems and Technologies

Supervisor: Prof. Dr. Feyzullah TEMURTAŞ

Co-Supervisor: Prof. Dr. Mehmet TEKTAŞ

May 2025, 100 pages

Autonomous vehicle technologies play a critical role in enhancing traffic safety and improving transportation efficiency. In this thesis, a new method called End-to-End Route, which predicts the trajectories of moving objects in traffic in an end-to-end manner—including object detection and tracking processes—is proposed. Additionally, a deep learning-based model named ROUTE, developed for trajectory prediction, is introduced.

In the study, moving objects in traffic were detected using data obtained from LiDAR sensors. These objects were then tracked across consecutive point cloud frames, with unique IDs assigned and motion trajectories constructed. For object detection and tracking stages, open-source and pre-trained deep learning models were utilized. Through the 3D bounding boxes obtained during detection, the position, velocity, acceleration, and orientation of each moving object were extracted. These features were then used to determine the temporal tracks and identities of the objects.

For the trajectory prediction stage, the ROTA model was developed by integrating a heterogeneous graph structure with spatio-temporal deep learning models. This model aims to accurately predict the possible future trajectories of traffic participants by learning their interactions and motion dynamics.

The proposed method and model have demonstrated successful results in predicting the trajectories of various types of moving objects, including vehicles, pedestrians, and cyclists. For 8-second future trajectory predictions, the method achieved an average 1.64 m Average Displacement Error (ADE) and 3.91 m Final Displacement Error (FDE), corresponding to improvements of 12.5% and 18% respectively compared to existing methods. These findings indicate that the proposed method and model offer a strong alternative in terms of accuracy and effectiveness.

Keywords: Autonomous Vehicles, Deep Learning, Heterogeneous Graphs, LiDAR Sensor, Trajectory Prediction.

İÇİNDEKİLER

	Sayfa
ÖNSÖZ	5
ÖZET	I
ABSTRACT	III
İÇİNDEKİLER	V
ŞEKİL LİSTESİ	VII
TABLO LİSTESİ.....	VIII
SİMGE VE KISALTMA LİSTESİ	X
1. GİRİŞ	1
2. OTONOM ARAÇLARDA ROTA TAHMİNİ İÇİN İHTİYAÇ DUYULAN DONANIM ve YAZILIM ALTYAPILARI	19
2.1 OTONOM ARAÇLARDA ÇEVRE ALGILAMA İÇİN KULLANILAN SENSÖRLER	19
2.1.1 LiDAR Sensörü	21
2.2 LiDAR VERİSİNİN İŞLENMESİ	23
2.3 NESNE ALGILAMA ve DEĞERLENDİRME KRİTERLERİ	25
2.4 NESNE TAKİBİ ve DEĞERLENDİRME KRİTERLERİ	29
2.5 ROTA TAHMİNİ ve DEĞERLENDİRME KRİTERLERİ.....	30
2.6 ZAMAN SERİSİ VERİLERİNİN DERİN ÖĞRENME YÖNTEMLERİYLE İŞLENMESİ	31
2.6.1 Uzun Kısa Süreli Hafıza (Long Short Term Memory -LSTM).....	32
2.6.2 Geçitli Tekrarlayan Birim (Gated Recurrent Unit- GRU).....	33
2.6.3 Zamansal Evrişimli Ağ (Temporal Convolutional Network -TCN-Conv1D)....	34
2.6.4 Dönüştürücü Ağlar (Transformer Network)	34
2.7 ÇİZGE İLE VERİ TEMSİLİ	35

2.8	DERİN GRAFİK ÖĞRENİMİ YÖNTEMLERİYLE ÇİZGELER İÇİN DÜĞÜM GÖMMELERİNİN HESAPLANMASI.....	37
2.8.1	Çizge Evrişimli Ağ (Graph Convolutional Network -GCN).....	38
2.8.2	Çizge Dikkat Ağı (Graph Attention Network -GAT).....	38
2.9	ORTALAMA KARE HATASI KAYIP FONKSİYONU.....	39
2.10	ADAM (ADAPTIVE MOMENT ESTIMATION) OPTİMİZASYON YÖNTEMİ 40	
3.	TRAFİKTEKİ HAREKETLİ NESNELER İÇİN UÇTAN-UCA ROTA TAHMİNİ.....	42
3.1	VERİ SETLERİ.....	43
3.2	LİDAR VERİSİNİN İŞLENMESİ ve NESNE ALGILAMA: CenterPoint.....	45
3.3	NESNE TAKİP: MCTrack.....	46
3.4	HETEROJEN ÇİZGE OLUŞTURMA.....	47
3.5	ROTA TAHMİN MODELİ	49
3.5.1	Zamansal Kodlama.....	50
3.5.2	Heterojen Çizge Özellik Çıkarımı.....	52
3.5.3	Önem Atfetme	54
3.5.4	Tahmin Başlığı	55
4.	BULGULAR.....	57
4.1	ÇALIŞMA ORTAMI KONFİGÜRASYONU, PARAMETRELER ve KISITLAR 57	
4.2	KARŞILAŞTIRMA İÇİN KULLANILAN ÇALIŞMALAR.....	59
4.3	ROTA MODELİNİN SONUÇLARI.....	60
4.4	UÇTAN-UCA ROTA YÖNTEMİNİN SONUÇLARI.....	66
5.	TARTIŞMA VE SONUÇ	69
6.	KAYNAKLAR	71
7.	EKLER	83
	ÖZGEÇMİŞ	84

ŞEKİL LİSTESİ

<u>Sekil</u>	<u>Sayfa</u>
Şekil 2-1: Otonom araçta bulunması gereken sistemler [98].....	20
Şekil 2-2: LiDAR sensörünün iç yapısı [101]	21
Şekil 2-3: LiDAR sensörünün çalışma prensibi [100]	22
Şekil 2-4: LiDAR ile elde edilen nokta bulutunun görselleştirilmesi[103]	23
Şekil 2-5: LiDAR ve radar ile aynı ortam için elde edilen görüntüler [104]	23
Şekil 2-6: 2B ve 3B çevreleyen kutu	25
Şekil 2-7: Homojen çizge ve çizgenin komşuluk matrisi ile temsili.....	36
Şekil 2-8: Heterojen çizge ve çizgenin komşuluk matrisi ile temsili	37
Şekil 3-1: Tez çalışmasında geliştirilen Uçtan-Uca Rota yönteminin akış diyagramı.....	42
Şekil 3-2: CenterPoint algoritmasının yapısı [33]	45
Şekil 3-3: MCTrack algoritmasının yapısı [61]	46
Şekil 3-4: Rota tahmin modelinin geliştirilmesi ve eğitimi aşamasında kullanılan heterojen çizgelerin oluşturulması için veri ön işleme akış diyagramı	49
Şekil 3-5: Rota tahmini için geliştirilen dinamik heterojen çizge tabanlı zamansal-uzamsal modelin yapısı	50
Şekil 3-6: Zamansal kodlamanın model yapısı. a) Conv1D, b) LSTM ve GRU	50
Şekil 3-7: Derin çizge öğrenimi ile heterojen çizge düğüm gömmelerinin oluşturulması akış diyagramı.....	53
Şekil 3-8: Önem atfetme aşamasının mimarisi	55
Şekil 4-1: Düğüm türü sayısına göre ROTA modelinin cevap üretme süresi. a) Homojen çizge b) İki tür düğüm olan heterojen çizge c) Üç tür düğüm olan heterojen çizge	65

TABLO LİSTESİ

<u>Tablo</u>	<u>Sayfa</u>
Tablo 2-1: Karmaşıklık matrisi. Tablodaki değerler: 1. Doğru Pozitif: Gerçekte pozitif olanı modelin pozitif olarak işaretlemesi; 2. Yanlış Pozitif: Gerçekte negatif olanı modelin pozitif olarak işaretlemesi; 3. Yanlış Negatif: Gerçekte pozitif olanı modelin negatif olarak işaretlemesi; 4. Doğru Negatif: Gerçekte negatif olanı sistemin de negatif işaretlemesini ifade eder.	26
Tablo 3-1: Otonom araçlar için nesne algılama, takip ve rota tahmini alanında yapılacak çalışmalarda kullanılabilecek veri setleri ve özellikleri	44
Tablo 3-2: Zamansal kodlama modellerinin parametre değerleri. Parametre isimleri PyTorch kütüphanesinde kullanılan isimler ile ifade edilmiştir.	51
Tablo 3-3: Çizge Evrişimli Ağ ve Çizge Dikkat Ağlarında kullanılan parametrelerin değerleri. Parametre isimleri DGL kütüphanesinde kullanılan isimler ile ifade edilmiştir.	54
Tablo 3-4: Önem atfetme aşamasında kullanılan modellerin parametre değerleri. Parametre isimleri PyTorch kütüphanesinde kullanılan isimler ile ifade edilmiştir.	55
Tablo 3-5: Tahmin katmanında kullanılan modelin parametre değerleri. Parametre isimleri PyTorch kütüphanesinde kullanılan isimler ile ifade edilmiştir.	56
Tablo 4-1: ROTA modelinin tek katman Conv1D ve Deney Seti 1 kullanıldığında 1 sn gözlemle elde ettiği sonuçlar	60
Tablo 4-2: Conv1D iki katman halinde kullanıldığında ROTA modeliyle Deney Seti1 üzerinde elde edilen sonuçlar. En iyi değerler kalın ve italik olarak ifade edilmiştir.	61
Tablo 4-3: ROTA modelinin farklı zaman serisi işleme yöntemlerine göre Deney Seti 1 üzerinde gelecekteki 5 ve 8 sn için rota tahmin sonuçları. En iyi değerler kalın ve italik olarak ifade edilmiştir.	61
Tablo 4-4: ROTA modelinin 2 sn gözlem ile gelecekteki 5 sn için Deney Seti 1 üzerindeki tahmin sonuçları. En iyi değerler kalın ve italik olarak ifade edilmiştir.	62
Tablo 4-5: ROTA modelinin Deney Seti 2 üzerinde 1 sn gözlem ile gelecekteki 5 ve 8 sn için rota tahmin sonuçları. En iyi değerler kalın ve italik olarak ifade edilmiştir.	63
Tablo 4-6: ROTA modelinin Argoverse veri seti (Deney veri seti 3) üzerinde 1 sn gözlem ile gelecekteki 5 sn ve 2 sn gözlem ile gelecekteki 3 sn için rota tahmin sonuçları. En iyi değerler kalın ve italik olarak ifade edilmiştir.	64

Tablo 4-7: D�ğ�m �eşidi sayısına g�re ROTA modelinin farklı zaman serisi iřleme y�ntemlerine g�re Deney Seti 2 �zerinde ortalama cevap �retme s�releri.....	65
Tablo 4-8: U�tan-Uca Rota y�nteminin Deney seti 2 �zerinde gelecekteki 5 ve 8 sn i�in tahmin sonu�ları. En iyi deėerler kalın ve italik olarak ifade edilmiřtir.....	67



SİMGE VE KISALTMA LİSTESİ

Simgeler

Açıklama

σ	: Sigmoid
$\emptyset$	: Yükseklik açısı
θ	: Azimut açısı
∇	: Kısmi türev
E	: Kenar
G	: Çizge
km/s	: Kilometre/saat
m	: Metre
m/sn	: Metre/saniye
m/sn ²	: Metre/saniye ²
nm	: Nanometre
sn	: Saniye
V	: Düğüm

Kısaltmalar

Açıklama

3B	: Üç Boyutlu
2B	: İki Boyutlu
ADAM	: Adaptive Moment Estimation
ADE	: Ortalama Yer Değiştirme Hatası (Average Displacement Error)
AMOTA	: Ortalama Çoklu Nesne Takip Doğruluğu
AP	: Ortalama Hassasiyet (Average Precision)
CNN	: Evrişimli Yapay Sinir Ağları (Convolutional Neural Network)
DETR	: Detection Transformer
DGL	: Deep Graph Library
FDE	: Son Yer Değiştirme Hatası (Final Displacement Error)
GAT	: Çizge Dikkat Ağı (Graph Attention Network)
GCN	: Çizge Evrişimli Ağ (Graph Convolutional Network)
GPU	: Grafik Tabanlı İşlemci Ünitesi

GRU	: Geçitli Tekrarlayan Birim (Gated Recurrent Unit)
ID	: Kimlik Numarası
IoU	: Çevreleyen Kutunun Gerçek Kutu ile Örtüşme Oranı
LiDAR	: Işık Tespiti ve Menzil Belirleme (Light Detection and Ranging)
LSTM	: Uzun Kısa Süreli Hafıza (Long Short-Term Memory)
mAP	: Ortalama Ortalama Hassasiyet (Mean Average Precision)
minADE	: Minimum Ortalama Yer Değiştirme Hatası (Minimum Average Displacement Error)
minFDE	: Minimum Son Yer Değiştirme Hatası (Minimum Final Displacement Error)
MOTA	: Çoklu Nesne Takip Doğruluğu
MOTP	: Çoklu Nesne Takip Hassasiyeti
MSE	: Ortalama Kare Hatası (Mean Squared Error)
R-CNN	: Bölgesel Evrişimli Ağ (Region-Based Convolution Neural Network)
ReLU	: Rectified Linear Unit
R-GCN	: İlişkisel Çizge Evrişimli Ağ (Relational Graph Convolutional Network)
RMSE	: Kök Ortalama Kare Hatası (Root Mean Square Error)
RNN	: Tekrarlayan Yapay Sinir Ağları (Recurrent Neural Network)
SSD	: Single Shot MultiBox Detector
TCN	: Zamansal Evrişimli Ağ (Temporal Convolutional Networks)
VGG16	: Visual Geometry Group 16
WPD	: Waymo Perception Dataset
YOLO	: You Look Only Once

1. GİRİŞ

Günümüzde hızla gelişen teknoloji, günlük hayatı kolaylaştırmak için önemli kazanımlar sunmaktadır. Bunlardan biri de modern akıllı ulaşım sistemlerinde devrim yaratma potansiyeline sahip olan ve hızla gelişen otonom araç teknolojisi. Otonom araç, herhangi bir kullanıcı katkısı olmadan donanım ve yazılımlar aracılığıyla etrafını algılayıp hareket edebilen araçtır. Kapalı alanlarda otonom hareket eden robot metrolar; insansız deniz, hava ve kara araçları; trafikte kullanılacak otonom araçlar gibi birçok alanda tasarımlar yapılmakta ve gelişmeler devam etmektedir.

Trafikte kullanılan otonom araçlar son zamanlarda popülerlik kazanmaya başlamıştır. Deloitte tarafından 20 ülkede 25.000 denek üzerinde yapılan ve Türkiye’den 2.988 kişinin katıldığı araştırmaya göre ülkemizde de otonom araçlara ilgi yüksektir [1]. Bu alana yönelik artan ilgi, otonom araçların geliştirilmesi ve tasarımı konusunda birçok şirketin ve araştırma grubunun dikkatini çekmektedir [2].

Araçlar otonomluk düzeyine göre 0 ile 5 arasında sınıflandırılmakta ve 0. seviyede otonom özellik bulunmazken 5. seviye sürücünün hiçbir müdahalesi olmadan aracın hareket ettiği tam otonom seviye olarak ele alınmaktadır. Sürüş kontrolünün tamamen araca bırakıldığı tam otonom araçların gelişimi ile dezavantajlı grupların seyahatlerinin kolaylaşması, seyahat süresinin farklı aktivitelerle değerlendirilebilmesi, insan kaynaklı trafik kazalarının önemli ölçüde azalması, yakıt tasarrufu yapılması, çevre kirliliğinin ve araçlardan oluşabilecek atıkların azalması gibi katkıların elde edilmesi beklenmektedir. Ancak trafik, sosyal etkileşimin yoğun olarak yaşandığı bir ortam olduğu için otonom araç teknolojisinin sosyal hayatta yaygın şekilde benimsenebilmesi, sadece teknik başarı ile değil aynı zamanda insanların güven duygusunu pekiştirecek şekilde sorunsuz ve tutarlı davranışlar sergilemesine bağlıdır [3]. Günümüzde trafiğe çıkabilen otonom araçların karıştığı kazalar bu araçlara karşı önyargıların oluşmasına sebep olmuştur. Bu nedenle tam otonom araçların trafiğe uyumlu ve çevresindeki insanları tedirgin etmeyecek şekilde hareket etmesi gerekmektedir. Aksi halde insanların otonom araçlara güven sorunları aşılamadığı gibi derinleşebilecektir [4].

Tam otonom araçların trafiğe çıkabilmesi ve çevresine zarar vermeden hareket edebilmesi için aşılması gereken bazı problemler bulunmaktadır. Örneğin; diğer sürücülerin davranışlarının algılanması [5], yolda yapısal bir problem olduğunda ne yapılacağına anlık karar verilmesi [6], trafik ışık [7] ve levhalarının [8] tespit edilmesi ve yorumlanması, otonom aracın konumunun [9] ve izleyeceği rotanın belirlenmesi [10], aracın siber saldırılara karşı güvenliğinin sağlanması [11],

trafikte karşılaşılabilecek araç, yaya, hayvan vs. gibi nesnelerin cinsinin tespit edilmesi [12] ve hareket rotasının tahmin edilmesi [13] gibi alanlarda aşılması gereken problemler üzerinde çalışmalar devam etmektedir [14].

Çözüm bekleyen alanlardan biri olan trafikteki hareketli nesnelerin gelecekte takip edebileceği rotanın tahmin edilmesi, yola çıkma veya kazaya sebep olma potansiyeline sahip nesnelere göre otonom aracın hızlanma, yavaşlama, durma, şerit değiştirme vb. aksiyonlara karar verebilmesi için çok kritik bir aşamadır. Kararın isabetli olabilmesi için rota tahmininin en az hatayla gerçekleştirilmesi gerekmektedir.

Trafikteki hareketli nesnelerin gelecekteki rotasını tahmin etme işlemi temel dört aşamadan oluşmaktadır:

Birinci aşama, aracın içinde bulunduğu trafik ortamıyla ilgili bilgilerin radar, LiDAR, ultrasonik, kamera gibi sensörler vasıtasıyla toplanmasıdır [2, 15]. Çalışmalarda genellikle LiDAR sensörü kullanılmaktadır. LiDAR sensörü, sensörden gönderilen ışığın nesnelere çarpıp yansmasıyla veri üretmektedir. Sensör verisinde, nesnenin sensörden uzaklığı, ışığın eğim ve azimut açısı bilgileri bulunmaktadır. Bu bilgiler kullanılarak nesneler üç boyutlu (3B) ortamda nokta bulutu şeklinde seyrek bir bilgi olarak ifade edilmektedir. Daha sonra bu bilgiler işleme kolaylığı sağlamak adına mesafe görünümü, voksel çevirimi yöntemi kullanılarak düzenli bir ızgara yapısı veya kuş bakışı görüntüsü formatında kullanılabilmektedir [16].

Tahmin işleminin ikinci aşaması çevredeki hareketli nesnelerin LiDAR sensör verilerinden algılanmasıdır. Nesne algılama, bir görüntüde veya 3B nokta bulutunda bulunan nesnelerin çevreleyen kutusunun ve sınıfının tespit edilmesi işlemidir. 3B nesne algılamada çevreleyen kutular nesnenin merkez noktasının 3B kartezyen koordinat sistemindeki koordinatı ve kutunun uzunluğu, genişliği, yüksekliği, doğrultu açısı ile ifade edilmektedir. Nesne algılamada genel olarak Evrişimli Sinir Ağları (Convolutional Neural Network- CNN) [17] tabanlı algoritmalar kullanılmaktadır. CNN'e girdi olarak verilen bilgiler üzerinde filtreler ile evrişim işlemi uygulanarak ağın ilk katmanlarında nesnelere ait temel bilgiler elde edilirken derin katmanlarda anlamsal bilgileri içeren özellik haritaları elde edilmektedir. Bu özellik haritaları kullanılarak nesnenin sınıfı ve çevreleyen kutusu bulunmaktadır. Trafik ortamında otonom araçlar için nesnenin sadece sınıfı ve çevreleyen kutusu değil aynı zamanda araca ne kadar mesafede olduğunun da belirlenmesi önemlidir. Bu nedenle genellikle LiDAR sensöründen elde edilen verileri kullanan 3B algılama süreçleri işletilmektedir.

Rota tahmininin üçüncü aşaması nesne takip aşamasıdır. Nesne takip, belirli bir zaman aralığında birbirini takip eden video veya 3B nokta bulutu çerçeveleri arasında aynı hareketli

nesnenin takip ettiđi yörünge ile yönünün tespit edilmesi ve aynı özel numara ile numaralandırılıp eşleştirilmesi işlemidir. Bu aşamada takip esnasında ortaya yeni çıkan nesnelerin takip edilecek nesneler listesine eklenmesi, izi kaybolan nesnelerin ise takip listesinden çıkarılması süreçlerinin de yüksek performansla gerçekleştirilmesi gerekmektedir. Otonom araçlar için 3B nesne algılama işlemleri sonucunda elde edilen çevreleyen kutular kullanılarak nesne takip işlemleri yapılmaktadır.

Ön hazırlık aşamaları tamamlanıp nesnelerin geçmiş hareket bilgileri elde edildikten sonra bu tez kapsamında geliştirilen derin öğrenme modelinin de görevi olan rota tahmin işlemine geçilmektedir. Rota tahmini, belirlenen gözlem süresi boyunca bir nesnenin algılama ve takip süreçleriyle elde edilen geçmişteki hareket bilgilerinin ışığında belirlenen süre boyunca gelecekte takip edebileceđi rotanın kestirimi işlemidir. Bu alandaki ilk çalışmalarda, genellikle fiziksel hareketin ve çevresel etkilerin mühendislik çalışmalarıyla modellenmesi ile kısa süreli tahminler yapılmıştır [18-20]. Uzun süreli rota tahmini derin öğrenmenin kullanılmasıyla beraber ilk defa 2016 yılında Alahi vd. [21] tarafından geliştirilen Social-LSTM ile gerçekleştirilmiştir. Çalışmada yayaların hareket karakteristiđi ve birbiriyle olan etkileşimi Uzun Kısa Süreli Hafıza (Long Short Term Memory- LSTM) [22] yardımıyla öğrenilmiştir. Social-LSTM'nin sunduđu bakış açısı, uzun süreli rota tahminlerinde derin öğrenme tabanlı çalışmalara hız kazandırmıştır.

Rota tahmini konusunda yapılmış çalışmalar incelendiğinde, uçtan uca gerçekleştirilen yaklaşımlarda nesnelerin gelecekteki rotasının tahminindeki hata miktarının tüm aşamalardaki hatalardan etkilenip katlanarak arttığı görülmektedir [23-25]. Trafik, dinamik bir ortam olduğundan, tam otonom araçların işlev gösterebilmesi için yukarıda örneklendirilen tüm aşamaları hızlı ve az hata payıyla gerçekleştirmesi gerekmektedir. Aksi taktirde, bir sistemdeki hata payı diđer sistemleri etkileyerek insanlarla hayvanların canına mal olunmasına ve/veya çevredeki tüm nesnelere zarar verebilecek kötü sonuçların ortaya çıkmasına sebep olabilecektir.

Rota tahmin işlemi aşamalarında karşılaşılabilecek bazı sorunlar aşağıdaki gibidir:

Nesne algılama aşamasında: Yoğun veya karmaşık trafik ortamlarında nesnelerin ayırt edilememesi, hangi LiDAR verisinin hangi nesneden yansıdığının ayrıştırılamaması, çevresel nesnelerin doğru bir şekilde tespit edilememesi, küçük veya bir kısmı başka nesne tarafından örtülmüş nesnelerin algılanamaması, çevrede olmadığı halde sistemin varmış gibi algıladığı nesneler, kötü hava koşullarında sensörlerin elde ettiđi verilerin hatalı olması, ışığın yansıma hataları, yüksek doğruluđu sağlarken hızlı sonuç üretememek gibi sorunlarla karşılaşılmaktadır [26].

Nesne takip aşamasında: Yanlış kimlik atamaları, algılanan nesnelerin görüş alanından çıktığında veya geçici olarak algılanamadıklarında izlerinin takip edilememesi, yoğun ortamlarda nesnelerin hareket yönünün ve hızının sıklıkla değişmesi ve bu nedenle nesnelerin hızının ve yörüngesinin doğru tahmin edilememesi, farklı hareket karakteristiğine sahip nesnelerin eş zamanlı ve doğru bir şekilde takip edilmesi zorlukları gibi problemlerle karşılaşmaktadır [27].

Bu problemlere ek olarak nesneler doğru algılanıp takipleri doğru yapılmış olsa bile rota tahmin işleminin de kendi açısından problemleri bulunmaktadır [28]. Tüm nesnelerin gelecekteki hareketleri, yollardaki engeller ve/veya kaldırımlardaki nesneler vb. statik çevresel faktörlerden etkilenmektedir. Bu nedenle geçmişte bulunulan ortamlarda olmayıp gelecekte her an karşılaşılabilecek yeni bir etken rotanın değişmesine sebep olabilmektedir. Bununla birlikte, bir nesnenin hareketi sadece statik çevresel etkenlerden değil ortamda hareket eden diğer dinamik nesnelerin hareketlerinden de etkilenmektedir. Gelecekte ortaya çıkabilecek veya hareket karakteristiğinde ani bir değişiklik yapan bir nesne, ortamdaki diğer nesnelerin rotasının değişmesine sebep olabilmektedir. Ayrıca, sistemlerin aynı anda ortamda bulunan tüm hareketli nesnelerin her biri için eş zamanlı olarak rota tahmini yapması gerekmektedir, bu da işlem maliyetini arttırmakta ve yüksek işlem gücüne ihtiyaç duyulmasına neden olmaktadır. Başka bir problemde, doğruluğu arttırmak adına birden fazla aynı veya birbirinden farklı yapıdaki sensörler ile elde edilen verilerin beraber kullanılmasında ortaya çıkmaktadır. Çok sayıda sensör ile elde edilen veriler beraber işlenmek istendiğinde sensörlerin senkronize edilmesinde hataların oluşması, sensörlerin kendi hata paylarının veri birleştirmede uyumsuzluklara sebep olması ve bu verilerin birleştirilmesi esnasında yüksek işlem gücüne ihtiyaç duyulması problemleri ortaya çıkmaktadır. Kullanılan derin öğrenme modelleri ve girdi verileri karmaşıktıkça tahmin işleminin cevap üretme süresi uzamaktadır bu durumda gerçek zamanlı çalışacak sistemler açısından problem oluşturmaktadır. Bu nedenle az miktarda veri ve az işlem maliyeti ile kısa sürede cevap üretecek, araçlardaki donanım tarafından desteklenecek modellerin geliştirilmesine ihtiyaç vardır.

Bu problemlerden yola çıkarak, bu tez çalışması kapsamında;

- Otonom araçlar için LiDAR sensörü verisini kullanıp trafikteki hareketli nesnelerin algılanması, takip edilmesi ve gelecekteki rotasının tahmin edilmesi süreçlerini kapsayan ve uçtan-uca sonuç üreten Uçtan-Uca Rota adında yeni bir yöntem geliştirilmiştir. Bu yöntemde algılama ve takip aşaması açık kullanım hakkı sunan önceden eğitilmiş derin öğrenme tabanlı model ile gerçekleştirilmiştir.

- Rota tahmin aşaması için derin öğrenme tabanlı ve dinamik değişen heterojen çizgeler sayesinde zamansal-mekânsal ilişkileri dikkate alabilen ROTA adında yeni bir yapay zekâ modeli geliştirilmiştir.
- Mesaj Yayılımı tabanlı geliştirilen yeni model Heterojen Çizge Dikkat Ağı ve Heterojen Çizge Evrişimli Ağı beraber kullanılmaktadır.
- Model sayesinde farklı türlere ait olsalar dahi o anda ortamda bulunan tüm nesneler için eş zamanlı olarak tek bir rota dizisi daha az hata oranlarıyla tahmin edilebilmektedir.
- Geliştirilen model, 9 ayrı kenar türü ve 3 ayrı düğüm türüne sahip heterojen çizgeleri işleyip nesneler arasındaki farklı türdeki etkileşimlerin karakteristiğini öğrenebilmektedir.
- Geliştirilen ROTA modeli farklı veri setleri üzerinde çalışabilmekte, veri setinden bağımsız sonuç üretebilmektedir. Bu tez çalışması kapsamında birbirinden farklı veri yapısına sahip iki farklı veri seti üzerinde rota tahmini gerçekleştirilmiştir.
- Çalışmalarda farklı zaman dilimleri kullanılarak gözlem ve tahmin süreçleri işletilmektedir. Çünkü rota tahmininde farklı trafik akış hızına bağlı olarak farklı zaman aralıkları için tahminlere ihtiyaç duyulmaktadır. Örneğin trafiğin hızlı aktığı otoyollarda 1-2 saniye geleceği tahmin etmek yeterli olabileceği gibi normal seyirdeki kent trafiklerinde 3-5 saniye; trafiğin sıkışık olduğu durumlarda 5 saniyeden daha uzun zaman dilimleri için tahminlere ihtiyaç duyulabilmektedir. Bu tez çalışması kapsamında da farklı zaman dilimleri için rota tahmini gerçekleştirilmiş ve modelin sonuçları raporlanmıştır.

Elde edilen sonuçlara bakıldığında geliştirilen yöntem ve yapay zekâ modelinin rota tahmin süreçlerine olumlu katkı sunarak yol gösterici bir çalışma olduğu görülmektedir.

Literatür incelendiğinde nesne algılama aşamasında LiDAR nokta bulutu tabanlı yapılan bazı çalışmalar aşağıdaki gibidir:

Yang vd. [29] PIXOR adını verdikleri çalışmalarında 3B nesne algılama için yeni bir derin öğrenme tabanlı model ve ATG4D adını verdikleri yeni bir veri seti sunmuşlardır. Modelde öncelikle LiDAR sensörü ile elde edilen 3B nokta bulutu kuş bakışı görüntü formatına dönüştürülmüştür. Daha sonra CNN katmanları, Artık Ağ [30] ve üst örnekleme işlemleri ile özellik haritaları elde edilmiştir. Elde edilen özellikler tahminin yapıldığı CNN katmanlarına gönderilerek nesnenin sınıfı ve çevreleyen kutusu bulunmuştur. KITTI [31] veri seti üzerinde araçlar için %75,74 ortalama hassasiyetle (AP) algılama işlemi gerçekleştirilmiştir.

Meyer vd. [32] LaserNet adını verdikleri 3B nesne algılama çalışmalarında LiDAR verisini mesafe görünümü formatında kullanmış ve çevreleyen kutuyu bulma işlemini olasılık dağılımı olarak ele almışlardır. Öncelikle nokta bulutundaki noktaların nesne sınıfını tespit etmek ve çevreleyen kutu olasılık dağılımı işlemlerinde kullanılacak parametre değerlerini elde etmek için CNN tabanlı bir segmentasyon işlemi yapılmıştır. Sistem her nokta için nesne sınıf olasılığı ve bir çevreleyen kutu parametre olasılık dağılımı üretmektedir. Aynı nesneye ait noktaların aynı olasılık dağılımı sonucunu üreteceği kabulüyle aynı dağılıma sahip çevreleyen kutuların merkez noktaları üzerinden ortalama işlemleriyle nesne için çevreleyen kutu üretilmiştir. ATG4D veri seti üzerinde 0-70 m aralığındaki arabalar, yayalar ve bisikletliler için sırasıyla %85,34, %61,93 ve %80,37 AP değeriyle nesne algılama işlemi gerçekleştirilmiştir.

Yin vd. [33], CenterPoint adını verdikleri çalışmalarında, LiDAR verilerinden 3B nesne algılama için nesnenin merkez noktasına odaklanmışlardır. VoxelNet [34] algoritmasının omurga model olarak kullanıldığı versiyonda LiDAR verisi, düzenli 3B voksel ızgarasına dönüştürülmüştür. Modelde iki aşamalı olarak nesne algılama işlemi yapılmıştır. Birinci aşamada, CenterNet [35] algoritması tabanlı bir yapı ile nesnelerin çevreleyen kutularının merkezlerine dayalı yoğunluk haritası, ofset değerleri ve buna bağlı olarak çevreleyen kutu merkez noktası koordinat bilgileri, boyutları, yönelim açıları ve hızları tahmin edilmiştir. İkinci aşamada her nesne için bir güven skoru ve çevreleyen kutu iyileştirme değerleri hesaplanıp son çevreleyen kutular elde edilmiştir. NuScenes [36] veri seti üzerinde %67,3 ve WAYMO Perception [37] veri seti üzerinde ikinci seviye zorluk derecesindeki araba ve yayalar için %72,2 ortalama ortalama hassasiyet (mAP) ile nesne algılama işlemi gerçekleştirilmiştir.

Fan vd. [38] RangeDet adını verdikleri çalışmalarında LiDAR sensöründen elde edilen mesafe görünümü formatındaki veriyi kullanarak nesne algılama işlemi gerçekleştirmiştir. Kartezyen koordinat düzlemindeki noktalardan öncelikle Çok Katmanlı Algılayıcı (Multi Layer Perceptron) ağ ile nokta bulutunun özellik bilgisi çıkarılmıştır. Aynı uzaklıktaki nesnelerin aynı katmanda işlem görmesini sağlayan yeni bir Özellik Piramidi Ağı [39] geliştirilmiştir. Piramidin farklı katmanlarında elde edilen bilgiler kullanılarak evrişim işlemiyle nesnelerin sınıflarının ve çevreleyen kutularının bulunması WAYMO Perception [40] veri seti üzerinde araç, yaya ve bisikletli için sırasıyla %72,9, %75,9 ve %65,8 AP değeriyle gerçekleştirilmiştir.

Sun vd. [41] tarafından yapılan çalışmada, öncelikle mesafe formatındaki LiDAR verisi üzerinde anlamsal segmentasyon yapılarak arka plan bilgileri elenip sadece ön plan nesnelere ait bilgiler elde edilmiştir. Bu aşama için 2B evrişim ve örnek azaltma-arttırma işlemlerinden oluşan kodlayıcı-çözümleyici yapıya sahip yapay sinir ağı modeli kullanılmıştır. Daha sonra ön plan

nesne bilgileri 3B vksel ızgaraları olarak ifade edilmiş ve 2B ile 3B evriřim iřlemleri kullanılarak nesnelere ait zellik haritaları elde edilmiştir. Son olarak CenterNet [35] tabanlı geliřtirdikleri tahmin ařamasıyla nesne algılama iřlemi gerekleřtirilmiştir. Zaman ierisinde otonom aracın hareketinin nesnelere ait verilere etkisini de ğrenmek iin  ereve bilgisi beraber kullanılmıştır. WAYMO Perception veri seti zerinde arabalar iin kolay ve zor tespit edilen nesneler aısından sırayla %78,4 ve %69,5; yayalar iin kolay ve zor tespit edilen nesneler aısından sırayla %79,4 ve %69,9 AP ile nesne algılama iřlemi yapılmıştır.

K. Hu vd. [42] alıřmalarında, LiDAR sensryle elde edilen verilerde grece daha kk olan veya uzaktaki nesnelerin daha az nokta ile ifade edildiğini ve bu durumun nesnelerin algılanmasında hassasiyeti dřrdğn belirtmiştir. Bu problemi zmek iin iki ařamalı bir mimari geliřtirilmiştir. ncelikle 3B LiDAR verilerinden vksel ızgara yapıları elde edilmiş ve vksellerin merkez noktaları tespit edilmiştir. Elde edilen bu bilgilerden Blge neri Modeli [43] ile olası nesne blgeleri bulunmuřtur. Nokta yoğunluğn dikkate alan İlgi Blgesi Havuzlama sistemi ve sonrasında z Dikkat Ağı [44] ile nesne algılamak iin odaklanması gereken alanlar tespit edilmiştir. Bu alanlar zerinde İleri Beslemeli Ağlar [45] ile nesne tr iin gven skoru ve evreleyen kutu bilgileri bulunmuřtur. WAYMO Perception veri seti zerinde en az bir LiDAR noktası barındıran ara, yaya ve bisikletli algılaması iin sırasıyla %69,30, %65,85 ve %66,49 AP değeri elde edilmiştir.

Huang vd. [46] MENet adını verdikleri alıřmalarında, LiDAR verilerinin voksele dnřtrlmř hali ile yksek znrlkl harita bilgisini birlikte kullanarak nesne algılama iřlemi gerekleřtirmişlerdir. Kodlama ařamasında LiDAR verileri ve harita verileri zerinde ayrı ayrı evriřim ve rnek azaltma iřlemleriyle zellik haritaları elde edilmiş daha sonra bu zellikler birleřtirilip z Dikkat Ağlarından geirilmiştir. zmleyici katmanında evriřim, rnek azaltma/arttırma ve normalizasyon iřlemleriyle nesnelerin tr ve evreleyen kutuları nuScenes [36] veri seti zerinde %56,90 mAP değeri ile elde edilmiştir.

Y. Chen vd. [47] VoxelNext adını verdikleri alıřmalarında vksel formatına evrilmiş LiDAR verileri zerinde seyrek evriřim katmanlarıyla zellik ıkarımı yapmışlardır. Diğerk alıřmalarda Artık Ağ yapısında drt katman olarak kullanılan ařağı rneklemeye yapısının byk nesnelerin tespitinde verimli alıřmadığını belirterek iki adet daha katman eklemişler ve altı katmanla seyrek vksel zelliklerinden daha detaylı bilgi elde edebilmişlerdir. Son  katmandaki verilerin bir araya getirilmesiyle elde edilen vksel zelliklerinden aynı konuma denk gelen veriler birleřtirilmiştir. Son olarak evriřim ve tam bağı katmanlar ve maksimum olmayan değerklerin

baskılanmasıyla nesne türü ve çevreleyen kutular nuScenes veri setinde %66,2 mAP ile tespit edilmiştir.

J.Wang vd. [48] çalışmalarında, voksel ızgaralarının içerisinde çok sayıda boş voksel bulunmasının işlem performansı açısından negatif etkiye sahip olmasına odaklanmışlardır. Bu nedenle öncelikle voksel ızgarası olarak ifade ettikleri LiDAR verilerini birbiriyle kesişmeyen ve içerisinde boş olmayan vokselleri barındıran pencerelere bölmüşlerdir. Bu pencerelerdeki vokseller arasında Öz Dikkat Ağları aracılığıyla bilgi geçişleri yaparak özellik çıkarımı yapılmış ve bu özellikler üzerine uygulanan kayan pencereler yöntemiyle de yeni sözde görüntüler oluşturulmuştur. Bu yeni sözde görüntüler Özellik Piramidi Ağına gönderilerek yeni özellik haritaları elde edilmiştir. SSD [49] algılama algoritmasının tahmin basamağı kullanılarak KITTI veri setinin kolay ve zor tespit durumlarında sırasıyla arabalar için %89 ve %79,61, yayalar için %66,21 ve %53,73 ve bisikletliler için %80,10 ve %59,81 AP ile algılama işlemi gerçekleştirilmiştir.

Qian vd. [50] çalışmalarında LiDAR sensörü pahalı olduğu için daha ucuz olan kameralardan elde edilen görüntüler ile sözde LiDAR verisi oluşturarak nesne algılama işlemini yapmak istediklerini belirtmişlerdir. Sözde LiDAR verisi oluşturma alanında yapılan diğer çalışmalarda, derinlik bulma ve nesne algılama algoritmalarının ayrı ayrı eğitildiği ve uçtan uca bir ağ geliştirildiğinde hata oranının arttığı rapor edilmiştir. Bu nedenle derinlik ve nesne algılama süreçlerini uçtan uca gerçekleştiren bir çalışma sunmuşlardır. Çalışmada öncelikle sağ ve sol olmak üzere iki kamera aracılığıyla elde edilen görüntüler üzerinden You vd. tarafından [51] geliştirilen hazır bir model ile derinlik algılama işlemi yapılmıştır. Nokta bulutu oluşturulurken iki kameradan alınan görüntülerden ayrı ayrı CNN modeli ile özellikler elde edilmiş, sonuçlar farklı boyutlardaki aşağı örnekleme piramidinden geçirilmiş ve en son elde edilen özellikler birleştirilerek 3B özellik tensörü oluşturulmuştur. Son aşamada veriler 3B CNN'den geçirilerek derinlik haritası bulunmuştur. Elde edilen derinlik bilgileri ile PointRCNN [52] algoritması kullanılarak KITTI veri setinde kolay tespit edilebilir arabalarda %79,6; zor tespit edilebilir arabalarda %52,1 AP ile nesne algılama işlemleri yapılmıştır.

Nesneler algılandıktan sonra bu nesnelerin 3B nokta bulutu çerçeveleri arasında takip edilmesi aşamasına geçilmektedir. 3B nesne çevreleyen kutularını kullanarak nesne takibi yapan bazı çalışmalar aşağıdaki gibidir:

Luo vd. [53], çalışmalarında algılama ve takip işlemlerini beraber gerçekleştirmişlerdir. Algılama için geliştirilen ve CenterPoint [33] algoritması tabanlı olan model ile LiDAR verilerinden yoğunluk haritaları, çevreleyen kutu ve hız bilgileri elde edilmiş ve takip aşamasına

girdi olarak gönderilmiştir. Takip aşamasında, otonom araca ait dönüşüm matrisi kullanılarak ilk çerçeveden elde edilen yoğunluk haritası dönüştürülüp ikinci çerçevede olması beklenen yoğunluk haritası hesaplanmıştır. Hesaplanan bu yeni harita ile ikinci çerçeve için algılama sonucu elde edilen yoğunluk haritasının ortalaması alınarak küçük değerler elenmiştir. Merkezi aynı koordinatlara denk gelen çerçeveler aynı nesneye ait kabul edilmiş ve nesnelerin kimlik numaraları ilk çerçeveden alınıp atanmış ve nesnelerin yörüngeleri de güncellenmiştir. Eğer nesne sadece ikinci çerçevede görünürse yalnızca ikinci yoğunluk haritası kullanılarak sisteme eklenmiş, eğer ikinci çerçevede görünmezse yoğunluk haritası oluşturulmamıştır. Bu yöntemle, yeni ortaya çıkan ve izi kaybolan nesnelerin de belirlenmesi sağlanmıştır. Sistem, nuScenes veri seti üzerinde genel sistem performansı açısından %66,1 ortalama çoklu nesne takip doğruluğu (AMOTA) ile sonuç üretmiştir.

Kim vd. [54], EagerMOT adını verdikleri çalışmalarında, LiDAR ve kamera verilerini birleştirerek 3B çoklu nesne takibi gerçekleştiren yeni bir yöntem geliştirmişlerdir. Öncelikle, LiDAR verisinden CenterPoint modeli ile 3B nesne algılama, kamera verilerinden Mask R-CNN [55] modeli ile 2B algılama gerçekleştirilerek nesnelerin bilgileri elde edilmiştir. Elde edilen sonuçlar sensör füzyon yöntemiyle birleştirilerek nihai 2B ve 3B nesne algılama sonuçları oluşturulmuştur. Kalman Filtresi [56] kullanılarak birinci çerçevede tespit edilen nesneler için ikinci çerçevede bulunacakları konum tahmin edilmiştir. İkinci çerçeve için elde edilen tahmin sonuçları ve ikinci çerçeve için gerçekleştirilen nesne algılama sonuçları kullanılarak iki aşamadan oluşan nesne eşleştirme yöntemi uygulanmıştır. Birinci aşamada, ikinci çerçeveden 3B algılama ile elde edilen nesne konumu, boyutu ve hız bilgileri ile nesnenin eski iz bilgileri kullanılarak Açgözlü Algoritma (Greedy Algorithm) [57] ile nesne eşleştirmesi yapılmıştır. İkinci aşamada ise elde edilen bu eşleştirmeler ve 2B algılama sonuçları kullanılarak yeni bir eşleştirme yapılmıştır. Elde edilen sonuçlar Kalman Filtresi'nden geçirilmiş ve izler güncellenerek nihai yörüngeler oluşturulmuştur. Çalışmada nuScenes veri setinde AMOTA açısından %68 doğrulukla sonuçlar elde edilmiştir.

Chiu vd. [58] çalışmalarında, LiDAR ve kamera verilerini birleştirerek 3D nesne takibi gerçekleştiren olasılıksal bir sistem geliştirmişlerdir. Çalışmada, LiDAR verilerinden 3B nesne algılama işlemi için CenterPoint [33] ve kamera verilerinden 2B algılama işlemi için Mask R-CNN kullanılmıştır. Kamera ve LiDAR'dan elde edilen veriler sensör füzyon yöntemiyle birleştirilmiştir. Takip sürecinde, nesnelerin özellikleri kullanılarak Kalman Filtresi ile birinci çerçevedeki nesnelerin ikinci çerçevedeki konumu tahmin edilmiştir. İkinci çerçeve için yapılan tahminler ve algılama sonuçları arasında Mahalanobis Mesafesi [59] yöntemiyle maliyet matrisi

oluşturulmuştur. Daha sonra en uygun eşleştirmeler, Açgözlü Algoritması kullanılarak tespit edilmiştir. Son olarak, eşleştirme sonuçları kullanılıp Kalman Filtresi'yle nesnelerin yörüngeleri güncellenmiştir. Sistem, nuScenes veri seti üzerinde AMOTA açısından %68,7 doğrulukla sonuç üretmiştir.

C. Zhang vd. [60] MotionTrack adını verdikleri çalışmalarında LiDAR ile kamera verilerini birleştirerek kullanan, çoklu nesne algılama ve takibini Dönüştürücü Ağ [44] tabanlı olarak uçtan uca gerçekleştiren yeni bir model geliştirmişlerdir. Algılama aşamasında VoxelNet [34] kullanılarak LiDAR nokta bulutlarından 3B nesne özellikleri çıkarılmış, ResNet [30] modeli ile kamera görüntülerinden görsel özellikler elde edilmiş ve sonuçlar Dönüştürücü Ağ ile birleştirilmiştir. Böylece algılama aşamasında, nesnelerin çevreleyen kutuları, merkez koordinatları, boyutları ve yönelim açıları elde edilmiştir. Takip aşamasında, bir önceki çerçevedeki yörüngelerle mevcut çerçevede algılanan nesneler arasında ilişkilendirme yapmak için Dönüştürücü Ağ tabanlı maliyet matrisi hesabı yapılmıştır. Maliyet matrisinin değerlerinin hesaplanması uzamsal ve zamansal bilgilerin birleştirilmesi, nesne merkezlerinin konumları, hız bilgileri ve yönelim açısı gibi özelliklerin dikkate alınmasıyla gerçekleştirilmiştir. Hesaplanan maliyet matrisi kullanılarak Açgözlü Algoritması ile eşleştirilmeler bulunmuştur. Eşleştirilemeyen algılamalar yeni nesneler olarak takip sistemine eklenirken, birkaç çerçeve boyunca tekrar algılanmayan nesneler takipten çıkarılmıştır. Kaybolan nesnelerin tekrar algılanması durumunda, geçmiş yörünge bilgileriyle ilişkilendirme yapılmıştır. Sistem, nuScenes veri setinde AMOTA açısından %55 doğruluk elde etmiştir.

X. Wang vd. [61], MCTrack adını verdikleri çalışmalarında LiDAR verilerini barındıran farklı veri setlerine uyumlu ve Kalman Filtresi tabanlı yeni bir algoritma geliştirerek hareketli nesne takibi yapmışlardır. Algoritmanın veri setlerinden bağımsız olabilmesi için, ön işlem adımında nesnelere ait çevreleyen kutu, doğrultu açısı, hız ve hızlanma gibi bilgiler temel bir veri yapısına dönüştürülmüştür. Sonra birbirini takip eden iki çerçevenin nesne bilgileri alınmış ve birinci çerçevedeki nesnelerin ikinci çerçevedeki konumları Kalman Filtresi ile tahmin edilmiştir. Algoritmada, nesnenin konum, boyut ve doğrultu açısı birbirinden farklı ifade özelliklerine sahip olması nedeniyle, her biri için ayrı bir Kalman Filtresi kullanılmış ve sonuçlar birleştirilmiştir. Nesneler için tahminler ve algılama sonuçları kullanılarak kuş bakışı görüntü ve kuş bakışı görüntülerde ile takip edilemeyen nesneleri takip edebilmek için mesafe görüntüsü oluşturulmuştur. Her iki görüntü türü için maliyet matrisleri hesaplanmıştır. Kuş bakışı görüntülerde Macar Algoritması [62] ve mesafe görüntülerinde Açgözlü Algoritma ile birbirinden bağımsız olarak eşleştirme gerçekleştirilmiş, ardından bu iki eşleştirme sonuçları birleştirilerek

nihai bir eşleştirme yapılmıştır. Daha sonra Kalman filtreleriyle, yörünge bilgileri güncellenmiştir. Ayrıca sistem, yeni ortaya çıkan nesneleri algılayıp bunları mevcut takip sistemi içine entegre edebilmiş ve kaybolan nesneleri tespit ederek takibi sonlandırabilmiştir. Sistem WAYMO Perception veri seti üzerinde test edilmiş ve araba, yaya ve bisikletliler için sırasıyla %74,65, %74,37 ve %71,30 çoklu nesne takip doğruluğu (MOTA) ile sonuçlar elde etmiştir.

Trafikteki hareketli nesnelerin algılanıp veri çerçeveleri arasında takip edilerek geçmiş hareket bilgileri elde edildikten sonra gelecekteki rotalarının tahmin edilmesi işlemine geçilmektedir. Rota tahmini için yapılan bazı çalışmalar aşağıdaki gibidir:

Ma vd. [63], TrafficPredict adını verdikleri algoritmalarında, trafikteki hareketli nesnelerin (yaya, bisikletli, araç) rotalarını tahmin etmeye çalışmışlardır. Çalışma kapsamında ayrıca ApolloScape adında yeni bir veri seti de sunulmuştur. Veri seti için kamera, LiDAR ve GPS sensörü kullanılarak gerçek trafik ortamından bilgiler toplanmıştır. Toplanan veriler üzerinden elde edilen ve global koordinat düzleminde ifade edilen hazır rota etiketleri kullanılarak bu çalışmada tahmin süreçleri işletilmiştir. Çalışmada trafikte tüm hareketli nesnelerin birbirinin hareketinden etkilendiği ve bu nedenle nesneyi tek başına değil etrafındaki diğer nesnelerle birlikte ele almak gerektiği ifade edilmiştir. Zamana bağlı (spatio-temporal) bir çizge modeli kullanılarak görüntüdeki nesneler düğüm, nesneler arası bağlantılar ise kenar olarak ele alınmıştır. Gözlem süresindeki tüm görüntüler arasında da aynı nesnenin ilişkileri kenarlar ile ifade edilmiştir. Sistemde nesnelerin rotasını tespit etmek için iki aşamalı LSTM kullanılmıştır: Nesne bazında LSTM, her bir nesnenin bireysel hareket özelliklerini; sınıf bazında LSTM ise sınıfa ait nesnelerin ortak hareket özelliklerini öğrenmiştir. Elde edilen özellikler birleştirilerek nesnenin hangi doğrultuda hareket edeceği kestirilmeye çalışılmıştır. Sistemde yeni oluşturulan veri seti üzerinde araba, yaya ve bisikletli için 2 saniye (sn) gözlem bilgilerine dayanarak gelecekteki 3 sn rota tahmininde sırasıyla 0,085, 0,080 ve 0,091 metre (m) ortalama yer değiştirme hatası (ADE) ve 0,131, 0,139 ve 0,150 m son yer değiştirme hatası (FDE) ile sonuçlar üretilmiştir.

Chandra vd. [64] TraPHic adını verdikleri çalışmalarında CNN ve LSTM kullanarak rota tahmini gerçekleştirmiş ve TRAF adını verdikleri yeni bir veri seti sunmuşlardır. Otonom aracın etrafındaki hareketli nesneler, belirlenen ufuk bölgesindekiler ve komşu nesneler olarak iki ayrı grupta ele alınmıştır. Ufuk bölgesi nesneleri, komşu nesneler ve otonom araç için birbirinden bağımsız olarak sırasıyla tam bağlı katman- LSTM-tam bağlı katman yapısına sahip bir model ile nesnelerin bireysel hareket karakteristiği elde edilmiştir. Bu bireysel bilgiler birleştirilerek ilgili gruptaki tüm nesnelerin bilgilerini barındıran yeni bir özellik tensörü oluşturulmuştur. Bu tensör üzerinde CNN ve maksimum havuzlama ile nesneler arasındaki ilişkiye ait özellikler elde

edilmiştir. Daha sonra birbirinden bağımsız olarak elde edilen zamansal-mekânsal ilişkilendirilmiş veriler LSTM'ye gönderilip rota tahmini gerçekleştirilmiştir. Gözlem süresi 3 sn ve tahmin süresi 5 sn olduğu durumda heterojen nesneleri barındıran TRAF veri seti üzerinde ADE ve FDE açısından sırasıyla 2,16 ve 6,99 m hata değerleriyle tahmin işlemi gerçekleştirilmiştir.

Jain vd. [65], çalışmalarında yayanın takip edebileceği rotayı olasılığa dayalı bir yaklaşımla hesaplamışlardır. Veri setinden yayalara ait eski rota bilgileri ve trafik ortamının anlamsal harita bilgileri (yollardaki trafik ışıkları, kavşak bilgileri vb.) kullanılarak kuş bakışı görüntü oluşturulmuştur. Bu görüntüler üzerinde ResNet-18 ve Özellik Piramit Ağı kullanılarak özellik haritaları elde edilip tahmin başlığına iletilmiştir. Bu başlıkla, başlangıçta bir ön dağılım tahmini yapıp ardından her zaman adımında bu dağılım güncellenerek gelecekteki yaya pozisyonlarının olasılık dağılımı iteratif olarak tahmin edilmiştir. Model, geçmiş 6 sn'lik (0,1 saniye aralıklarla 60 kare) gözlem verilerini kullanarak, gelecekteki 1, 3 ve 10 sn'lik (0,1 saniye aralıklarla 10, 30 ve 100 kare) zamanlar için yayaların rotasını tahmin etmektedir. Çalışma kapsamında geliştirilen veri seti üzerindeki testlerde ADE açısından 10 sn için 1,23 m; FDE açısından 1,3 ve 10 sn için sırayla 0,35, 0,62 ve 2,71 m hata değerleriyle tahmin işlemi yapılmıştır.

Li vd. [66] GRIP++ adını verdikleri çalışmalarında çizge tabanlı olarak ifade edilen veriler üzerinden rota tahmin işlemi yapmışlardır. Öncelikle gözlem sürecindeki tüm çerçeveler için nesneleri barındıran statik çizgeler; nesnelerin koordinat bilgileri için normalizasyon ve evrişim ile gömmeler oluşturulmuştur. Bu bilgiler beraber kullanılarak dinamik çizgeler oluşturulmuş ve üç katman halinde sırasıyla Çizge Evrişimli Ağ [67], Normalizasyon, Zamansal Evrişimli Ağ (Temporal Convolutional Networks, TCN) [68] ve normalizasyon işlemleriyle nesneler için yeni özellik haritaları oluşturulmuştur. Elde edilen bu zamana bağlı nesne özellikleri kullanılarak Geçitli Tekrarlayan Birim (Gated Recurrent Unit- GRU) [69] tabanlı kodlayıcı-çözümleyici yapı ile rota tahmini gerçekleştirilmiştir. 2 sn gözlem ile 5 sn gelecek tahmini ApolloScape veri seti üzerinde araç, yaya ve bisikletliler için sırasıyla ADE açısından 2,24, 0,7142 ve 1,8024 m; FDE açısından 4,0762, 1,3732 ve 3,4155 m hata değerleriyle gerçekleştirilmiştir.

Cheng vd. [70], trafikteki nesnelerin (araçlar, yayalar vb.) gelecekteki olası çoklu rotalarını tahmin etmek amacıyla DCENet adı verdikleri bir model geliştirmişlerdir. Her nesnenin geçmiş 8 adımlık (3,2 sn) iz bilgileri üzerinde TCN, Tam Bağlı katman, Öz Dikkat Ağı ve havuzlama ile nesne bazlı zamansal bağlam; iz, yönelim ve hız bilgilerinden elde edilen anlamsal haritalar üzerinde iki boyutlu evrişim, Tam Bağlı Katman, Öz Dikkat Ağı ve LSTM ile nesneler arası mekânsal bağlam oluşturulup sonuçlar birleştirilmiştir. Elde edilen sonuçlar Koşullu Varyasyonel Otomatik Kodlayıcı (Conditional Variational Autoencoder-CVAE) [71] ile gelecek rotanın

tahmininde kullanılacak olasılık dağılımının hesaplanması için kullanılmıştır. CVAE'den elde edilen örneklemeler üzerinde LSTM ile gelecekteki 12 adım (4,8 sn) için çoklu rota tahmini yapılmıştır. Daha sonra elde edilen bu çoklu rotalardan gerçek rotaya en yakın olan rota tercih edilerek sistemin sonuçları rapor edilmiştir. inD [72] veri seti üzerinde minimum ortalama yer değiştirme hatası (minADE) ve minimum son yer değiştirme hatası (minFDE) açısından sırasıyla 0,52 m ve 1,23 m hata değerleriyle sonuç üretilmiştir.

Zhang vd. [73] STINET adını verdikleri çalışmalarında LiDAR verisi üzerinde yaya tespiti ve gelecekteki rotalarının tahmini işlemlerini çizge tabanlı bir yöntemle gerçekleştirmişlerdir. LiDAR ile elde edilmiş nokta bulutu PointPillars [74] yöntemi kullanılarak 2B sözde görüntülere dönüştürülmüştür. Gözlem süresince kullanılacak bu yeni görüntüler üzerinde CNN tabanlı ResNet ve U-Net [75] kullanılarak yayalar için özellik haritaları elde edilmiştir. Bu haritalar üzerinde Faster R-CNN'de [76] kullanılan Bölge Öneri Ağı ile her zaman dilimi için nesnelerin olması muhtemel alanlar belirlenmiştir. Daha sonra hem bölge önerileri hem de nesne özellikleri birlikte kullanılıp düğüm özelliği oluşturulmuş ve bu düğümlerin kullanıldığı zamansal bir çizge meydana getirilmiştir. Çizge Dikkat Ağı [77] ile düğümlere ait gömmeler elde edilip sonuçlar tahmin ağına gönderilmiştir. WAYMO Perception veri seti üzerinde 1 sn gözlem ile 3 sn gelecek rota tahmininde ADE ve FDE açısından sırasıyla 0,3367 ve 0,576 m ile tahmin yapılmıştır.

Zhang vd. [78] çalışmalarında nesnelere ait koordinat bilgileri ve trafik ortamında bulunan yol, trafik işareti, kavşak vb. bilgileri kullanarak rota tahminini Çizge Dikkat ve Dönüştürücü Ağ tabanlı gerçekleştiren yeni bir model geliştirmişlerdir. Her zaman dilimi için oluşturulan çizgeler ve anlamsal trafik ortam bilgileri birleştirilerek Pozisyon Kodlaması [44] hesaplanıp sonuçlar Çizge Dikkat Ağına gönderilmiştir. Çizge düğüm gömmeleri ve gözlemdeki son çerçevenin bilgileri kullanarak Dönüştürücü Ağ tabanlı kodlayıcı-çözümleyici katman ile rota tahmini yapılmıştır. Lyft [79] veri seti üzerinde 3 sn gözlem ile gelecekteki 5 sn için rota tahmini ADE ve FDE açısından sırayla 0,838 m ve 1,957 m hata değerleriyle gerçekleştirilmiştir.

Mo vd. [80] çalışmalarında nesnelerin bilgilerini araç ve yaya/bisikletli olmak üzere iki düğüm türüyle ifade eden heterojen çizgeleri ve trafik ortamının anlamsal harita bilgilerini kullanarak rota tahmini gerçekleştirmişlerdir. Nesnelerin geçmiş bilgilerinin zamansal özellikleri GRU ile elde edildikten sonra bu bilgiler, etkileşimin öğrenilmesi için heterojen çizgelerin düğüm özellikleri oluşturulmasında ve ilgili nesne için trafik ortamının anlamsal harita bilgisinin elde edilmesinde kullanılmıştır. Heterojen çizge üzerinde kenar tiplerine göre ayrıştırılmış Çizge Dikkat Ağı kullanılarak nesnelerin birbiriyle etkileşim bilgisi elde edilmiştir. Daha sonra harita, nesnelerin hareket bilgileri ve heterojen çizge gömmeleri kullanılarak nesne türüne göre

özelleştirilmiş LSTM ile rota tahmini gerçekleştirilmiştir. Model, INTERACTION [81] veri seti üzerinde 1 sn gözlem ile gelecekteki 3 sn ve 8 sn için araçlar ve diğer nesneler için sırasıyla ADE açısından 0,19 ve 0,66 m; FDE açısından 2,97 ve 8,56 m hata oranlarıyla tahmin gerçekleştirmiştir.

Najibi vd. [24] çalışmalarında denetimsiz öğrenme tabanlı geliştirdikleri model ile nesne algılama, takip ve rota tahmini işlemi gerçekleştirmişlerdir. Sistemde önceden etiketlenmiş verilere ihtiyaç duymadan kümeleme algoritmalarıyla nesnelere ait etiketleme yapılarak sonuç üretilmiştir. Öncelikle NSFP [82] algoritmasında yapılan geliştirmelerle beraber birbirini takip eden iki LiDAR taramasında elde edilen nokta bulutları karşılaştırılarak nesnelere ait olabilecek noktalar ve bu noktaların hızları tespit edilmiştir. Daha sonra elde edilen bu bilgiler Otomatik Etiketleme aşamasına gönderilmiş ve denetimsiz öğrenme yöntemi olan kümeleme algoritması ile noktalar kümelerine ayrılmış ve etiket ataması yapılmıştır. Etiketler ve nokta bulutları kullanılarak PointPillars [74] algoritması ile nesnelerin sınıfları ve çevreleyen kutuları elde edilmiştir. Çevreleyen kutular kullanılarak Macar Algoritması ve Kalman Filtresi ile nesne takip işlemi gerçekleştirilmiştir. Takip sonucu elde edilen yörüngeler MultiPath++ [83] algoritmasına gönderilerek araç ve trafikte zarar görecekt nesneler (yaya, bisikletli gibi) açısından iki nesne sınıfı için rota tahmini yapılmıştır. WAYMO Perception veri seti üzerinde 1 sn gözlem ile gelecekteki 8 sn rota tahmini -eğitimde araçlar için önceden tanımlanmış etiketlerin de kullanıldığı durumda minADE ve minFDE açısından sırasıyla 1,89 m ve 4,79 m hata değerleriyle gerçekleştirilmiştir.

Zhang vd. [84] AI-TP adını verdikleri çalışmalarında trafik nesnelerini homojen bir çizge ile ifade edip düğüm ve kenar özelliklerini kullanarak kodlayıcı-çözümleyici yapısına sahip model ile rota tahmini gerçekleştirmişlerdir. Homojen çizgeler Çizge Dikkat Ağı katmanlarından geçirilerek düğüm gömmeleri elde edilmiş ve sonrasında gömmeler CNN tabanlı GRU kodlayıcı katmanına girdi olarak gönderilmiştir. Kodlayıcı katmanından elde edilen sonuçlar çözümleyici bloğunun ilk gizli katmanının durum bilgisi; gözlem sürecinin son zaman diliminde elde edilen veriler çözümleyici bloğunun girdi verisinin ilk hali olarak ele alınarak çözümleme işlemi gerçekleştirilmiştir. Çözümleyici bloğunda, tahmin zaman dilimi adedince birbirinin aynısı olan çözümleyici katman bulunmaktadır. Her çözümleyici katman ilgili zaman dilimi için rota bilgisini üretmektedir. Her çözümleyici katmanının gizli katman bilgisi ve sonuç bilgisi bir sonraki çözümleyici katmana girdi olarak gönderilmektedir. Çalışmada ApolloScape veri seti üzerinde 3 sn gözlem ile gelecekteki 5 sn için ağırlıklandırılmış toplam ADE ve FDE açısından sırasıyla 1,1559 m ve 2,1324 m hata değeriyle sonuç üretilmiştir.

Xu vd. [85] MVHGN adını verdikleri çalışmalarında trafik ortamındaki büyük araç, küçük araç, yaya, bisikletli ve motosikletli olmak üzere beş farklı nesne sınıfının düğüm olarak ifade

edildiği homojen çizge tabanlı rota tahmini gerçekleştirmişlerdir. Öncelikle nesnelere ait yörünge, nesne sınıfı ve doğrultu açısı bilgileri üzerinde korelasyon yapılarak her bir bilgi grubunun kendi içerisinde ilişkilendirilmesi gerçekleştirilmiştir. Daha sonra bu bilgiler kullanılarak Çizge Evrişimli Ağ ve TCN katmanıyla nesnelere ait düğüm gömmeleri elde edilmiştir. Elde edilen bu yeni çizge kullanılarak nesnelerin yakın komşuları arasındaki mikro etkileşim ve bölgelere ayrılmış ortamlardaki nesnelerin bölgesel etkileşimi dikkate alınarak makro etkileşim bulunmuş ve düğüm gömmeleri yeniden hesaplanmıştır. Elde edilen bilgiler Çizge Evrişimli Ağ ve TCN katmanlarıyla birleştirilerek tahmin katmanına gönderilmiştir. Tahmin katmanında GRU tabanlı kodlayıcı-çözümleyici kullanılmış ve ApolloScape veri seti üzerinde 3 sn gözlemle gelecekteki 3 sn için rota tahmini araç, yaya ve bisikletli için sırayla ADE açısından 2,1349, 0,7192 ve 1,649 m ve FDE açısından 3,8925, 1,3876 ve 3,0483 m hata değerleriyle gerçekleştirilmiştir.

Jia vd. [86] HDGT olarak adlandırdıkları modellerinde yolun yapısal elemanlarının bulunduğu harita bilgilerini ve nesnelerin geçmiş hareket bilgilerini kullanarak oluşturdukları heterojen çizgeler ile rota tahmini gerçekleştirmişlerdir. Çizgede düğüm olarak trafik ajanı (araç, yaya ve bisikletli tek tip olarak), trafik işaretleri, yol çizgileri; kenar olarak ajanlarla diğer düğüm tipleri arasındaki ilişkinin ele alındığı kenar tipleri kullanılmıştır. Harita elemanlarıyla ajanlar arasında kenar bağlantısı olup olmadığı bulunurken ajan ile eleman arasındaki mesafe dikkate alınmıştır. Elde edilen heterojen çizgeden Öz Dikkat Ağ ile düğüm gömmeleri, Çok Katmanlı Algılayıcı ağıyla da kenar gömmeleri elde edilmiştir. Elde edilen yeni çizge üzerinde Heterojen Çizge Dönüştürücü Ağ [87] ile düğümler arasındaki etkileşime dayalı olarak düğümlerin yeni özellik haritaları elde edilmiştir. Son aşamada ajan türüne göre özelleştirilmiş çok katmanlı algılayıcı ağlar kullanılarak her hareketli nesne için çoklu rota tahmini gerçekleştirilmiştir. WAYMO Open Motion [88] veri seti üzerinde 1sn gözlem ile gelecekteki 8 sn için rota tahmini minADE ve minFDE açısından sırasıyla 0,5933 ve 1,2055 m hata ile gerçekleştirilmiştir.

Zeng vd. [89] çalışmalarında çevredeki araçlar, yayalar ve bisikletliler için gelecekteki 6 saniyelik (12 örnekleme) rotaları tahmin etmek amacıyla, nesnelerin geçmiş 2 saniyelik (4 örnekleme) hareket bilgileri ile içinde bulunulan harita bilgisini birleştiren bir yöntem önermiştir. Modelde, nesnelerin geçmiş konum, hız, yönelim gibi dinamik bilgileri ile yol geometrisi, şerit sınırları ve trafik işaretlerini içeren yüksek çözünürlüklü harita verileri kullanılmıştır. Öncelikle, TCN ile nesnelerin geçmiş hareketlerinin zaman serisi özellikleri öğrenilmiş ve bu bilgiler nesneler arasında paylaşılmıştır. Harita bilgilerinin semantik ve geometrik özellikleri, Dönüştürücü Ağ tabanlı bir dikkat mekanizması ile işlenerek, her nesnenin hareketini etkileyen en önemli harita bölgelerinin özellikleri çıkarılmış ve şerit sınırları, kavşaklar ve yol eğrileri gibi faktörler

önceliklendirilmiştir. Daha sonra, nesne hareket özellikleri ve harita özellikleri kullanılarak, Dönüştürücü Ağ tabanlı bir tahmin modülü ile gelecekteki rotalar tahmin edilmiştir. Model, nuScenes veri seti üzerinde minADE ve minFDE açısından 10 iz arasından en küçük hataların dikkate alındığı durumda sırasıyla 1,54 ve 3,87 m hata oranıyla rota tahmini yapmıştır.

Tang vd. [90] çalışmalarında çizge olarak ifade ettikleri trafik verilerinden üç farklı komşuluk matrisi ve çoklu zamansal ve mekânsal çizge özellik bulma adımlarıyla rota tahmini gerçekleştirmişlerdir. Nesneler arasındaki mesafeye, nesnelerin birbiriyle çarpışma ihtimaline ve nesnelerin düğüm gömmelerinin çarpımıyla elde edilen düğüm bağımlılıklarına dayalı olarak komşuluk matrisleri oluşturulmuştur. Elde edilen komşuluk matrisleri kullanılarak çok katmanlı bir yapıda TCN ve Çizge Evrişimli Ağ kullanılarak nesnelerin hareket karakteristiği ve birbirleriyle etkileşimi öğrenilmiştir. Elde edilen çizge bilgileri LSTM tabanlı kodlayıcı-çözümleyici tahmin katmanına gönderilerek ApolloScape veri seti üzerinde 3 sn gözlemle gelecekteki 5 sn rota tahmini araç, yaya ve bisikletli için sırasıyla ADE açısından 1,985, 0,671 ve 1,6745 m ve FDE açısından 3,5842, 1,2652 ve 3,0792 m hata değerleriyle gerçekleştirilmiştir.

Amin vd. [91] TrajectoFormer adını verdikleri çalışmalarında, mekânsal-zamansal komşuluk etkilerini dikkate alan ve geçmiş 3,2 saniyelik hareket verilerini gözlemleyerek gelecekteki 4,8 saniyelik zaman için trafikteki arabaların rotasını tahmin eden Dönüştürücü Ağ tabanlı bir model önermişlerdir. Bu model, ego aracın çevresindeki araçlarla olan etkileşimlerini öğrenerek hem ego aracın hem de komşu araçların gelecekteki hareketlerini tahmin etmeyi amaçlamıştır. Veri ön işleme aşamasında, ego araç için çevresindeki en yakın 8 komşu araç seçilmiş ve bu araçların geçmiş konum, hız, yönelim gibi dinamik hareket bilgileri kullanılarak her bir araç için hareket özellikleri elde edilmiştir. Elde edilen bu özellikler kullanılarak Dönüştürücü Ağ tabanlı kodlama aşamasında arabalar arasındaki mekânsal-zamansal bağımlılıklarla geçmiş rota özellik haritası bulunmuştur. Daha sonra hesaplanan rota özelliklerine gözlem esnasındaki son çerçevedeki bilgiler de dahil edilerek, Dönüştürücü Ağ tabanlı çözümleyici ile hem ego aracın hem de komşu araçların gelecekteki rotaları tahmin edilmiştir. Modelin performansı, Kök Ortalama Kare Hatası (RMSE) metriği ile değerlendirilmiştir. Model, NGSIM [92] US-101 ve I-80 üzerinde sırasıyla 1,47 ve 1,56 m hata değerleriyle sonuç üretmiştir.

Awan vd. [93] çalışmalarında çarpışmayı önleyici bir yaklaşımla heterojen çizge tabanlı bir model geliştirmişlerdir. Öncelikle yaya, araç ve bisikletlilerden oluşan ve düğüm özelliği olarak nesnelerin koordinat bilgisinin kullanıldığı bir heterojen çizge oluşturulmuştur. Bu çizgeden global bazda farklı nesne türlerinin etkileşimini öğrenmek için İlişkisel Çizge Evrişimli Ağı (Relational Graph Convolutional Network- R-GCN) [94] ve yerel bazda alt çizgeler oluşturularak nesnelerin

etkileşimini öğrenmek için Çizge Evrişimli Ağ kullanılmıştır. Heterojen etkileşim bilgileri bir önceki çerçeveden elde edilen bilgiler ile dikkat mekanizmasına gönderilmiştir. Böylece farklı zaman dilimlerindeki nesne etkileşimleri birleştirilmiştir. Daha sonra dikkat tabanlı heterojen ilişkiler kullanılarak çarpışmayı önlemek için Pekiştirmeli Öğrenme [95] yöntemiyle nesnelerin hangi yöne gitme eğiliminde olacağı hesaplanmıştır. Elde edilen bütün bilgiler nesne sınıfı bazlı oluşturulmuş olan LSTM yapısına gönderilerek nihai rotalar tahmin edilmiştir. 2 sn gözlem ile gelecekteki 3 sn için rota tahmini Argoverse 2 [96] veri setinde ADE ve FDE açısından 0,501 ve 1,028 m hata değerleriyle gerçekleştirilmiştir.

Casas vd. [25] tarafından geliştirilen DeTra çalışmasında LiDAR verileri kullanılarak algılama, takip ve rota tahmini işlemi yapılmıştır. Öncelikle LiDAR verilerinden PointNet ile LiDAR özellikleri çıkarılmış daha sonra ResNet, öz dikkat ve dinamik evrişim tabanlı bir omurga ile farklı boyutlarda özellik haritaları elde edilmiştir. Ayrıca LaneGCN altyapısıyla harita bilgilerinin özellikleri elde edilmiştir. Elde edilen bu özellikler kullanılarak nesneler arasındaki etkileşimi, nesne harita etkileşimini ve nesnenin geçmiş hareket karakteristiğini öğrenmek için özelleştirilmiş Dönüştürücü Ağ katmanlarından oluşan bir modelle yeni rotaların tahmini için veri oluşturulmuştur. Elde edilen bu verilerden GRU ve tam bağlı katman aracılığıyla sadece arabalar için rota tahmini gerçekleştirilmiştir. 0,5 sn izlemle gelecekteki 5 sn için tek rota dizisi oluşturulduğunda ADE ve FDE açısından sırasıyla WAYMO Perception veri seti üzerinde 1,54 ve 3,59 m; Argoverse 2 Sensor veri seti üzerinde 1,87 ve 4,29 m hata ile sonuç üretilmiştir.

Bu tez çalışmasının kitabının yapısı aşağıdaki gibidir:

Bölüm 1’de otonom araçlar ve tam otonom araçların trafiğe çıkması için gerek duyulan ihtiyaçlardan, rota tahmini için gerçekleştirilmesi gereken aşamalardan ve bu aşamalarda yapılan çalışmalardan, çalışmaların geliştirilmesi gereken yönlerinden ve bu tezin rota tahminine katkısından bahsedilmiştir.

Bölüm 2’de otonom araçlarda kullanılan sensörlerden, LiDAR sensörünün çalışma prensibinden, sensör verisinin işlenmesinden, nesne algılama-takip ve rota tahmin işlemlerinin genel bilgileri ve değerlendirme kriterlerinden, verilerin çizge ile temsilinden, heterojen çizge yapısından, derin grafik öğrenimi yöntemlerinden, zaman serisi verisi işlemede kullanılacak derin öğrenme yöntemlerinden, kullanılan optimizasyon yöntemi ve kayıp fonksiyonundan bahsedilmiştir.

Bölüm 3’te tez kapsamında geliştirilen yöntemin ve modelin detaylı bilgileri verilmiştir. Kullanılan veri setlerinden, nesne algılama ve takip işlemleri için kullanılan hazır modellerin genel bilgileri ve çalışma prensiplerinden, verilerin heterojen çizge ile temsil edilmesi için

gerçekleştirilen adımlardan, geliştirilen yöntemin ve modelin aşamalarının yapısından bahsedilmiştir.

Bölüm 4’te rota tahmin modelinin ve yönteminin sonuçları tablolar halinde verilmiştir. Bu aşamada farklı gözlem ve tahmin zaman dilimleri ve farklı veri setleri için elde edilen sonuçlar raporlanmıştır. Ayrıca geliştirilen modelin farklı zaman serisi işleme yöntemlerine göre sonuçları gösterilmiştir. Karşılaştırma için kullanılan çalışmalara göre tezin katkıları ele alınmıştır.

Bölüm 5’te tezin sonuçları ve sonraki çalışmalar için öneriler ifade edilmiştir.



2. OTONOM ARAÇLARDA ROTA TAHMİNİ İÇİN İHTİYAÇ DUYULAN DONANIM ve YAZILIM ALTYAPILARI

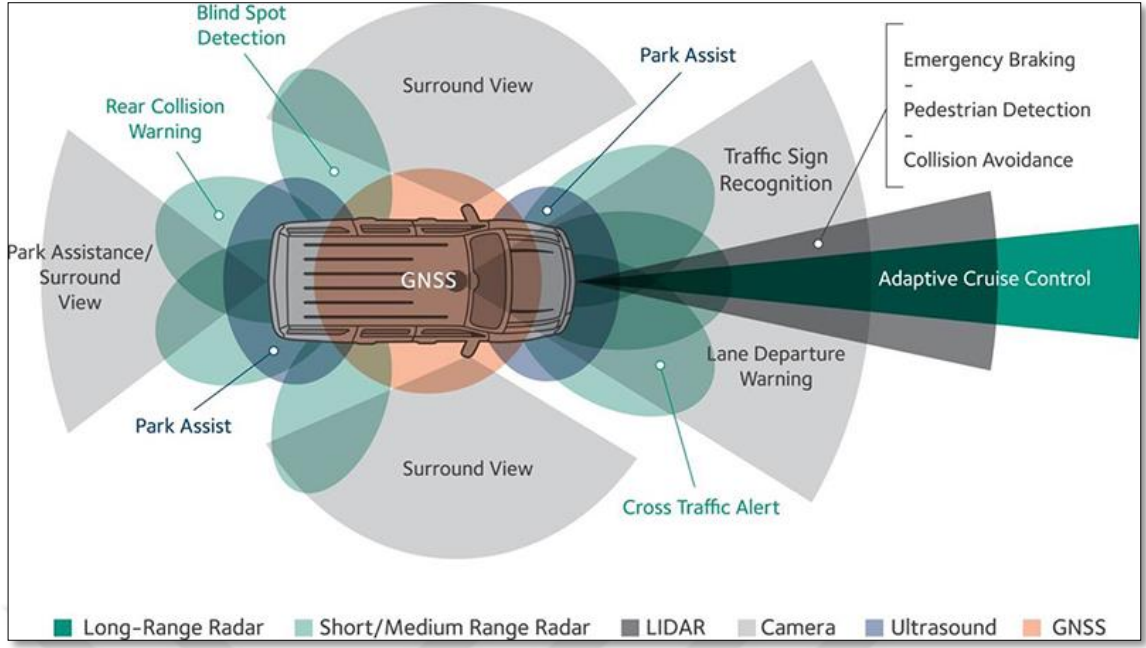
Otonom araç, sürüş süreçleri için yazılım ve donanımlar aracılığıyla gerekli kararları alıp hareket edebilen araçtır. Araçlardaki otonomluk derecesi 0-5 arasında altı seviyede değerlendirilmektedir. Sıfırıncı seviye, otonom herhangi bir özellik içermeyen kontrolün tamamen sürücüde olduğu seviyedir. Birinci seviyede, araca sürücü asistanı ve uyarı sistemleri eklenmiştir. Araç sürücüye hız kontrolünde yardımcı olur. İkinci seviye, kısmi otonomluk olarak adlandırılmakta ve araç sürücüye hız-direksiyon kontrolü, şerit takibi ve park gibi konularda yardımcı olabilmektedir. Koşullu otonomluk olarak ifade edilen üçüncü seviyede araç 60 km/s hızın altındayken etrafını algılayıp kendi kendine hareket edebilmektedir. Günümüzde kullanımda olan otonom araçlar üçüncü seviyedir [97]. Test aşamasında olan otonom araç çalışmaları dördüncü seviye olan yüksek otonom seviyesindedir ve karmaşık trafik koşulları hariç, özel olarak ayrılmış yolları kullanarak belirlenen rota dahilinde araç kendi kendine hareket edebilmektedir. Ulaşılmak istenen beşinci seviye tam otonom araçlarda, kontrolün tamamen araca ait olması; fren, gaz gibi pedallar ile direksiyona ihtiyaç olmaması; sürücünün yolcu durumunda olması ve aracın bir noktadan diğerine güvenli bir şekilde her türlü trafik ortamına uyum sağlayarak varması hedeflenmektedir [15].

Tam otonomluk seviyesine ulaşmak için Şekil 2.1’de de görülebileceği gibi birçok sürecin beraber işletilmesi gerekmektedir: Nesne algılama-takip-rota tahmini, park etme, trafik işaret tespiti ve yorumlama, kör nokta tespiti, çarpışma uyarı ve kaçınma, şerit takibi, acil fren vb.

Bu tez kapsamında nesne algılama, takip ve rota tahmini süreçlerinin uçtan uca bir yöntemle işletilmesine odaklanılmıştır. Bu bölümde rota tahmini için tam otonom bir aracın sahip olması gereken sensörlerin yapısı ile çalışma prensiplerinden ve yazılım altyapılarından bahsedilmiştir.

2.1 OTONOM ARAÇLARDA ÇEVRE ALGILAMA İÇİN KULLANILAN SENSÖRLER

Otonom araçların çevrelerini algılaması ve çevreden gelen bilgiye göre alacağı aksiyona (fren, şerit değiştirme, yavaşlama, durma vb.) karar vermesi gerekmektedir. Bu amaçla otonom araçlarda LiDAR, radar, kamera ve ultrasonik sensörleri bulunmaktadır.



Şekil 2-1: Otonom araçta bulunması gereken sistemler [98]

Radar sensörü, sensörden gönderilen elektromanyetik dalgaların nesnelerden yansırarak geri gelmesi prensibiyle nesnelerin yerini ve hızını tespit etmek için kullanılmaktadır. Nesnenin sensöre mesafesi, elektromanyetik dalganın hızı ve sinyalin toplam hareket süresinin yarısı çarpılarak hesaplanmaktadır.

Kameralar, aracın içerisinde bulunduğu trafik ortamının görsel algılamasını sağlayan sensörlerdir. Kameralar otonom araçların etrafındaki nesneleri, trafik ışık ve işaretlerini, yol çizgilerini vb. tespit etmek için kullanılmaktadır. Ayrıca LiDAR, radar vb. sensörlerin detaylı ve doğru sonuç oluşturamadığı durumlarda aracın çevresini tanıması için önemli bir sensördür [15]. Araçlarda tek (mono), çift (stereo) ve çoklu (multi) kamera kullanılabilir.

Ultrasonik sensör yakın mesafedeki nesneleri ses dalgaları ile tespit edebilen ve genellikle park etme sistemleri için kullanılan sensörlerdir [15]. Ses dalgalarının nesnelere çarparak geri dönme süresi ve sesin havada ilerleme hızına bağlı olarak nesnelerin mesafesi belirlenmektedir.

LiDAR sensörü ışık kaynağından gönderilen ışığın nesnelere çarparak geri dönmesine dayalı olarak veri üretmektedir. Sensörün detayları 2.1.1 başlığı altında anlatılmaktadır.

Otonom araçlarda çeşitli sensörlere ihtiyaç duyulmasının sebebi, her sensörün yerine getirebileceği belirli işlemlerin olması ve dezavantajlarının bulunmasıdır [99]. Örneğin, kamera görüntüden özellik çıkarımı, renk ve derinlik bilgisi elde edilmesi imkânı sağlarken ışık yetersizliğinde, yansımalarda, sisli ve aşırı yağışlı havalarda görevini yerine getirememektedir. Ayrıca elde edilen görseldeki nesnelerin gerçek mi yoksa başka bir resimden mi elde edildiğini

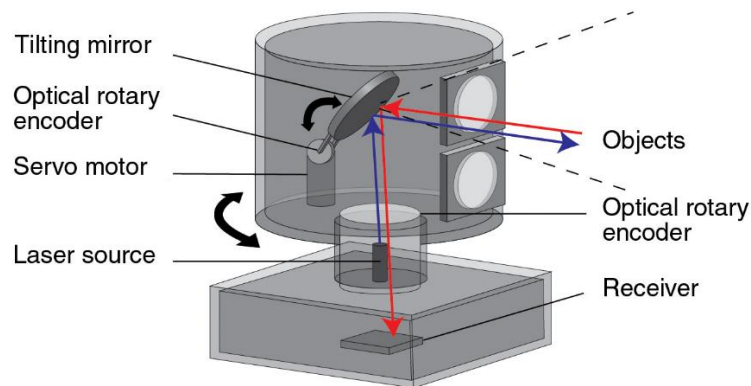
tespitte yetersiz kalmaktadır, mesela bir reklam afişindeki insan gerçek insan gibi algılanabilmektedir. LiDAR etraftaki nesnelerin yerini yüksek hassasiyetle belirleyebilip nesnenin ışığı yansıtmasından dolayı formundan gerçek olup olmadığını tespit edebilirken kötü hava şartlarından etkilenebilmekte ve yansıyan ışığı yeterince algılayamamaktadır. Ayrıca, renk bilgisine ihtiyaç duyulan trafik levhalarının yerini belirleyebilse bile gösterdiği bilgiyi elde edememektedir. Radar hava koşullarından etkilenmeden hız ve nesne tespiti yapabilirken hassasiyet değeri düşüktür, uzak mesafelerde iyi sonuç verememektedir [15].

Bu sorunları aşabilmek için çalışmalarda sensörlerden gelen bilgiler birleştirilerek kullanılabilir. Bu durumda işlem maliyetini arttırmakta ve gerçek zamanlı işleyiş açısından otonom araçlarda yüksek işlem kapasitesine sahip donanım ihtiyacına sebep olmaktadır.

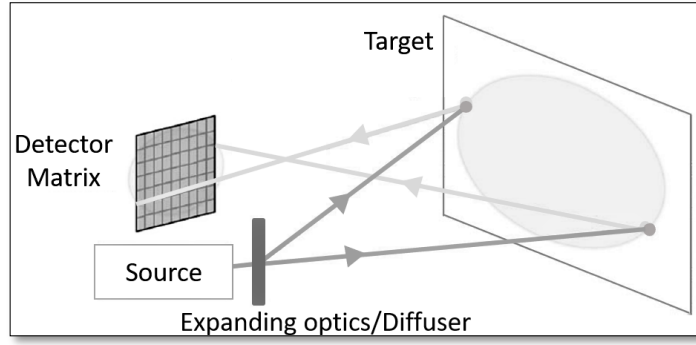
Bu tez çalışması kapsamında, hareketli nesnelere odaklanıldığı ve nesnelerin otonom araca uzaklığı bilgisine ihtiyaç duyulduğu için LiDAR sensöründen elde edilen veriler kullanılmıştır.

2.1.1 LiDAR Sensörü

Şekil 2.2’de iç yapısı görünen LiDAR sensörü, ışık kaynağından (lazer, ultraviyole, kızılötesi) gönderilen ışığın yansıtıcılara çarpıp geri dönmesi esnasında geçen süre ile nesnelerin uzaklığının tespit edilmesini amaçlayan sensör sistemidir [15, 100]. Şekil 2.3’te görüleceği gibi, LiDAR sensöründen çıkan lazer ışınları nesnelere çarparak yansımakta ve yansıyan ışık sensörün alıcıları tarafından algılanarak kaydedilmektedir.



Şekil 2-2: LiDAR sensörünün iç yapısı [101]



Şekil 2-3: LiDAR sensörünün çalışma prensibi [100]

Yatayda 360 derece etrafı tarayabilen sensörde, 950 nm ve 1550 nm olmak üzere canlıların gözlerine zarar vermeyen iki dalga boyu kullanılmaktadır ve kullanılan dalga boyuna göre tarama mesafesi 300 metreye kadar çıkabilmektedir [15, 102]. LiDAR sensörüyle mesafe, yükseklik açısı, azimut açısı, ışığın yansıma değeri ve lazer numarası bilgileri elde edilebilmektedir.

Sensör tarafından tespit edilen nesneye ait noktanın sensöre olan mesafesi d ; c ışık hızını ve t ışığın kaynaktan çıkıp nesneler tarafından yansıtıldıktan sonra tekrar sensör tarafından algılanması esnasında geçen süreyi ifade etmek üzere (2.1) numaralı denklem ile hesaplanmaktadır:

$$d = \frac{c \times t}{2} \quad (2.1)$$

Nesneye ait noktaların 3B kartezyen koordinat sistemindeki konumu (x, y, z) ; d mesafe, $\emptyset$ yükseklik açısı, θ azimut açısı olmak üzere (2.2, 2.3 ve 2.4) numaralı denklemler ile elde edilmektedir:

$$x = d \times \cos \emptyset \times \sin \theta \quad (2.2)$$

$$y = d \times \cos \emptyset \times \cos \theta \quad (2.3)$$

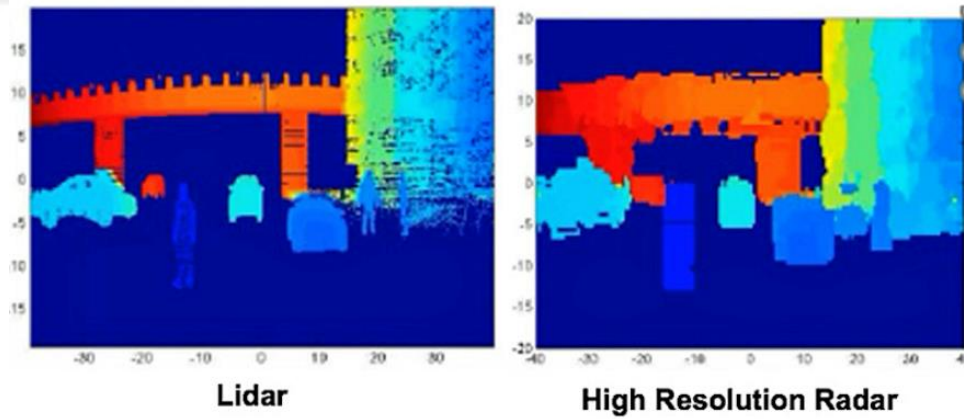
$$z = d \times \sin \emptyset \quad (2.4)$$

Şekil 2.4'te LiDAR sensöründen elde edilen bilgiler ile 3B kartezyen koordinat sistemine çevrilmiş nokta bulutunun görselleştirilmiş hali gösterilmektedir. Oluşturulan bu nokta bulutu ile nesnenin şekilsel özellikleri ve konumu elde edilmektedir.



Şekil 2-4: LiDAR ile elde edilen nokta bulutunun görselleştirilmesi[103]

LiDAR sensörüyle elde edilen verinin çözünürlüğü radar sensörüne göre daha yüksek olup LiDAR ile elde edilen verilerden daha hassas tespitler gerçekleştirilebilmektedir. Şekil 2.5'te LiDAR ve radar sensörleriyle aynı ortam için elde edilen bilgiler ile oluşturulan görüntüler gösterilmektedir.



Şekil 2-5: LiDAR ve radar ile aynı ortam için elde edilen görüntüler [104]

2.2 LiDAR VERİSİNİN İŞLENMESİ

LiDAR sensörüyle elde edilen veriler düzensiz seyrek nokta bulutu şeklindedir. Bu durum verinin ham haliyle işlenmesi için yüksek hafıza ve işlem gücü gerektirmektedir. Bu nedenle, LiDAR verisinin nesne algılama algoritmaları tarafından etkin bir şekilde işlenebilmesi için, ham

3B nokta bulutlarının daha düzenli ve analiz edilebilir formatlara dönüştürülmesi gerekmektedir. Bu dönüşüm işlemleri, verinin yapısını koruyarak algoritmaların performansını arttırmaktadır. Özellikle Voksel Izgara, Kuş Bakışı Görünüm (BEV) ve Mesafe Görünümü dönüşüm yöntemleri bu amaçla yaygın olarak kullanılmaktadır.

Voksel Izgara dönüşüm yönteminde, 3B nokta bulutu düzenli bir ızgaraya bölünmekte ve her bir hücre (voksel) belirli bir hacmi temsil etmektedir. Her voksel, içinde barındırdığı noktaların sayısı veya ortalama özellikleri gibi bilgilerle doldurulmaktadır. Bu dönüşüm sayesinde düzenli bir veri yapısı oluşturularak 3B Evrişimli Sinir Ağları (CNN) gibi derin öğrenme modellerinin kullanımı kolaylaştırılmakta ve nokta bulutunun seyrek yapısı yoğunlaştırılarak işlemler daha verimli hale getirilmektedir. Dönüşüm sırasında vokselin çözünürlüğünün seçimi kritiktir; düşük çözünürlük detay kaybına, yüksek çözünürlük ise bellek ve işlem maliyetinin artmasına neden olmaktadır [34].

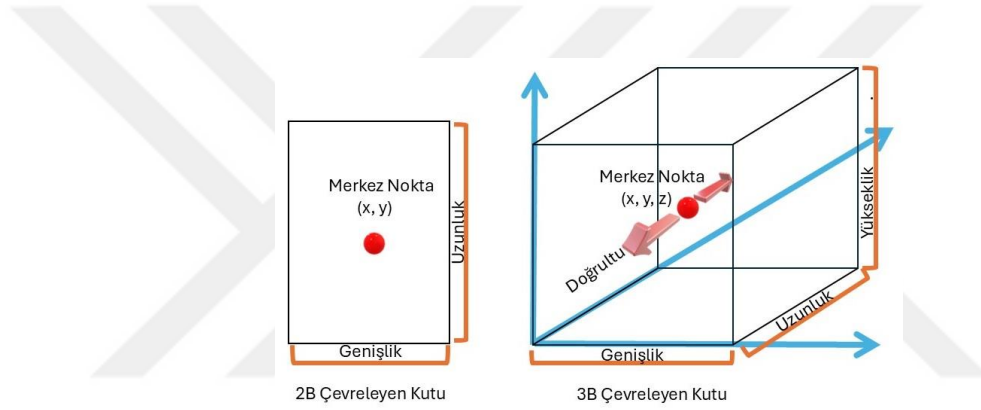
Kuş Bakışı Görünüm yönteminde 3B koordinat düzleminde ifade edilen LiDAR verisinin 2B koordinat düzlemine projeksiyonu gerçekleştirilmekte ve veri üstten görünüm perspektifine dönüştürülmektedir. Böylece veriyi 2B olarak işleyebilen daha hızlı ve optimize edilmiş algoritmalara uyumlu görüntü elde edilmektedir. Bu sayede 3B algılama işlemlerinin aksine daha az işlem gücü ve bellek kullanılarak veriye ait özellikler bulunabilmektedir. Bu dönüşümde yükseklik bilgisi veya nesnelerin üst üste binme durumları kaybolabildiği için havada hareket eden nesnelerin algılanması gerektiği durumlarda kullanıma uygun değildir. Ancak günümüz trafik ortamında sensör tarama alanına denk gelecek üst üste trafik nesneleri olmadığı için otonom araçlar açısından kullanılabilecek bir dönüşüm türüdür. Ancak uzak mesafedeki nesnelerin çözünürlüğü düşebilmektedir [29].

Mesafe Görünümü dönüşümü yönteminde LiDAR verisi, sensörün azimut ve yükseklik açılarının yatay ve dikey eksenlerini oluşturduğu yeni bir 2B düzleme projeksiyonu ile görüntüye dönüştürülür. Projeksiyon sırasında her noktanın mesafesi, yoğunluğu veya diğer özellikleri kaydedilir. Böylece sensörün doğal veri yapısına uygun bir dönüşüm yapılarak veriyi sıkıştırarak hesaplama maliyetleri azaltılmaktadır. Bu yöntemde çok yakın veya çok uzak noktalar arasında çözünürlük farkı oluşabilmektedir. Anlamsal segmentasyon yapılan uygulamalarda tercih edilmektedir [38].

Bu tez çalışması kapsamında nesne algılama aşaması için veri hazırlamak amacıyla LiDAR verisi voksel yapısına dönüştürülmüştür.

2.3 NESNE ALGILAMA ve DEĞERLENDİRME KRİTERLERİ

Nesne algılama, 2B görüntü veya 3B nokta bulutunda bulunan nesnelerin sınıfının ve çevreleyen kutusunun bulunması işlemidir [105]. 3B algılama ile 2B algılama arasında bazı farklılıklar bulunmaktadır. Örneğin, Şekil 2.6’da görüleceği üzere: 2B algılamada çevreleyen kutunun 2B kartezyen koordinat sisteminde merkez noktasının koordinatı, genişlik ve yükseklik bilgileri bulunurken 3B algılamada çevreleyen kutunun yükseklik, genişlik, uzunluk, 3B kartezyen koordinat sisteminde merkez nokta koordinatı ve kutunun doğrultusu bulunmaktadır. Bir diğer fark değerlendirme metriklerinde kullanılan tahmin edilen kutu ile gerçek kutunun örtüşme oranı (Intersection over Union- IoU) hesaplanmasında bulunmaktadır. IoU değeri, 2B’de alanların 3B’de ise hacimlerin örtüşmesi üzerinden hesaplanmaktadır.



Şekil 2-6: 2B ve 3B çevreleyen kutu

T tahmin edilen çevreleyen kutu ve G gerçek çevreleyen kutu olmak üzere (2.5) numaralı denklem ile hesaplanan IoU, bir eşik değeri (0,5, 0,75 veya 0,95) karşılaştırılmakta ve eşik değerin üzerinde olan sonuçlar başarılı kabul edilmektedir.

$$IoU = \frac{T \cap G}{T \cup G} \quad (2.5)$$

Nesne algılama işlemlerinde değerlendirme kriteri hesaplanırken nesnenin sınıfının doğru tahmin edilme oranı ve IoU değeri dikkate alınmaktadır. IoU ve nesne sınıfı tahmin sonuçları kullanılarak Tablo 2.1’deki gibi karışıklık matrisi elde edildikten sonra (2.6) ve (2.7) numaralı denklemler ile Kesinlik ve Duyarlılık değerleri hesaplanmaktadır. Daha sonra Kesinlik-Duyarlılık eğrisi elde edilip bu eğrinin altında kalan alanlar toplanarak (2.8) numaralı denklem ile nesne algılama işlemlerinde değerlendirme kriteri olan Ortalama Hassasiyet (AP) değeri bulunmaktadır.

Ortalama hassasiyetin 1'e yaklaşması sistemin doğru sonuçlar ürettiğini göstermektedir. Eğer eş zamanlı birkaç nesne sınıfı için tahmin yapılıyorsa ortalama hassasiyetlerin ortalamaları alınarak Ortalama Ortalama Hassasiyet (mAP) değeri bulunmaktadır.

Tablo 2-1: Karmaşıklık matrisi. Tablodaki değerler: 1. Doğru Pozitif: Gerçekte pozitif olanı modelin pozitif olarak işaretlemesi; 2. Yanlış Pozitif: Gerçekte negatif olanı modelin pozitif olarak işaretlemesi; 3. Yanlış Negatif: Gerçekte pozitif olanı modelin negatif olarak işaretlemesi; 4. Doğru Negatif: Gerçekte negatif olanı sistemin de negatif işaretlemesini ifade eder.

Karışıklık Matrisi	Gerçek Pozitif (1)	Gerçek Negatif (0)
Tahmin Pozitif (1)	<i>Doğru Pozitif (DP)</i>	<i>Yanlış Pozitif (YP)</i>
Tahmin Negatif (0)	<i>Yanlış Negatif (YN)</i>	<i>Doğru Negatif (DN)</i>

$$Kesinlik (K) = \frac{DP}{DP+YP} \quad (2.6)$$

$$Duyarlılık (D) = \frac{DP}{DP+YN} \quad (2.7)$$

$$AP = \sum_r (D_r - D_{r-1}) \times K(D_r)_r, r \in (0.0, 0.1, \dots, 1.0) \quad (2.8)$$

Nesne algılama işlemlerinde derin öğrenme tabanlı ilk çalışmalar genellikle 2B görüntü üzerinde gerçekleştirilmiş olup bu çalışmalar 3B ortamındaki uygulamalar için omurga model olarak kullanılmaktadır. 2B nesne algılama alanında bir aşamalı ve iki aşamalı olarak gruplandırılan algoritmalar geliştirilmektedir. Aşağıda açıklamalarına yer verilen algoritmalar birçok çalışma tarafından omurga model olarak kullanılmaktadır:

İki aşamalı olarak gerçekleştirilen nesne algılama işlemlerinde en temel algoritma Girshick vd. [106] tarafından geliştirilen CNN tabanlı R-CNN algoritmasıdır. Öncelikle resimde nesnenin olabileceği 2000 kadar bölge Seçici Arama yöntemiyle tespit edilmiştir. Daha sonra belirlenen her bölge için ayrı ayrı CNN ile özellik haritası çıkarılıp nesne algılama ve çevreleyen kutu bulma işlemleri gerçekleştirilmiştir. Nesnelerin sınıflandırması aşamasında Destek Vektör Makineleri kullanılmış ve çevreleyen kutu bulmak için regresyon işlemleri yapılmıştır.

R-CNN'nin yavaş sonuç üretmesi üzerine yeni bir versiyon olan Fast R-CNN [43] geliştirilmiştir. Hedef bölgelerin başlangıçta bulunması adımı çıkarılmıştır. Resim CNN'e direkt verilmiş ve elde edilen özellik haritalarından Seçici Arama yöntemiyle hedef bölgeler tespit edilmiştir. Belirlenen hedef bölgelerin boyutları İlgi Bölgesi Havuzlama tekniğiyle sabit boyutlara

dönüştürülmüştür. Bulunan sabit boyutlu bu hedef bölgeler tam bağlı katmana gönderilmiş, SoftMax algoritması ile sınıflandırma; çevreleyen kutu regresyonu ile konum belirleme işlemi yapılmıştır.

R-CNN ve Fast R-CNN algoritmaları hedef bölgeleri seçerken Seçici Arama algoritmasını kullandığı için gerçek zamanlı uygulamalar açısından yavaş kalmaktadır. Bu nedenle Ren vd. [107] tarafından Faster R-CNN algoritması geliştirilmiştir. Faster R-CNN’de hedef bölgeleri belirlemek için yeni bir tam bağlı yapay sinir ağı eklenmiştir. Bu sayede algoritmanın her aşaması Grafik Tabanlı İşlemci Ünitesi (GPU) ile çalışabilir hale getirilmiştir. Faster R-CNN’de girdi olarak verilen görüntüler üzerinde evrişim işlemleri ile elde edilen özellik haritaları Bölge Teklif Ağı’na gönderilmiştir. Ağ sonucu elde edilen bölgeler üzerinde İlgi Bölgesi Havuzlama işlemi yapılmıştır. Elde edilen sonuçlar ve ilk özellik haritaları kullanılarak sınıflandırma ve konumlandırma işlemleri gerçekleştirilmiştir. Faster R-CNN algoritması 3B algılama işleminde omurga model olarak Shi vd. [52] tarafından geliştirilen PointRCNN ve Y. Zhou vd. [34] tarafından geliştirilen VoxelNet çalışmalarında kullanılmıştır.

İki aşamalı algoritmaların gerçek zamanlı uygulamalar için yeterli hızı sağlayamayacağını bildiren araştırmacılar tek aşamalı nesne algılama algoritmalarına yönelmiştir. Bunlardan sık kullanılan bazı yöntemler aşağıdaki gibidir:

Redmon vd. [108] YOLO adını verdikleri çalışmalarında, nesnenin ağırlık merkezinin bulunması ve bu ağırlık merkezini çevreleyen kutunun tespit edilmesi üzerinde çalışmışlardır. CNN tabanlı olan algırmada öncelikle görüntü ızgara yapısına bölünmüş ve her bir ızgara parçasında herhangi bir nesnenin ağırlık merkezinin olup olmadığı araştırılmıştır. Eğer bir ızgara parçasında ağırlık merkezi tespit edilebilirse ona nesnenin çevreleyen kutusunu ve sınıfını bulma görevi verilmiştir. İlk sürümde aynı ızgara parçasına birden fazla nesnenin merkez noktası denk geldiğinde nesneler algılanamamaktaydı. Bu problemi çözmek adına Çapa Kutuları yöntemi eklenmiştir. Çapa kutuları yönteminde önceden tanımlanmış ve farklı nesneler için farklı en-boy oranına sahip çok sayıda dikdörtgen bulunmaktadır. Bu yöntemin kullanılmasıyla bir ızgarada nesnelerin olma ihtimali tespit edildiğinde çapa kutularıyla eşleşen nesneler araştırılmaya başlanmıştır. Bu yöntemde hassas sonuçların üretilebilmesi için çapa kutularının doğru tanımlanması kilit öneme sahiptir. Ayrıca bir diğer problem olan küçük nesnelerin tespiti zorluğunu aşmak için Özellik Piramidi Ağı eklenmiştir. Sonraki sürümlerde aktivasyon fonksiyonları ve parametre optimizasyonları ile sistemin hassasiyeti ve hızı arttırılmaya çalışılmıştır. Simon vd. [109] tarafından YOLO algoritması 3B algılama için uyarlanmış ve Complex-YOLO adıyla yayınlanmıştır.

Liu vd. [49], SSD adını verdikleri çalışmalarında VGG16 [110] tabanlı nesne algılama algoritması geliştirmişlerdir. VGG16 mimarisinin sınıflandırma katmanı çıkarılmış nesne sınıflarını ve çevreleyen kutuları belirlemek için 6 adet evrişim katmanı eklenmiştir. Farklı boyutlardaki nesneleri algılayabilmek için VGG16 ve eklenen 6 evrişim katmanının her birinden sonra çevreleyen kutu ve nesne sınıfı tahminleri yapılmıştır. Elde edilen bu tahminlerden en yüksek tahmin oranlarına sahip olan değerler kullanılıp diğerleri silinmiştir. Yeni sürümlerinde küçük nesneler üzerinde hassasiyet arttırımı için Gelişmiş Özellik Haritaları ve Dikkat Ağları kullanılmıştır. SSD algoritması, J. Wang vd. [48] tarafından geliştirilen 3B nesne algılama algoritmasının tahmin aşamasında kullanılmıştır.

Law ve Deng [111] CornerNet adını verdikleri çalışmalarında nesne için anahtar nokta olarak tanımladıkları çevreleyen kutunun köşe noktalarını bulmaya odaklanmışlardır. HourGlass [112] ağıyla elde edilen özellik haritalarında köşe noktalarını bulma işlemi için en büyük değerlerin toplanması mantığına dayalı Köşe Havuzlama işlemi yapılmıştır. Özellik haritaları üzerinde uygulanan köşe havuzlama işleminde üst-sol ve alt-sağ köşe noktaları için ayrı ayrı soldan sağa ve aşağıdan yukarıya doğru en büyük değer bulunmuş ve bulunan bu değerler toplanmıştır. Daha sonra bu değerler kullanılarak üst-sol ve alt-sağ köşe noktaları için yoğunluk haritaları, köşe noktası özellik bilgileri ve orijinal kutudan ne kadar konum kayması olabileceği bilgileri elde edilmiştir. Bu bilgiler kullanılarak aynı nesneye ait olan köşe noktaları için özellikler arasındaki farkın küçük olacağı esasına dayalı olarak nesnenin anahtar noktaları tespit edilmiştir. Böylece çapa kutuları kullanma ihtiyacını ve farklı boyutlardaki nesnelerin tespiti için farklı aşamalarda tahmin işlemi yapma gereksinimlerini ortadan kaldırdıklarını ifade etmektedirler.

Duan vd. [35] CenterNet adını verdikleri ve CornerNet çalışması tabanlı geliştirdikleri algoritmalarında nesnenin merkez noktasının tespitini de kullanarak çevreleyen kutu bulmaya çalışmışlardır. Köşe noktaları bulma işlemleri merkez noktası için de işletilmiş ve bu üç noktaya ait bilgiler kullanarak daha yüksek doğruluk oranıyla nesne algılama işlemi yapılmıştır. Sun vd. [41] ve Yin vd. [33] 3B algılama için geliştirdikleri algoritmalarında CenterNet'i omurga algoritma olarak kullanmışlardır.

Carion vd. [113] DETR adını verdikleri çalışmalarında nesne algılama prosedürünü zaman serisi verisi üzerinden tahmin problemi gibi ele almışlardır. Kodlayıcı-çözümleyici Dönüştürücü Ağ tabanlı geliştirdikleri algoritmalarında, Evrişimli Ağ ile elde ettikleri özellik haritalarını ve bu özellikler için elde ettikleri Konum Kodlama bilgilerini kullanarak nesne algılama işlemi gerçekleştirmişlerdir. Böylece çapa kutuları ve anahtar nokta bulma yöntemleri için gereken ön parametre belirleme işlemlerine ihtiyaç duymayan bir sistem geliştirmişlerdir. Zhu vd. [114] DS-

Trans adını verdikleri çalışmalarında altyapı olarak değiştirilmiş DETR kullanarak 3B nesne algılama gerçekleştirmişlerdir.

Bu tez çalışmasında geliştirilen uçtan uca yöntemde veri olarak LiDAR sensöründen elde edilen 3B nokta bulutu kullanıldığı için 3B nesne algılama süreçleri gerçekleştirilmiştir.

2.4 NESNE TAKİBİ ve DEĞERLENDİRME KRİTERLERİ

Nesne takibi, belirli bir zaman aralığında elde edilen görsel akıştaki çerçevelerde aynı nesnelerin izlerinin tespit edilip hareket yörüngesinin bulunması ve bu nesnelere çerçeveler boyunca aynı kimlik numarasının atanması işlemidir. Birbirini takip eden iki çerçeve arasındaki çevreleyen kutuların eşleştirilmesi Atama Problemi olarak ele alınmaktadır. f_{t-1} , bir önceki zaman diliminde şimdiki zaman dilimi için yapılan nesne takip tahminleri; f_t , şimdiki zaman dilimindeki çerçeve için nesne bilgileri olmak üzere bu iki zamandaki nesnelerin çevreleyen kutu sayısı sırasıyla n ve m olduğu bir durumda, bu iki zamandaki nesnelerin birbiriyle ilişki durumu $n \times m$ boyutlu bir maliyet matrisi (kare matrise çevrilerek) ile ifade edilmektedir. Maliyet matrisinin elemanları: f_{t-1} 'deki nesneler için yapılan tahmin ile f_t 'deki nesnelerin çevreleyen kutularının IoU değerleri, nesnelerin noktalarının birbirinden uzaklıkları veya başka uygun bir benzerlik ölçütü cinsinden maliyet değerleri olarak hesaplanmaktadır. Macar Algoritması gibi bir optimizasyon algoritması kullanılarak matristeki verilerin optimum değerleri hesaplanıp en iyi eşleşme değerine sahip olan çevreleyen kutular aynı nesneye ait olarak işaretlenmektedir. Daha sonra Kalman Filtresi vb. bir tahmin yöntemiyle eşleşen kutuların pozisyon, hız gibi durum bilgilerini güncellenerek bir sonraki çerçeve için durum tahmini yapılmaktadır. İlk defa karşılaşılan nesneler için bir sonraki çerçeve tahmini yapıldıktan sonra takip listesine dahil edilmektedir. Belirlenen süre içerisinde izi takip edilemeyen nesneler listeden çıkarılmaktadır.

Nesne takibi işlemlerinde değerlendirme ölçütü olarak genellikle çoklu nesne takip doğruluğu (MOTA) ve hassasiyeti (MOTP) değerleri kullanılmaktadır [115-116]. MOTA hatalı, kaybedilen ve uyumsuzluk gösteren izlerin ve yanlış kimlik numarası ataması oranının 1'den çıkarılmasıyla hesaplanmaktadır. MOTP ise çerçeveler arasında aynı olduğu tespit edilmiş nesnelerin konumunun gerçek konumundan farkı alınarak hesaplanmaktadır. Denklemde t ilgili zaman dilimi, Y_{P_t} yanlış pozitif sayısı, Y_{N_t} yanlış negatif sayısı, Y_{ID_t} yanlış eşleştirilen kimlik numarası sayısı, GT_t gerçek nesne bilgisi, $m_{i,t}$ numaralı nesnenin çevreleyen kutuları arasındaki mesafe ve d_t doğru eşleştirilen nesne sayısı olmak üzere MOTA ve MOTP sırasıyla (2.9) ve (2.10) numaralı denklemler ile hesaplanmaktadır:

$$MOTA = 1 - \frac{\sum_t (YP_t + YN_t + YID_t)}{\sum_t GT_t} \quad (2.9)$$

$$MOTP = \frac{\sum_{i,t} m_{i,t}}{d_t} \quad (2.10)$$

Nesne takibi alanında SORT [117] algoritması birçok çalışmaya yol gösterici olmuştur. SORT algoritmasında öncelikle birbirini takip eden iki çerçevedeki nesneler algılanmıştır. İlk çerçevedeki nesnelerin özellikleri kullanılarak Kalman Filtresi yardımıyla ikinci çerçevede nesnelerin nerede olabileceği tahmin edilmiştir. Daha sonra elde edilen bu tahminler ile ikinci çerçevede algılanan nesneler arasında IoU'a dayalı maliyet matrisi oluşturulmuştur. Maliyet matrisi üzerinde Macar algoritması çalıştırılarak optimum eşleşmeler tespit edilmiş ve Kalman filtresi ile izler güncellenmiştir. İlk defa karşılaşılan nesneler takip edilen izlere eklenirken, eşleşmesi kaybolan nesneler takipten çıkarılmıştır. Algoritma MOTA açısında %34 doğruluk ile takip işlemini gerçekleştirmiştir.

2B ve 3B nesne takibinde bazı farklılıklar bulunmaktadır. Mesela, IoU değeri üzerinden maliyet değeri hesaplanırken hacimsel örtüşme dikkate alınmalıdır. Noktalar arasındaki mesafe hesabında nokta bulutunun 3B özelliğine dikkat edilmelidir.

Bu tez çalışması kapsamında ardışık iki çerçeveden algılanan nesnelerin 3B çevreleyen kutularının eşleştirilmesi için 3B nesne takip süreçleri gerçekleştirilmiştir.

2.5 ROTA TAHMİNİ ve DEĞERLENDİRME KRİTERLERİ

Rota tahmini, hareketli nesnelerin belirli bir zaman aralığında gözlemlenmesiyle elde edilen hareket özellikleri kullanılarak belirli bir gelecek zaman aralığında takip edebileceği rotanın tahmin edilmesi işlemidir. t ilgili zaman dilimi, $t_{göz}$ gözlem yapılacak zaman aralığı, f_t ilgili zamandaki nesnenin özellik vektörü, t_{tah} tahmin yapılacak zaman aralığı, x_t ve y_t ilgili zaman için nesnenin tahmin edilen koordinat bilgileri olmak üzere rota tahmin işlemi (2.11) numaralı denklem ile ifade edilebilir:

$$\{t = 0, 1, \dots, t_{göz} \mid [f_t] \rightarrow \text{Yapay Zeka Modeli} \rightarrow \{t = t_{göz} + 1, \dots, t_{tah} \mid [x_t, y_t]\} \quad (2.11)$$

Rota tahmininde, nesneler için tek veya çoklu rota dizisi tahmini yapılabilmektedir. Bu tezde de kullanılan tek rota dizisi tahmininde değerlendirme kriteri olarak Ortalama Yer

Değiştirme Hatası (ADE) ve Son Yer Değiştirme Hatası (FDE) kullanılmaktadır. ADE hesaplanırken, nesnenin gelecekteki rotası için gelecek zaman aralığındaki tüm zaman dilimleri için tahmin edilmiş koordinatlarının referans bilgisindeki koordinat noktalarına uzaklığının ortalaması alınmaktadır. FDE hesaplanırken, varılacak son noktanın tahmin edilen koordinatıyla referans bilgisindeki son noktanın koordinatı arasındaki uzaklık hesaplanmaktadır [118]. N toplam nesne sayısını, n ilgili nesneyi, t ilgili zaman dilimini, T_p tahminde bulunacak zaman aralığını, (x_{tn}, y_{tn}) ilgili zaman diliminde nesnenin tahmin edilen konumunu, $(\widehat{x}_{tn}, \widehat{y}_{tn})$ gerçek konumu, $(x_{T_p n}, y_{T_p n})$ son bulunulması gereken nokta için tahmin edilen konumu, $(\widehat{x}_{T_p n}, \widehat{y}_{T_p n})$ son gerçek konumu ifade etmek üzere ADE ve FDE sırasıyla denklem (2.12) ve (2.13) ile hesaplanmaktadır:

$$ADE = \frac{\sum_{n \in N} \sum_{t \in T_p} \sqrt{(\widehat{x}_{tn} - x_{tn})^2 + (\widehat{y}_{tn} - y_{tn})^2}}{N \times T_p} \quad (2.12)$$

$$FDE = \frac{\sum_{n \in N} \sqrt{(\widehat{x}_{T_p n} - x_{T_p n})^2 + (\widehat{y}_{T_p n} - y_{T_p n})^2}}{N} \quad (2.13)$$

Trafik sürekli değişen, dinamik ve sosyal etkileşimin olduğu bir ortamdır. Bu nedenle rota tahmini için geliştirilen yöntemlerin sürekli değişebilecek yapıda olan veriyi, bu veri içerisinde farklı karakteristik özelliklere sahip nesneleri ve bu nesnelerin birbirini etkileme özelliklerini göz önünde bulundurması gerekmektedir. Bu bakış açısıyla tez kapsamında rota tahmini aşaması için derin öğrenme tabanlı model geliştirilmiştir.

2.6 ZAMAN SERİSİ VERİLERİNİN DERİN ÖĞRENME YÖNTEMLERİYLE İŞLENMESİ

Zaman serisi verileri, bir alandaki herhangi bir değer belirlenen zaman aralığında düzenli aralıklarla ölçülmesiyle elde edilen sıralı veri türüdür. Örneğin trafikteki nesnelerin konumlarının belirli zaman aralıklarıyla ölçülüp kaydedilmesiyle zaman serisi türünde bir veri elde edilebilmektedir. Bu yapıya sahip veriler üzerinde yapılan analizlerin temel amacı, verilerin geçmiş durumlarını inceleyerek gelecekteki eğilimlerini, davranış kalıplarını ve anomalileri tahmin etmektir. Zaman serilerini analiz etmek için kullanılan derin öğrenme yöntemlerinden bu tez kapsamında kullanılanlar aşağıdaki gibidir:

2.6.1 Uzun Kısa Süreli Hafıza (Long Short Term Memory -LSTM)

Hochreiter ve Schmidhuber [119] tarafından geliştirilen LSTM, zaman serileri ile çalışıp uzun süreli bağımlılıkları takip edebilen tekrarlayan sinir ağıdır. LSTM zaman serilerini her bir zaman dilimi için sırayla ele alıp ilgili zaman dilimi için sonuç üretmektedir. Bir LSTM hücresinin giriş veri düzeni: N yığın adedi, T zaman aralığı, F veriyi ifade ederken kullanılan özellik sayısı olmak üzere PyTorch kütüphanesi için yığının ilk değeri olarak belirtildiği durumda $[N, T, F]$ şeklinde olmalıdır.

LSTM hücresel durum adı verilen uzun vadeli bir hafıza birimi, ilgili zaman dilimi için üretilen gizli katman verisi ve bilgi akışını kontrol eden üç kapı mekanizmasından oluşmaktadır. Sigmoid (σ) fonksiyonu ile 0 ile 1 arasında değer üretilmesini sağlayan bu kapılardan: Giriş kapısı ile giriş verilerinin hangi kısmının hücresel duruma etki edeceği, unutma kapısı ile hücresel durum verisinin hangi kısmının unutulacağı ve çıkış kapısı ile hücresel durumun çıktıya hangi veriyi göndereceği belirlenmektedir.

Unutma kapısında t zaman dilimindeki verilerden hangilerinin unutulacağını barındıran unutma vektörü f_t ; W_f ağırlık vektörü, h_{t-1} bir önceki zaman diliminin gizli katman verisi, x_t ilgili zaman diliminin girdi verisi, b_f ön yargı değeri ve σ sigmoid aktivasyon fonksiyonu olmak üzere (2.14) numaralı denklem ile hesaplanmaktadır. Unutma vektöründeki değerler 0 ile 1 arasındadır ve giriş kapısındaki hesaplamalar esnasında 0'a yakın olanlara karşılık gelen hücre durumu verileri unutulurken 1'e yakın olanlar hafızada kalmaya devam etmektedir.

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (2.14)$$

Giriş kapısında t zaman diliminde giriş vektörü i_t ve yeni hücre durumu C_t ; W_i ve W_C ağırlık vektörü, h_{t-1} bir önceki zaman diliminin gizli katman verisi, x_t ilgili zaman diliminin girdi verisi, b_i ve b_C ön yargı değerleri, σ sigmoid aktivasyon fonksiyonu, C_{t-1} bir önceki zaman dilimindeki hücre durumu ve f_t unutma vektörü olmak üzere (2.15) ve (2.16) numaralı denklemler ile hesaplanmaktadır. Giriş vektöründeki değerler 0 ile 1 arasındadır ve giriş verilerinin hangilerinin hücre durumu güncellenirken kullanılacağına karar vermek amaçlı kullanılmaktadır.

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (2.15)$$

$$C_t = f_t \cdot C_{t-1} + i_t \cdot (\tanh(W_C \cdot [h_{t-1}, x_t] + b_C)) \quad (2.16)$$

Çıkış kapısında t zaman dilimindeki çıkış vektörü o_t ve yeni gizli katman verisi h_t ; W_o ağırlık vektörü, h_{t-1} bir önceki zaman diliminin gizli katman verisi, x_t ilgili zaman diliminin girdi verisi, b_o ön yargı değeri, σ sigmoid aktivasyon fonksiyonu, C_t ilgili zaman dilimindeki güncellenmiş hücre durumu ve $\tanh$ aktivasyon fonksiyonu olmak üzere (2.17) ve (2.18) numaralı denklemler ile hesaplanmaktadır. Çıkış vektöründeki değerler 0 ile 1 arasındadır ve hücre durumu verilerinin hangilerinin gizli katman sonucu hesaplanırken kullanılacağına karar vermek amaçlı kullanılmaktadır.

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (2.17)$$

$$h_t = o_t \cdot \tanh(C_t) \quad (2.18)$$

2.6.2 Geçitli Tekrarlayan Birim (Gated Recurrent Unit- GRU)

Cho vd. [69] tarafından geliştirilen GRU, zaman serileri üzerinde analiz yapabilen LSTM'ye göre daha basit bir yapıya sahip olan tekrarlayan sinir ağıdır. Aynı LSTM'de olduğu gibi GRU zaman serilerini her bir zaman dilimi için sırayla ele alıp ilgili zaman dilimi için sonuç üretmektedir. Bir GRU biriminin giriş veri düzeni N yığın adedi, T zaman aralığı, F veriyi ifade ederken kullanılan özellik sayısı olmak üzere PyTorch kütüphanesi için yığının ilk değeri olarak belirtildiği durumda $[N, T, F]$ şeklinde olmalıdır.

GRU, gizli durum ve bilgi akışını kontrol eden iki kapı mekanizmasından oluşmaktadır. Sigmoid (σ) fonksiyonu ile 0 ile 1 arasında değer üreten bu kapılardan: Güncelleme kapısı ile bilginin ne kadarının önceki gizli durumdan aktarılacağını ve ne kadarının şimdiki durumda yenileneceğini, resetleme kapısı ise önceki durumun ne kadarının unutulacağını belirlemektedir.

Güncelleme kapısında t zaman dilimindeki önceki gizli durumun ne kadarının korunacağını ve yeni bilginin ne kadarının ekleneceğini belirleyen güncelleme vektörü z_t ; W_z ağırlık vektörü, h_{t-1} bir önceki zaman diliminin gizli durumu, x_t ilgili zaman diliminin girdi verisi, b_z ön yargı değeri ve σ sigmoid aktivasyon fonksiyonu olmak üzere (2.19) numaralı denklem ile hesaplanmaktadır.

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t] + b_z) \quad (2.19)$$

Resetleme kapısında t zaman dilimindeki önceki gizli durumun ne kadarının sıfırlanacağını belirleyen resetleme vektörü r_t ve yeni gizli durum h_t ; W_r ve W_h ağırlık vektörleri, h_{t-1} bir önceki zaman diliminin gizli durumu, x_t ilgili zaman diliminin girdi verisi, b_h ve b_r ön yargı değeri ve σ sigmoid aktivasyon fonksiyonu olmak üzere (2.20) ve (2.21) numaralı denklemler ile hesaplanmaktadır.

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t] + b_r) \quad (2.20)$$

$$h_t = z_t \cdot h_{t-1} + (1 - z_t) \cdot \tanh(W_h \cdot [r_t \cdot h_{t-1}, x_t] + b_h) \quad (2.21)$$

2.6.3 Zamansal Evrişimli Ağ (Temporal Convolutional Network -TCN-Conv1D)

Bai vd. [68] tarafında geliştirilen TCN zaman serilerinin üzerinde bir boyutlu evrişim işlemleri uygulayarak analiz yapılmasını sağlamaktadır. TCN zaman serisi verilerini tek seferde alıp işlemektedir. Bu sayede paralel işlem yapılmasına olanak sağlamaktadır. TCN için giriş veri düzeni N yığın adedi, T zaman aralığı, F veriyi ifade ederken kullanılan özellik sayısı olmak üzere PyTorch kütüphanesi için $[N, F, T]$ şeklinde olmalıdır.

Genel olarak TCN'nin yapısında bir boyutlu evrişim modeli kullanılmaktadır ve bu modelde t zaman diliminde sonuç $y[t]$; i ilgili evrişim filtresi, $w[i]$ evrişim filtresinin ağırlık vektörü, r genişletme oranı, $x[t - r \cdot i]$ geçmiş girdi verisi ve k filtre uzunluğu olmak üzere (2.22) numaralı denklem ile hesaplanmaktadır.

$$y[t] = \sum_{i=0}^{k-1} w[i] \cdot x[t - r \cdot i] \quad (2.22)$$

Elde edilen sonuçlar normalizasyon ve aktivasyon fonksiyonlarından geçirilerek nihai değerler elde edilmektedir.

2.6.4 Dönüştürücü Ağlar (Transformer Network)

Vaswani vd. [44] tarafından geliştirilen Dönüştürücü Ağ, uzun vadeli bağımlılıkları öğrenebilen paralel işlem yapısına uygun bir yöntemdir. Dönüştürücü Ağın temel yeniliği, girişteki her bir verinin, diğer tüm verilerle olan ilişkisinin dikkate alınmasını sağlayan öz dikkat mekanizmasıdır. Dönüştürücü Ağ kodlayıcı ve çözümleyici olmak üzere iki bölümden

oluşmaktadır. Her iki bölümde de çok başlıklı öz dikkat ağı ve tam bağlı katmanlar kullanılmaktadır. Dönüştürücü ağların sadece kodlama aşaması, herhangi bir amaç için geliştirilmek istenen modeldeki girdi verilerinin hedeflenen sonuca etkisini ağırlıklandırmak istenildiğinde de kullanılabilir. Ayrıca, Dönüştürücü Ağ ile geliştirilen model için girdi verisindeki değerlerin sırası önemli ise Pozisyonel Kodlama katmanı ile her öğeye bir konum bilgisi de eklenmektedir.

Öz dikkat mekanizmasında girişteki her verinin diğer tüm verilerle olan bağımlı öğrenmek için öncelikle girdi verisi sorgu, anahtar ve değer ağırlık matrisleriyle çarpılarak Sorgu (Q), Anahtar (K) ve Değer (V) bileşenlerine dönüştürülmektedir. Daha sonra Q ve K 'nin transpozesi çarpılıp anahtar matrisinin boyutunun (d_k) kareköküne bölünerek dikkat skorları bulunmaktadır. Dikkat skorları SoftMax fonksiyonundan geçirildikten sonra V ile çarpılarak öz dikkat değerleri (2.23) numaralı denklem ile hesaplanmaktadır.

$$\text{öz dikkat} = \text{SoftMax}\left(\frac{Q \cdot K^T}{\sqrt{d_k}}\right) \cdot V \quad (2.23)$$

Dönüştürücü ağlarda çok başlıklı öz dikkat yapısı kullanılmaktadır. Her başlık kendine ait ağırlık vektörlerine sahiptir. Tüm başlıklardan elde edilen öz dikkat değerleri birleştirilerek nihai sonuç üretilmektedir.

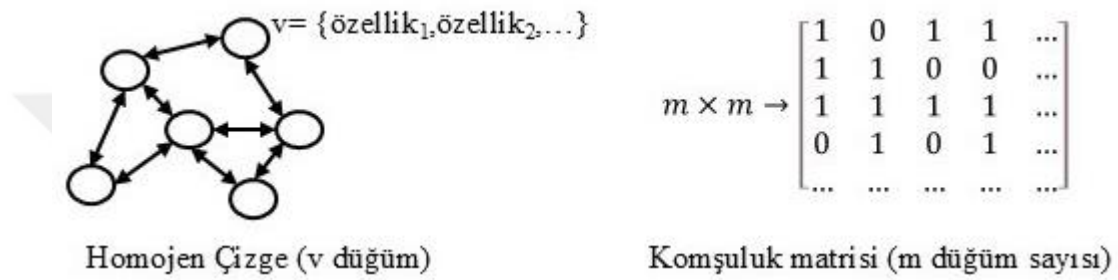
Bu tez çalışması kapsamında, nesne algılama ve takip aşamaları sonucunda zaman serisi verisi olarak elde edilen nesnelere ait geçmiş hareket bilgilerinden nesnenin hareket karakteristiğini öğrenmek için Conv1D, LSTM ve GRU kullanılmış ve bu modellerin yönteme katkısı incelenmiştir. Dönüştürücü ağ ise nesnelerin özellik vektörlerindeki verilerin ağırlıklandırılarak rota tahminine etkilerinin belirlenmesi amacıyla kullanılmıştır.

2.7 ÇİZGE İLE VERİ TEMSİLİ

Trafik etkileşimli bir ortam olduğu için nesnelerin bireysel özelliklerinin yanında birbirleriyle etkileşiminin de dikkate alınması gerekmektedir. Bu nedenle yeni nesil çalışmalarda trafik verisi çizge yapıları ile ele alınmaktadır [84-86, 90, 93]. Çizgeler Öklid mesafesiyle doğrudan anlamlandırılmayan düzensiz verilerin temsiliyi sağlayan bir veri yapısıdır. Çizgelerde, düğümler (V) ve düğümler arasındaki bağlantıyı ifade eden kenarlar (E) bulunmaktadır ve genel olarak bir çizge $G = \{V, E\}$ şeklinde ifade edilmektedir. Kenarlar yönlü ve yönsüz olmak üzere iki

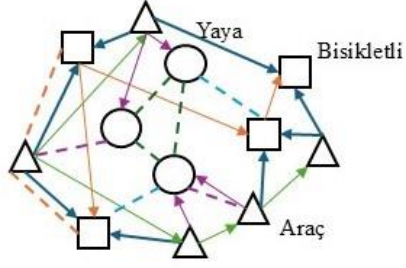
tipde tanımlanabilmektedir. Yönsüz kenarlarda iki düğüm arasında eş bir bağlantı, yönlü kenarlarda ise iki düğüm arasında sadece belirtilen yönde bir bağlantı bulunmaktadır.

Çizgeler düğüm ve kenar tiplerinin çeşitliliğine göre homojen ve heterojen olmak üzere iki gruba ayrılmaktadır [120]. Homojen çizgede tek bir tip düğüm ve kenar bulunmaktadır. Düğümlerin ve kenarların özellikleri kendi aralarında özdeş vektörler ile, düğümler arasındaki ilişki ise tek bir komşuluk matrisi ile temsil edilmektedir. Şekil 2.7’de homojen bir çizge ve bu çizgenin komşuluk matrisi gösterilmektedir. Homojen çizgeler basit veri temsili sayesinde düşük işlem maliyetiyle düğümler arasındaki etkileşimin modellenmesine olanak sağlamaktadır [121].



Şekil 2-7: Homojen çizge ve çizgenin komşuluk matrisi ile temsili

Trafik ortamında birbirinden farklı karakteristiği sahip nesneler olduğu için birbirleriyle etkileşimleri de farklı özellikte olmaktadır. Bu nedenle trafik ortamını en iyi ifade etme yöntemi heterojen çizgelerdir [63]. Heterojen çizge yapısında birbirinden farklı özelliklere sahip düğümler ve kenarlar bulunmaktadır. G çizge, V düğüm, E kenar, V_t düğüm tipi, E_t kenar tipi olmak üzere heterojen çizge, $G = \{V, E, V_t, E_t\}$ şeklinde ifade edilmektedir. Şekil 2.8’de görülebileceği gibi heterojen çizgelerle birden fazla düğüm ve kenar türü ele alınabilmektedir. Böylece daha detaylı bir temsil elde edilebilmektedir [120]. Her düğüm ve kenar türünün kendine özgü özellik vektörü yapıları bulunabilmektedir. Düğümler arasındaki etkileşim farklı komşuluk matrisleriyle ifade edilebilmektedir [120]. Heterojen çizgeler kullanılırken, heterojen çizgelerin detaylı temsil yeteneğinin karmaşıklığı arttırmasından dolayı hesaplama maliyetini arttırdığını göz önünde bulundurmak gerekmektedir [121].



Heterojen Çizge

$$\begin{aligned}
 a \times a &\rightarrow \begin{bmatrix} 1 & 0 & 1 & \dots \\ 0 & 1 & 1 & \dots \\ \dots & \dots & \dots & \dots \end{bmatrix} & y \times y &\rightarrow \begin{bmatrix} 1 & 0 & 1 & \dots \\ 0 & 1 & 1 & \dots \\ \dots & \dots & \dots & \dots \end{bmatrix} & b \times b &\rightarrow \begin{bmatrix} 1 & 0 & 1 & \dots \\ 0 & 1 & 1 & \dots \\ \dots & \dots & \dots & \dots \end{bmatrix} \\
 a \times y &\rightarrow \begin{bmatrix} 1 & 0 & 1 & \dots \\ 0 & 1 & 1 & \dots \\ \dots & \dots & \dots & \dots \end{bmatrix} & a \times b &\rightarrow \begin{bmatrix} 1 & 0 & 1 & \dots \\ 0 & 1 & 1 & \dots \\ \dots & \dots & \dots & \dots \end{bmatrix} & y \times b &\rightarrow \begin{bmatrix} 1 & 0 & 1 & \dots \\ 0 & 1 & 1 & \dots \\ \dots & \dots & \dots & \dots \end{bmatrix}
 \end{aligned}$$

Heterojen Çizge Komşuluk Matrisi. a araç sayısı, b bisikletli sayısı, y yaya sayısı

Şekil 2-8: Heterojen çizge ve çizgenin komşuluk matrisi ile temsili

Bu tez çalışması kapsamında trafik ortamındaki hareketli nesneler ve bu nesnelerin birbiriyle etkileşimi heterojen yönlü çizgelerle ifade edilmiştir. Böylece tüm nesne türleri arasındaki etkileşim detaylı bir şekilde ele alınabilmektedir.

2.8 DERİN GRAFİK ÖĞRENİMİ YÖNTEMLERİYLE ÇİZGELER İÇİN DÜĞÜM GÖMMELERİNİN HESAPLANMASI

Derin grafik öğrenimi, çizge olarak ifade edilmiş düzensiz yapıya sahip verileri işleyerek verinin öz niteliklerini bulmaya odaklanmış derin öğrenme tekniğidir. Derin grafik öğrenme yönteminde düğüm ve kenar özellikleri kullanılarak çizgeyi daha iyi analiz etmek için düğümler arasındaki ilişkileri de temsil eden gömmeler elde edilmektedir. Gömmeler, düğümlerin ve kenarların çizgedeki topolojik ilişkilerini ve bağlamsal bilgilerini koruyarak komşuları üzerinden elde edilen bilgiler aracılığıyla yeni bir veri temsili oluşturmaktadır [122]. Derin grafik öğrenimi sosyal ağ analizi, biyoinformatik, kimyasal analiz, finansal analiz, trafik ve ulaşım sistemlerinin analizi ve kontrolü, doğal dil işleme gibi alanlarda sıklıkla kullanılmaktadır [123].

Gerçek hayat verisine bakıldığında heterojen veri yapısıyla karşılaşılmaktadır. Heterojen çizgeleri işlemek için temel çizge komşuluk matrisleri kullanılarak alt homojen çizgelere bölünmekte ve her bir alt çizgeden elde edilen bilgiler birleştirilerek gömmelerin son hali oluşturulmaktadır [120].

Derin grafik öğreniminde düğümlerin komşu sayısının sabit olmaması, verinin sırasının öğrenmede anlamlı olmaması, büyük çizgelerde hangi düğümlerin ele alınması gerektiğine karar verilmesi, düğümün komşularıyla birlikte anlamlı olması nedeniyle komşulardan bilgi alınması gerekliliği ve kaçınıcı seviye komşuya kadar ulaşıp bilgi alınması gerektiğine karar verilmesi gibi aşılması gereken problemler bulunmaktadır [124]. Bu amaçla geliştirilen modellerden ikisi Çizge

Evrişimli Ağ ve Çizge Dikkat Ağıdır. Tek katman olarak kullanılan ağlarda sadece düğümün kendi komşularından bilgi elde edilebilirken çok katmanlı yapılarda komşuların komşularından da bilgi aktarımı yapılarak daha geniş komşuluk çevresinden bilgilerin elde edilmesi sağlanabilmektedir.

2.8.1 Çizge Evrişimli Ağ (Graph Convolutional Network -GCN)

Kipf ve Welling [67] tarafından geliştirilen GCN, düğüm özellikleri ile düğümün komşularından gelen bilgileri birleştirip düğüm gömmeleri hesaplamaktadır. GCN, tüm komşu düğümleri eşit olarak dikkate almaktadır. A komşuluk matrisi, D düğümlerin derecesini ifade eden diyagonal matris, I birim matrisi, X düğümlerin özellik matrisi ve W ağırlık matrisi ve σ tercih edilen aktivasyon fonksiyonu olmak üzere GCN ile düğüm gömmeleri (2.24) numaralı denklem ile hesaplanmaktadır.

$$f(X, A) = \sigma(D^{-\frac{1}{2}}(A + I)D^{-\frac{1}{2}}XW) \quad (2.24)$$

(2.24) numaralı denklemde komşuluk matrisine birim matris eklenerek her düğümün kendisine bir bağlantıya sahip olması sağlanmakta böylece hesaplamalar yapılırken düğümün kendi özelliğinin de dahil edilmesi sağlanmaktadır. Daha sonra karekökü alınan derece matrisleriyle komşuluk matrisinin yeni hali çarpılarak komşuluk matrisine normalizasyon işlemi uygulanmaktadır. Böylece düğümlerin sonuçlardaki etkinliği elde edilmektedir. Elde edilen sonuç düğümlerin özellik matrisleriyle çarpılıp Mesaj Geçişi (Message Passing) ile komşulardan bilgilerin alınması sağlanmaktadır. Son olarak öğrenilebilir ağırlık matrisiyle çarpılarak güncellenmiş düğüm gömmeleri elde edilmektedir.

2.8.2 Çizge Dikkat Ağı (Graph Attention Network -GAT)

Veličković vd. [77] tarafından geliştirilen GAT, düğüm gömmelerini oluştururken dikkat mekanizmasını kullanan GCN'nin daha gelişmiş bir versiyonudur. GAT, her düğüm için komşu düğümlerin katkısını öğrenirken, düğüm çiftleri arasındaki ilişkinin önemini belirleyen bir dikkat değeri kullanmaktadır.

GAT yapısında iki düğüm arasındaki dikkat skoru e_{ij} ; *SoftMax* normalizasyon fonksiyonu, a dikkat mekanizmasının ağırlık vektörü, W öğrenilebilir ağırlık matrisi, h_i ve h_j iki komşu

düğümün özellik vektörleri ve σ tercih edilen aktivasyon fonksiyonu olmak üzere (2.25) numaralı denklem ile hesaplanmaktadır.

$$e_{ij} = \text{SoftMax}(\sigma(a^T [Wh_i, Wh_j])) \quad (2.25)$$

Dikkat skorları hesaplandıktan sonra i düğümünün gömmesi g_i ; N ilgili düğümün komşuları olmak üzere (2.26) numaralı denklem ile hesaplanmaktadır.

$$g_i = \sigma(\sum_{j \in N(i)} e_{ij} Wh_j) \quad (2.26)$$

Dikkat mekanizması çok başlıklı olarak da kullanılabilir. Bu durumda belirlenen adet kadar başlıktan dönen bilgiler birleştirilerek nihai gömme değerleri oluşturulmaktadır.

Bu tez çalışması kapsamında heterojen çizgelerin düğüm gömmelerini elde etmek için Çizge Evrişimli Ağ ve Çizge Dikkat Ağı ile elde edilen sonuçlar birleştirilerek kullanılmıştır. Böylece iki yöntemin avantajlarından yararlanılmıştır.

2.9 ORTALAMA KARE HATASI KAYIP FONKSİYONU

Kayıp fonksiyonu, yapay zekâ modellerinin eğitimi esnasında modelin ürettiği sonuç ile gerçek sonuç arasındaki farkın hesaplanmasında kullanılan fonksiyondur. Bu fonksiyon aracılığıyla hesaplanan değerler kullanılarak modelin parametrelerinin güncellenmesi sağlanmaktadır. Optimizasyon açısından bakıldığında kayıp fonksiyonu optimize edilmesi gereken amaç fonksiyonu olarak ele alınmaktadır.

Hangi kayıp fonksiyonunun kullanılacağına problemin yapısına bağlı olarak karar verilmektedir. Rota tahmin uygulamalarında konuma bağlı olarak Öklid uzaklığı tabanlı hata hesabı yapıldığı için genel olarak Ortalama Kare Hatası (MSE) fonksiyonu kullanılmaktadır. MSE kayıp fonksiyonunda modelin oluşturduğu tüm hataların kareleri alınıp ortalaması alındığı için büyük hatalar ön plana çıkarılarak modelin ciddi sapmalarının düzeltilmesini sağlanmaktadır.

N örnek sayısı, y gerçek değer, $\hat{y}$ modelin tahmin ettiği değer olmak üzere MSE (2.27) numaralı denklem ile hesaplanmaktadır:

$$MSE = \frac{1}{N} \sum_{i=1}^N (y - \hat{y})^2 \quad (2.27)$$

Bu tez çalışması kapsamında rota tahmin modelinin eğitilmesi aşamasında, modelin hatasını hesaplamak için tahmin edilen rotanın koordinatları ile gerçek rotanın koordinatları arasındaki Öklid mesafesi kullanılmaktadır. Bu nedenle kayıp fonksiyonu olarak MSE kullanılmıştır.

2.10 ADAM (ADAPTIVE MOMENT ESTIMATION) OPTİMİZASYON YÖNTEMİ

Optimizasyon, bir problemi tanımlayan matematiksel model içinde, belirli bir amaç fonksiyonunu en iyi şekilde (maksimum veya minimum değerine ulaşarak) çözmeyi hedefleyen yöntemlerin genel adıdır [125]. Derin öğrenme açısından optimizasyonun görevi, yapay zekâ modelinde kullanılan parametrelerin değerlerinin en ideal duruma getirilmesi için parametrelerde yapılması gereken değişikliklerin (artış/azalış) hesaplanmasıdır.

Derin öğrenme modellerinde sıklıkla kullanılan optimizasyon yöntemlerinden biri ADAM optimizasyon yöntemidir [126]. ADAM yönteminde her bir parametre için gradyanlara bağlı olarak iki farklı momentum değeri hesaplanmaktadır. Elde edilen bu değerler ile öğrenme katsayısının hem gradyanın büyüklüğüne hem de gradyanın yönündeki değişimlere göre ölçeklenmesine olanak sağlanmaktadır.

ADAM algoritmasında öncelikle modeli eğitmek için tercih edilen kayıp fonksiyonuna bağlı olarak her bir parametre için kısmi türev aracılığıyla gradyan vektörü (g_t); t zaman dilimi için f kayıp fonksiyonu w güncellenecek model parametreleri olmak üzere (2.28) numaralı denklem ile hesaplanır:

$$g_t = \nabla w f_t(w_{t-1}) \quad (2.28)$$

Hiper parametre olarak belirlenen β_1 ve β_2 katsayı değerleri ve gradyanların karesi (g_t^2) kullanılarak birinci (m) ve ikinci momentum (v) değerleri düzeltilmiş olarak sırasıyla (2.29) ve (2.30) numaralı denklemler ile hesaplanır. Birinci ve ikinci momentum değerleri ilk zaman dilimi için 0 kabul edilir.

$$m_t = (\beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot g_t) \div (1 - \beta_1) \quad (2.29)$$

$$v_t = (\beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot g_t^2) \div (1 - \beta_2) \quad (2.30)$$

Son aşama olarak hesaplanan düzeltilmiş momentum değerleri kullanılarak yapay zekâ modelinin parametrelerinin son hali, α öğrenme katsayısı ve ε sıfıra bölünme hatasını önlemek amacıyla kullanılan küçük sayıyı ifade etmek üzere (2.31) numaralı denklem ile güncellenmektedir:

$$w_t = w_{t-1} - \alpha. (m_t / (\sqrt{v_t} + \varepsilon)) \quad (2.31)$$

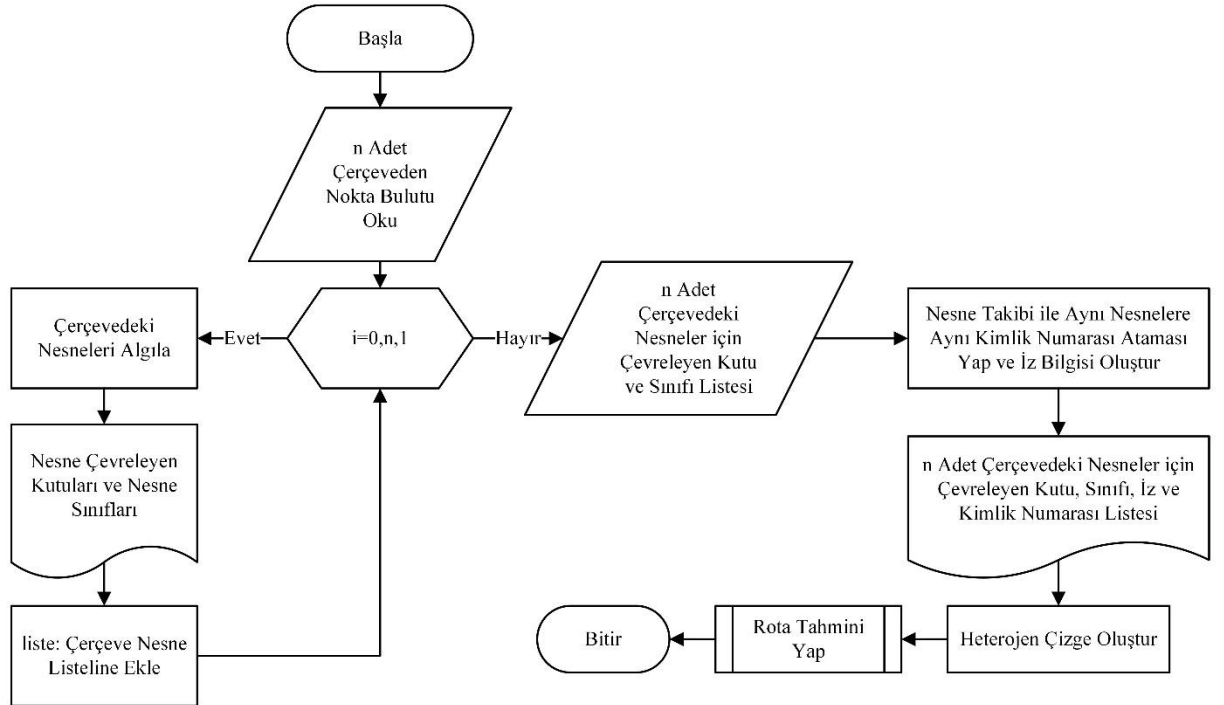
Bu tez çalışması kapsamında geliştirilen rota tahmin modelinin eğitilmesi aşamasında parametrelerinin güncellenmesi için ADAM optimizasyon yöntemi kullanılmıştır.



3. TRAFİKTEKİ HAREKETLİ NESNELER İÇİN UÇTAN-UCA ROTA TAHMİNİ

Tam otonom araçların trafikte güvenli bir şekilde seyahat edebilmesi için bulundukları ortamdaki nesneleri algılayıp bu nesnelerin gelecekteki rotasını tahmin etmesi gerekmektedir. Bu tez çalışması kapsamında, LiDAR sensöründen elde edilen bilgileri kullanarak trafikteki hareketli nesneleri algılayıp nokta bulutu çerçeveleri arasında takip ettikten sonra bu nesnelerin gelecekteki rotasını tahmin eden, derin öğrenme tabanlı Uçtan-Uca Rota adında yeni bir yöntem geliştirilmiştir. Şekil 3.1’de de görüleceği üzere bu yöntemin temel beş aşaması bulunmaktadır:

- 1- LiDAR verisinin okunup işlenecek hale getirilmesi
- 2- Aracın etrafındaki nesnelerin algılanması
- 3- Algılanan nesnelerin LiDAR çerçeveleri arasında takip edilmesi
- 4- Heterojen çizge ile nesnelerin özelliklerinin ve birbiriyle etkileşiminin temsili
- 5- Nesneler için rota tahmininin eş zamanlı olarak gerçekleştirilmesi



Şekil 3-1: Tez çalışmasında geliştirilen Uçtan-Uca Rota yönteminin akış diyagramı

Uçtan-Uca Rota yönteminde ilk iki aşama açık kullanım hakları ve kaynak kodlarına sahip CenterPoint, üçüncü aşama ise açık kullanım hakları ve kaynak kodlarına sahip MCTrack modeli

kullanılarak gerçekleştirilmiştir. Her iki modelde de yazarlar tarafından sunulan hazır eğitilmiş modeller kullanılmış ve modeller üzerinde herhangi bir aktarım öğrenme (transfer learning) veya uyumlama (fine-tuning) tekniği kullanılmamıştır. Modellerin veri seti üzerinde yapılmasına ihtiyaç duyduğu veri ön işleme aşamaları, yazarlar tarafından sunulan açık kaynak kodlar kullanılarak veri setinin eğitim ve doğrulama dosyaları üzerinde gerçekleştirilmiştir.

Dördüncü aşama için tez çalışması kapsamında geliştirilen yazılım ile heterojen çizgeler oluşturulmuştur.

Beşinci aşama için tez çalışması kapsamında tasarlanan ve eğitilen ROTA adında yeni bir model geliştirilip kullanılmıştır. Model öncelikle veri setindeki etiket bilgileri kullanılarak hata oranı az olan veriler ile eğitilmiştir. Geliştirilen Uçtan-Uca Rota yönteminde kullanılan veri, veri setindeki etiket bilgileriyle tam örtüşmediği için rota tahmin modelinin yeni veri karakteristiğini de tanınması amacıyla uyumlama tekniği kullanılarak eğitime devam edilmiştir. Böylece algılama ve takip aşamalarından kaynaklanacak hataların azaltılması sağlanmıştır.

Tez çalışmasının bu bölümünde; kullanılan veri setlerinin özelliklerinden, kullanılan hazır algılama ve takip yapay zekâ modellerinden, heterojen çizgi oluşturma aşamalarından ve tez kapsamında geliştirilen rota takip modelinin aşamalarından bahsedilmiştir.

3.1 VERİ SETLERİ

Son dönemde, otonom araçların çevreyi algılaması ve çevresel koşullara göre hareket karakteristiğini belirleyebilmesi amacıyla kullanılabilecek gelişmiş veri setleri yayınlanmıştır. Veri setlerinin içeriğinde kamera sensöründen elde edilen resim ve videolar bulunduğu gibi LiDAR sensöründen elde edilen 3B nokta bulutu bilgileri de bulunmaktadır. Sensörlerden elde edilen her bir çerçeve bazında yaya, araba, bisikletli gibi trafik ortamında bulunabilecek nesnelerin sınıfı, kimlik numarası, çevreleyen kutusu, hızı ve ivmelenmesi gibi etiket bilgileri; çerçeve bazında dönüşüm ve kalibrasyon matrisleri, içinde bulunulan ortamın ışık ve hava durumu gibi bilgiler bulunabilmektedir. Bu tez kapsamında nesne algılama, takip ve rota tahmin işlemleri birlikte yapılmaktadır. Bu nedenle tezde Tablo 3.1’de de özeti bulunan veri setlerinden yeni nesil çalışmalar tarafından da sıklıkla tercih edilen ve detaylı veri sunan WAYMO Perception (bu aşamadan sonra WAYMO olarak ifade edilecektir) [37] ile Argoverse 2 Sensor (bu aşamadan sonra Argoverse olarak ifade edilecektir) [96] veri setleri kullanılmıştır.

Tablo 3-1: Otonom araçlar için nesne algılama, takip ve rota tahmini alanında yapılacak çalışmalarda kullanılabilir veri setleri ve özellikleri

Veri Seti	İçerik	Veriler	Etiketlerdeki Bilgiler
WAYMO Perception	Eğitim için 773, doğrulama için 202, test için 150 senaryo	Kamera, LiDAR, Harita	Nesne sınıfı, kimlik numarası, 2B çevreleyen kutu, 3B çevreleyen kutu, hız, ivmelenme ve doğrultu açısı
Argoverse 2 Sensor	Eğitim için 700, doğrulama ve test için 150'şer senaryo	Kamera, LiDAR, Araç Hız ve hareket, Harita	Nesne sınıfı, kimlik numarası, 3B çevreleyen kutu, rotasyon açısı
ApolloScape Rota	53 dakika eğitim, 50 dakika test verisi	Kamera, LiDAR	Nesne sınıfı, kimlik numarası, dünya koordinatında 3B çevreleyen kutu, doğrultu açısı
nuScene	Eğitim için 700, doğrulama ve test için 150'şer senaryo	Kamera, LiDAR, IMU, GPS, Harita	Nesne sınıfı, kimlik numarası, 3B çevreleyen kutu
WAYMO Motion	Eğitim için 1000, doğrulama ve test için 150	LiDAR, Rota, Harita	Nesne sınıfı, kimlik numarası, 3B çevreleyen kutu, hız, ivme ve doğrultu açısı
Argoverse Motion	Eğitim için 205942, doğrulama için 39472, test için 78143 iz örneği	Rota, Harita	Nesne sınıfı, kimlik numarası, 3B çevreleyen kutu, hız ve doğrultu açısı
KITTI	Eğitim ve doğrulama için 7481 görüntü, test için 7518 görüntü (etiketsiz)	Stereo Kamera, LiDAR, GPS/IMU	Nesne sınıfı, kimlik numarası, 3B çevreleyen kutu, rotasyon açısı

Rota tahmin modelinin ilk geliştirilmesi aşamasında veri seti olarak, kaydolunması durumunda dosyalara erişim hakkına sahip olunabilen, WAYMO tercih edilmiş olup 1.4.2 versiyonundaki arşiv dosyaları kullanılmıştır. WAYMO veri seti, otonom sürüş araştırmalarında kullanılmak üzere yüksek kaliteli gerçek dünya verileri sağlamaktadır.

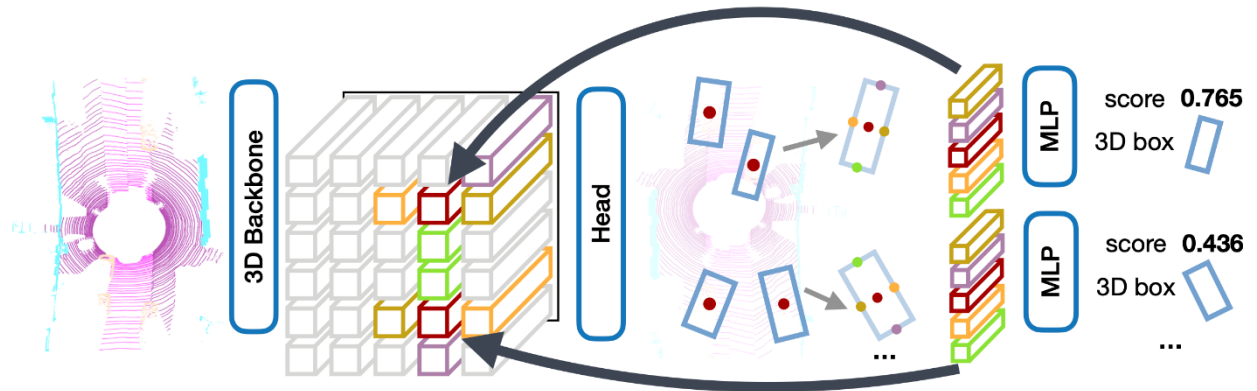
Veri setinin kullanılan arşiv dosyasında 773 eğitim, 202 doğrulama ve 150 test dosyası bulunmaktadır. Her dosya 10 Hertz frekansında 20 sn uzunluğunda olup kamera ve lidar sensörü ile elde edilen verilere sahiptir. Dosyalarda yaklaşık 199 çerçeve bulunmakta ve bu çerçevelerdeki nesneler için kamera ve lidar sensörleriyle elde edilen veriler ve nesnelere ait etiketler birbirinden bağımsız olarak bulunmaktadır. Nesneler, kimlik numarası verilirken 2B ve 3B ortamında ayrı numaralarla etiketlenmiş ve birbiriyle ilişkilendirilmemiştir. 3B ortamında nesneler etiketlenirken *3B çevreleyen kutu özellikleri, nesne sınıfı, kimlik numarası*, x ve y ekseninde *hız*, x ve y ekseninde *ivmelenme* ve *doğrultu* bilgisi kullanılmaktadır. 3B kutunun koordinat ve doğrultu bilgileri aracın koordinat düzleminde. Her bir çerçeve için kamera ve LiDAR kalibrasyon matrisleri ve koordinat düzlemleri arasında çevirim işlemlerini yapabilmek için dönüşüm matrisleri

bulunmaktadır. Eğitim ve doğrulama setlerinde çok geniş etiket bilgileri sunulurken, test seti değerlendirme için ayrıldığından etiket bilgileri bulundurmamaktadır.

Modelin farklı veri setleriyle uyumlu çalışmasının araştırılması için ikinci veri seti olarak Argoverse veri seti kullanılmıştır. Veri setinde kamera ve LiDAR sensörleriyle elde edilen eğitim için 700, doğrulama ve test için 150'şer senaryo bulunmaktadır. Her senaryo 15 sn uzunluğunda ve 10 Hertz frekansındadır. Senaryodaki her bir sensör için 0,1 saniyede bir kayıt yapılarak yaklaşık 157 dosya oluşturulmuş; etiket ve otonom araca ait bilgiler 15 sn zaman aralığını kapsayacak şekilde ayrı dosyalar olarak kaydedilmiştir. Nesnelerin 3B etiketleri bulunmaktadır. Bu etiketlerde *3B çevreleyen kutu*, *nesne sınıfı*, *kimlik numarası* ve *doğrultu* bilgisi bulunmaktadır. Hız ve ivmelenme bilgileri hazır bulunmadığı için bu bilgiler aynı nesnenin birbirini takip eden iki zaman dilimindeki konum ve hız değişikliği bilgileri dikkate alınarak hesaplanmıştır. Her bir çerçeve için kamera ve LiDAR kalibrasyon matrisleri ve koordinat düzlemleri arasında çevirim işlemlerini yapabilmek için dönüşüm matrisleri bulunmaktadır. WAYMO veri setinde olduğu gibi eğitim ve doğrulama setlerinde etiket bilgileri sunulurken, test seti değerlendirme için ayrılmış olup etiket bilgileri bulunmamaktadır.

3.2 LiDAR VERİSİNİN İŞLENMESİ ve NESNE ALGILAMA: CenterPoint

Bu tez çalışması kapsamında LiDAR verisinin okunması ve nesne algılama işlemleri için CenterPoint [33] algoritması kullanılmıştır. CenterPoint algoritması, 3B LiDAR verileri üzerinde nesne algılama işlemi gerçekleştirebilmekte ve pek çok çalışma tarafından omurga model olarak kullanılmaktadır [53-54, 58].

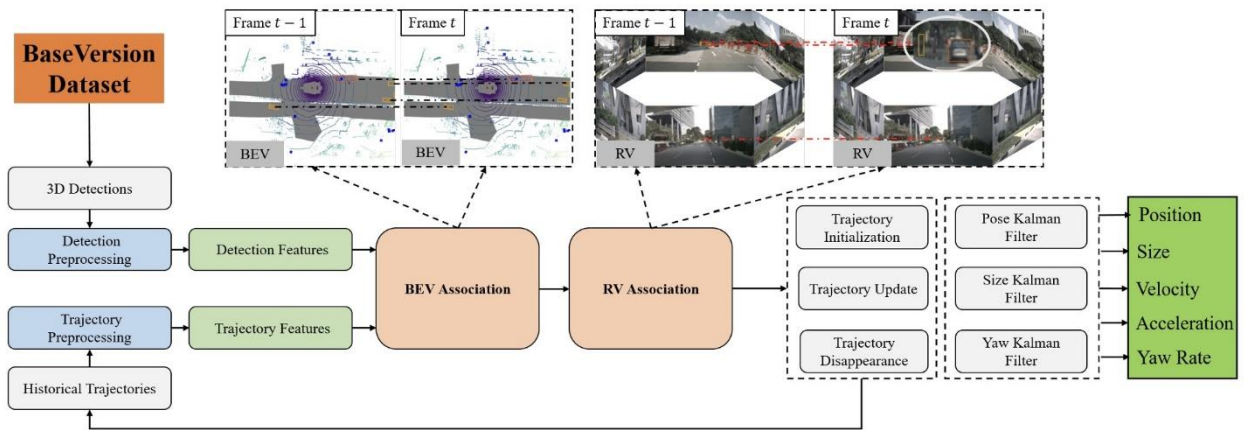


Şekil 3-2: CenterPoint algoritmasının yapısı [33]

Şekil 3.2’de akış diyagramı gösterilen ve iki aşamalı olarak nesne algılama işlemi gerçekleştiren CenterPoint çalışmasında, LiDAR verilerinin önışlemesinde omurga ağ olarak VoxelNet kullanılan model tercih edilmiştir. CenterPoint algoritmasında öncelikle VoxelNet ile verinin özellik haritaları elde edilmektedir. Elde edilen bu özellik haritalarından CenterNet algoritmasına dayalı olarak nesnenin merkez noktasında zirve yapan ısı yoğunluk haritası ve düzeltme vektörü nesne bazlı oluşturulmaktadır. Veri setinde yayınlanmış çevreleyen kutu bilgilerinden oluşturulan 2B Gauss dağılımı kullanılarak yoğunluk haritaları üzerinden nesne algılaması gerçekleştirilmektedir. Böylece her nesne için nesne sınıfı, nesne boyutu, alt-voxel konum iyileştirmesi, dönüş açısı ve hız bilgileri tahmin edilmektedir. Bu işlemlerden sonra, omurga ağdan elde edilen veriler üzerinden İleri Beslemeli Ağ ile yeni özellikler bulunarak kutuların iyileştirilme değerleri ve güven skorları elde edilip nihai çevreleyen kutular hesaplanmaktadır.

3.3 NESNE TAKİP: MCTrack

Bu tez çalışması kapsamında video kareleri arasında nesneleri takip edip nesnelere kimlik numarası atamak için Şekil 3.3’te akış diyagramı görünen MCTrack [61] algoritması kullanılmıştır.



Şekil 3-3: MCTrack algoritmasının yapısı [61]

MCTrack, 3B çoklu nesne takibini gerçekleştirmek için önce farklı veri setleri kullanılarak elde edilmiş olabilecek algılama sonuçlarının ortak bir formata dönüştürülmesini beklemekte ve verileri küresel koordinat sisteminde işlemektedir. Takip süreci için nesnelerin pozisyonu, boyutu

ve yönünü ayrı ayrı modelleyen üç farklı Kalman filtresi bulunmaktadır, böylece farklı özelliğe sahip verilerden daha detaylı analiz yapılması hedeflenmiştir. İlgili zaman diliminde algılanan nesnelerle önceki kareden elde edilen izlerin eşleştirilmesi, öncelikle kuş bakışı görünümünde Macar Algoritması ile gerçekleştirilmektedir. Bu aşamada benzerlik metriğinin hesaplanmasında hem kutu örtüşmesi hem de yönelme farkı dikkate alınmıştır. Eşleştirilemeyen izler ve algılama sonuçları için ikinci aşamada kamera görüntüsüne yansıtılan LiDAR verisinde mesafe görünümü elde edilmekte ve Aç Gözlü Algoritması ile tekrar eşleştirme yapılmaktadır. Elde edilen eşleştirmeler Kalman filtrelerine gönderilerek çıkan sonuçlar ile rotalar güncellenmekte ve bir sonraki zaman diliminde eşleştirmenin yapılması için iz tahminleri yapılmaktadır.

3.4 HETEROJEN ÇİZGE OLUŞTURMA

Bu tez çalışmasında rota tahmin aşamasında dinamik heterojen yönlü çizgeler girdi olarak kullanılmaktadır. Bu amaçla oluşturulan heterojen yönlü çizgede trafik nesneleri *yaya*, *araç* ve *bisikletli* tipinde üç farklı düğüm ile ifade edilmiştir. Gözlem süresi boyunca elde edilen bilgilerden *id* nesnenin kimlik numarası, *s* nesnenin sınıfı, *x* ve *y* nesnenin global koordinat düzlemindeki koordinat bilgisi, v_x ve v_y sırasıyla *x* ve *y* eksenini doğrultusundaki hız, a_x ve a_y sırasıyla *x* ve *y* eksenini doğrultusundaki ivmelenme ve *h* doğrultu olmak üzere ilgili *t* zaman diliminde *n* nesnesi için düğüm özellik vektörü 3.1 numaralı denklem ile ifade edilmiştir. Üç nesne sınıfı için de aynı özellik vektörü yapısı kullanılmıştır.

$$f_n^t = \{id_n^t, s_n^t, x_n^t, y_n^t, v_{x_n}^t, v_{y_n}^t, h_n^t, a_{x_n}^t, a_{y_n}^t\} \quad (3.1)$$

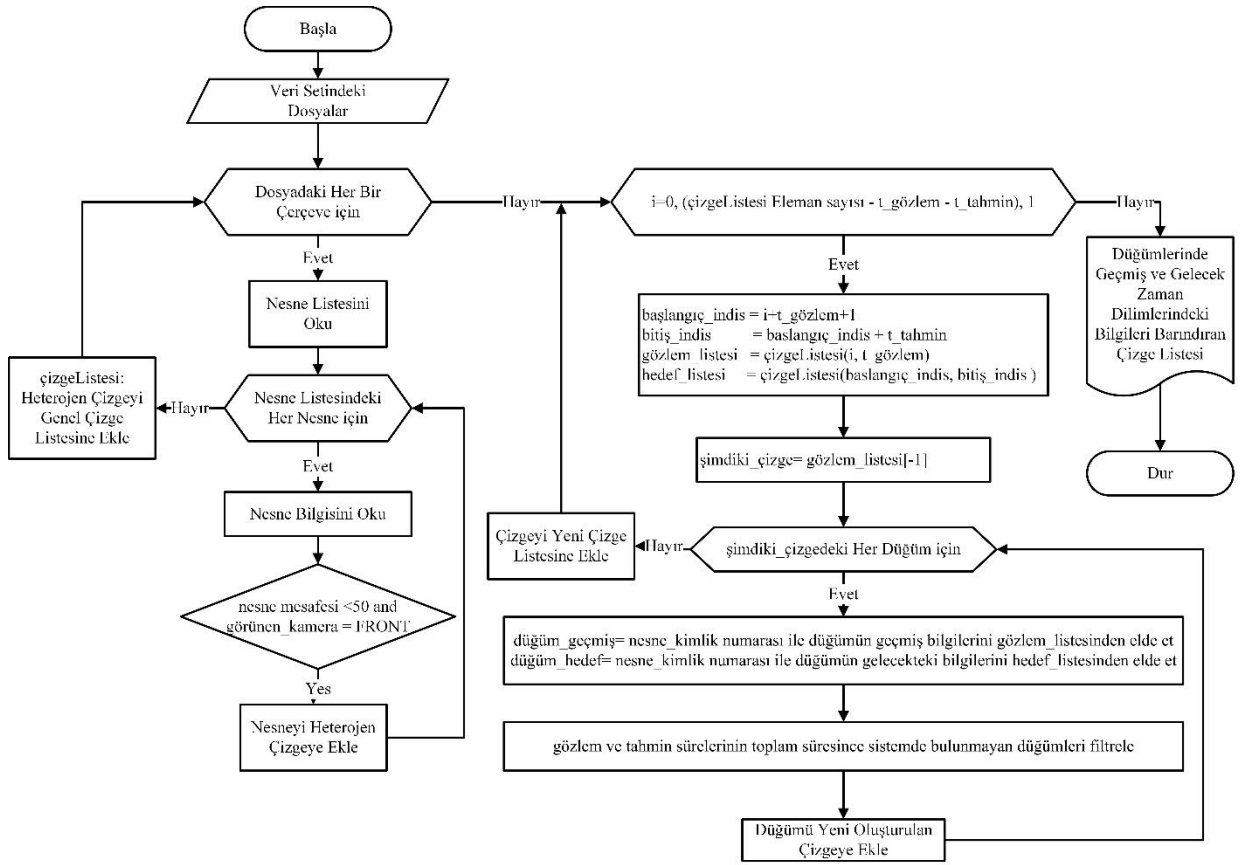
Nesneler aralarındaki etkileşim *araç-etkiler-araç*, *araç-etkiler-yaya*, *araç-etkiler-bisikletli*, *yaya-etkiler-yaya*, *yaya-etkiler-bisikletli*, *yaya-etkiler-araç*, *bisikletli-etkiler-bisikletli*, *bisikletli-etkiler-yaya* ve *bisikletli-etkiler-araç* tipinde dokuz ayrı yönlü kenar türü ile ifade edilmiştir. Böylece nesnelerin, diğer nesne türlerini ne kadar etkileyeceği ve diğer nesne türlerinden ne kadar etkileneceği detaylı bir şekilde ayrı ayrı takip edilebilmektedir.

Rota tahmin aşaması için geliştirilen ROTA modelinin eğitiminde, gözlem ve tahmin için oluşturulan heterojen çizgelere ait düğümlerin geçmiş ve gelecek hareket bilgilerinin elde edilmesi için WAYMO veri setindeki hazır etiketler kullanılmıştır. Veri setindeki her bir dosyadaki çerçeveler için LiDAR verisi olarak sunulan *nesne kimlik numarası*, 3B çevreleyen kutunun merkez noktasının *x* ve *y* koordinatı, *nesne sınıfı*, *x* ve *y* doğrultularındaki hız (*m/s*), *doğrultu*, *x* ve

y doğrultusundaki *ivmelenme* (m/s^2) bilgileri okunmuştur. x , y koordinatı ve *doğrultu* bilgileri araç koordinat düzleminde olduğu için dönüşüm matrisiyle global koordinat sistemine dönüştürülmüştür. Hız ve ivmelenme verileri global düzlemde sunulmaktadır. Nesnelerden otonom araca 50 m mesafeden yakın olanlar dikkate alınarak her bir çerçeve için heterojen çizge oluşturulmuştur. Bu aşamada iki farklı veri oluşturulmuş olup birinci deney setinde sadece ön kameranın görüş açısında olan nesneler, ikincisinde ise LiDAR ile algılanan tüm nesneler dikkate alınmıştır. Böylece veri setindeki her bir dosya için 199 adet heterojen çizge elde edilmiştir.

Argoverse veri setinde de aynı yöntemler izlenerek bilgiler elde edilmiştir. Fakat Argoverse veri setinde hız ve ivmelenme değerleri hazır olarak bulunmadığı için çevreleyen kutuların orta nokta koordinat değerlerinin çerçeveler arasındaki değişimi üzerinden x ve y doğrultusundaki hızları hesaplanmıştır. Çerçeveler arasındaki hız bilgilerinin farkı kullanılarak da x ve y doğrultusundaki ivmelenme değerleri hesaplanmıştır. Argoverse veri seti kullanıldığı durumda senaryoların uzunluğu 15 sn olduğu için her bir dosyadan 150'şer adet heterojen çizge elde edilmiştir.

Elde edilen bu çizgeler model tarafından kullanılmak üzere bir araya getirilerek nihai çizgeler elde edilmiştir. Bu aşamada gözlem süresi olarak -çalışmalarda sıklıkla tercih edilen- 1 sn (10 çerçeve) ve 2 sn (20 çerçeve) olmak üzere iki farklı süre; tahmin süresi olarak ise 1, 2, 3, 4, 5, 8 sn olmak üzere 6 farklı süre kullanılmıştır. Tercih edilen gözlem ve tahmin sürelerine göre bu çizgelerden *gözlem süresi* $\times$ 10 tanesi geçmiş, *tahmin süresi* $\times$ 10 tanesi etiket çizgesi olarak ele alınmıştır. Gözlem listesindeki son çizge baz alınarak, aynı kimlik numarasına sahip düğümler için geçmiş çizgelerden bilgiler bir araya getirilerek düğümün geçmiş özellik vektörü ve etiket çizgelerinden sadece konum bilgisi alınarak düğümün etiket vektörü oluşturulmuş ve sonuçlar yeni oluşturulan heterojen çizgeye eklenmiştir. Her defasında çerçeveler arasında bir adım kaydırma yapılarak yeni çizge oluşturma işlemleri tekrarlanmıştır. Heterojen çizge oluşturma algoritmasının akış diyagramı Şekil 3.4'te gösterilmektedir.

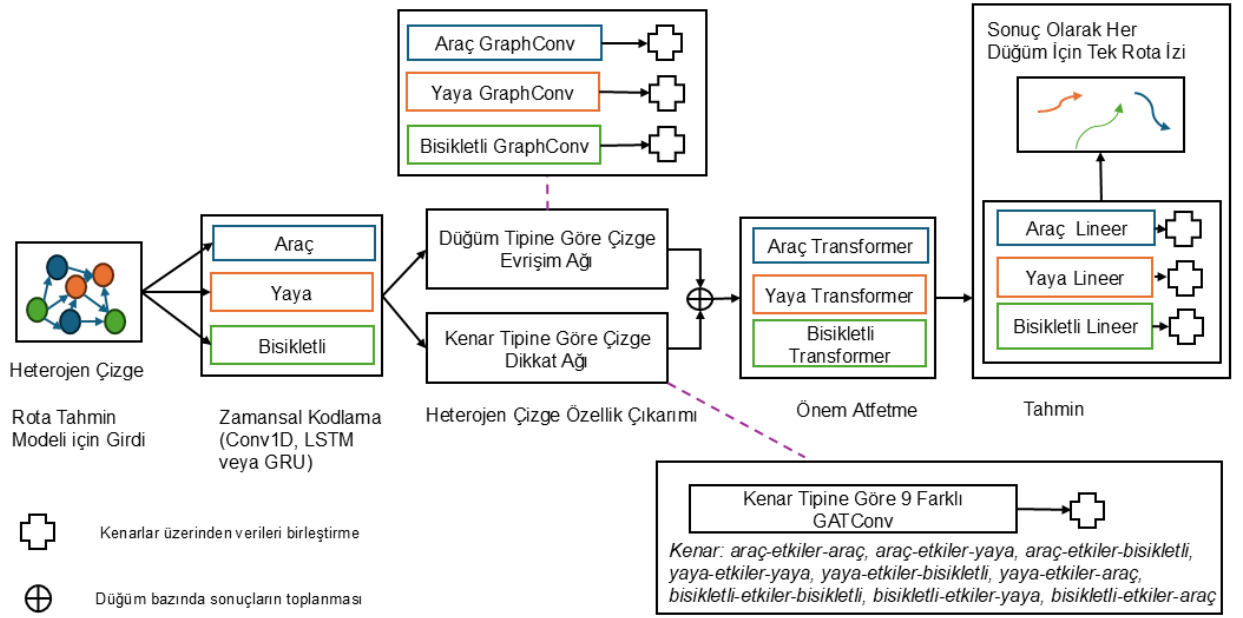


Şekil 3-4: Rota tahmin modelinin geliştirilmesi ve eğitimi aşamasında kullanılan heterojen çizgelerin oluşturulması için veri ön işleme akış diyagramı

3.5 ROTA TAHMİN MODELİ

Bu tez çalışması kapsamında heterojen trafik nesnelerinin gelecekte takip edebilecekleri rotayı tahmin etmek üzere dinamik heterojen çizge tabanlı zamansal-uzamsal ilişkileri dikkate alan, öz dikkat mekanizmalarını kullanan ROTA adında yeni bir model geliştirilmiştir. Bu model, yapısı dinamik olarak değişebilen heterojen çizgeleri girdi olarak alıp çizgedeki her bir nesne için eş zamanlı bir şekilde gelecekte takip edebilecekleri tek bir rota dizisi üretmektedir. Şekil 3.5'te görüleceği üzere model dört katmandan oluşmaktadır:

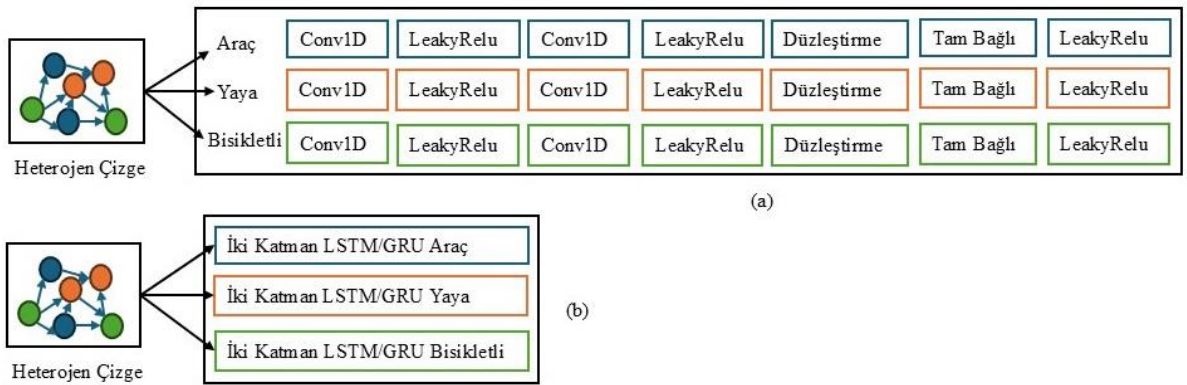
- 1- Zamansal Kodlama
- 2- Heterojen Çizge Özellik Çıkarımı
- 3- Önem Atfetme
- 4- Tahmin



Şekil 3-5: Rota tahmini için geliştirilen dinamik heterojen çizge tabanlı zamansal-uzamsal modelin yapısı

3.5.1 Zamansal Kodlama

ROTA modelinin zamansal kodlama aşamasında modele girdi olarak gönderilen heterojen çizgeden nesnelerin geçmiş hareket bilgileri zaman serisi olarak elde edilip bu verilerinden bireysel hareket karakteristikleri öğrenilmiştir. Şekil 3.6’da görüleceği üzere modelde üç tip nesne türü dikkate alınmış ve her bir nesne türü için özelleştirilmiş zamansal kodlama yapılmıştır. Böylece ilgili nesne türü bazında genel bir hareket karakteristiği yapısı öğrenilmiştir.



Şekil 3-6: Zamansal kodlamanın model yapısı. a) Conv1D, b) LSTM ve GRU

Zaman serisi verisi işleme açısından TCN (Conv1D), LSTM ve GRU olmak üzere üç farklı yöntemle model ayrı ayrı eğitilmiştir. Kullanılan bu yöntemlerde girdi verisi olarak, nesnelere ait özellik değerlerinin gözlem zaman aralığındaki değişimleri kullanılmıştır. Modellerin başlangıç parametre değerleri Tablo 3.2’de gösterilmektedir. Modeller PyTorch kütüphanesi üzerinden kullanıldığı için tabloda kütüphanenin belirlediği parametre isimleri kullanılmıştır.

Tablo 3-2: Zamansal kodlama modellerinin parametre değerleri. Parametre isimleri PyTorch kütüphanesinde kullanılan isimler ile ifade edilmiştir.

Model	Parametreler
Conv1D	Girdi verisi yapısı: $N \times F \times T$ N: Anlık ortamdaki düğüm sayısı F: Düğümlerdeki özellik sayısı olarak 7 T: Gözlem zaman aralığı 1sn için 10, 2 sn için 20 kernel_size :3 in_channels: Birinci katman 7, ikinci katman 128 out_channels: 128 padding: 1 Alt modül Tam Bağlı Katman (Linear) için: in_features: out_channels*T out_features: 128 bias: True
LSTM ve GRU	Girdi verisi yapısı: $N \times T \times F$ N: Anlık ortamdaki düğüm sayısı F: Düğümlerdeki özellik sayısı olarak 7 T: Gözlem zaman aralığı 1sn için 10, 2 sn için 20 input_size: 7 hidden_size: 128 num_layers: 2 batch_first: True Diğer parametreler varsayılan değer

Conv1D kullanılan versiyonda, Conv1D modeli bir ve iki katman halinde denenmiştir. Bir katman halinde eğitilen sistemde sonuç LeakyReLU aktivasyon fonksiyonundan geçirilmiş ve

sonular izgedeki kenarlar zerinden dğm tipine gre birleřtirilmiřtir. İki katman kullanıldığında, her bir Conv1D katmanından sonra LeakyReLU aktivasyon fonksiyonu kullanılmıř ve iki katman sonrasında elde edilen sonu dzleřtirilip tam baėlı bir katmandan geirilmiřtir.

LSTM ve GRU kullanıldığında her iki yntemde modeller iki katman olacak řekilde alıřtırılmıřtır.

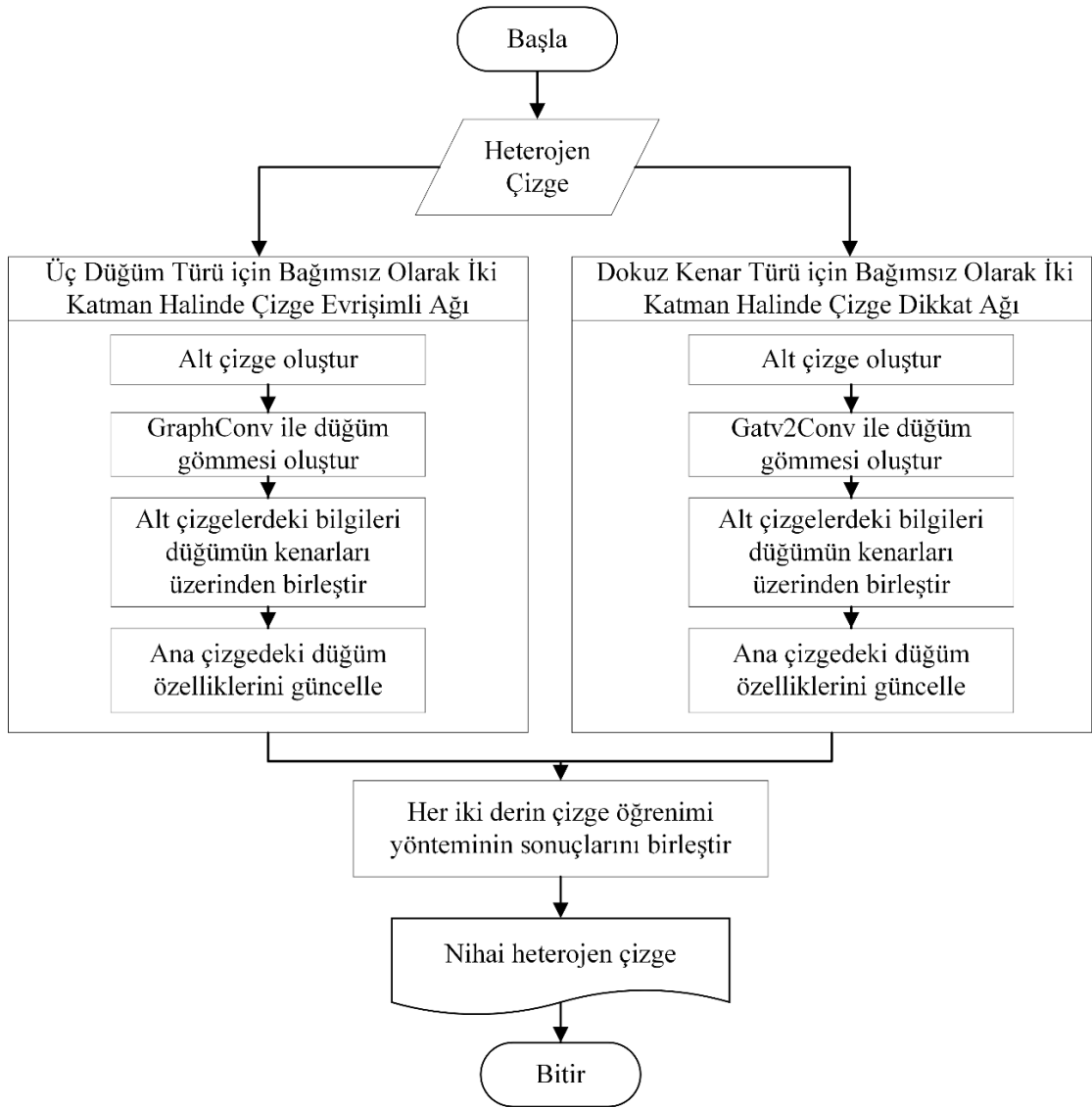
Zamansal kodlamanın ıktısı ile meknsal etkileřimi ğrenmek iin gerekleřtirilecek olan Heterojen izge zellik ıkarımı ařamasında kullanılmak zere izgelerdeki nesnelerin yeni zellik vektrleri oluřturulmuřtur.

3.5.2 Heterojen izge zellik ıkarımı

Heterojen izge zellik ıkarımı ařamasında, zamansal kodlama ařamasının sonucunda oluřturulan yeni zellik vektrleri dğm zelliėi olarak ele alınmıř ve heterojen izgeden nesneler arasındaki etkileřim ğrenilmiřtir. řekil 3.7’de akıř řeması gsterilen bu ařamada, nesneler arasındaki etkileřim  farklı izge Evriřimli Aė ve dokuz farklı izge Dikkat Aė ile ğrenilmiřtir. Her iki aėda da aktivasyon fonksiyonu olarak LeakyReLU kullanılmıřtır.

izge Evriřimli Aė blmnde her nesne tr iin bir aė ataması yapılmıř ve daha sonra ilgili nesnenin sahip olduėu kenarlara gre farklı komřuluk matrisi zerinden gelen bilgiler birleřtirilmiřtir. Bu aėda btn nesnelerin birbirini aynı oranda etkileyeceėi varsayılarak komřulardan gelen bilgilerin ortalaması ile izge dğm gmmeleri hesaplanmaktadır. İki katman halinde izge Evriřimli Aė kullanılmıřtır. Aėlarda normalizasyon iřlemi aktif edilmiřtir.

izge Dikkat Aė blmnde sistemde gz nnde bulundurulan dokuz ayrı kenar tr iin birer aė ataması yapılmıřtır. Bu aėlarda komřu nesnelerin birbirini ne derecede etkileyeceėi dikkate alınarak hesaplama yapılmaktadır. ncelikle ilgili nesnelerin sahip olduėu kenar trleri bazında dğm zellikleri komřulardan gelen aėrılıklandırılmıř veriler ile hesaplanmıřtır. Daha sonra nesnenin sahip olduėu tm kenarlar zerinden elde edilen bu veriler birleřtirilerek nesneler iin yeni zellik gmmeleri elde edilmiřtir. İki katman olarak izge Dikkat Aėı kullanılmıřtır. Aėlarda artık deėer kullanımı ve dğm atma iřlemi aktifleřtirilmiř olup z dikkat ařaması iin bařlık sayısı 4 olarak belirlenmiřtir.



Şekil 3-7: Derin çizge öğrenimi ile heterojen çizge düğüm gömmelerinin oluşturulması akış diyagramı

Her iki ağ sonucu elde edilen nihai çizge gömmeleri toplanarak nesneler için son düğüm gömmeleri elde edilmiştir. Böylece iki farklı yaklaşıma sahip ağ türünün ürettiği sonuçlar bir araya getirilerek iki ağın sağladığı avantajlar kullanılmıştır.

Her iki çizge öğrenim metodunda kullanılan parametre değerleri Tablo 3.3'te gösterilmektedir. Çizge öğrenimi için DGL kütüphanesi kullanıldığı için tablodaki parametre isimleri kütüphanede belirlendiği şekilde yazılmıştır.

Tablo 3-3: Çizge Evrişimli Ağ ve Çizge Dikkat Ağlarında kullanılan parametrelerin değerleri. Parametre isimleri DGL kütüphanesinde kullanılan isimler ile ifade edilmiştir.

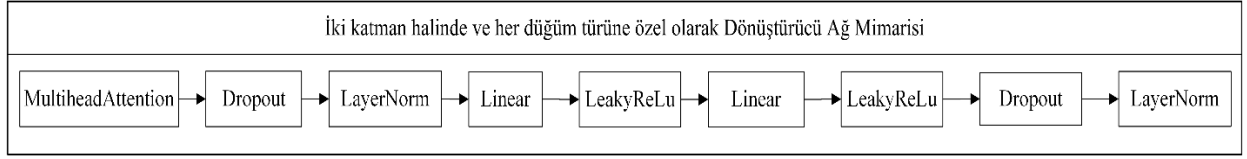
Model	Parametreler
Çizge Evrişimli Ağ (GraphConv)	in_feat: 128 out_feat: 128 norm: both weight: true bias: true activation: LeakyReLU allow_zero_in_degree: False
Çizge Dikkat Ağı (GATv2Conv)	in_feat: 128 out_feat: 128 num_heads: 4 feat_drop: 0,5 activation: LeakyReLU allow_zero_in_degree: False residual: True Diğer parametreler varsayılan değerdir.

3.5.3 Önem Atfetme

Önem atfetme aşamasının amacı nesneler için elde edilen özellik gömmelerindeki verilerin rota tahmininde sonuca ne kadar etki edeceğinin belirlenmesidir. Bu aşamada nesne türüne göre özelleştirilmiş üç adet Dönüştürücü Ağ bulunmaktadır. İki katman halinde kullanılan Dönüştürücü Ağ ile nesnelerin önceki aşamalar sonucunda elde edilen gömme değerleri için dikkat değerleri hesaplanmış ve gömmeler bu dikkat değerleri ile ağırlıklandırılmıştır. Dönüştürücü Ağ oluşturulurken Vaswani vd. [44] tarafından geliştirilen ağınlı kodlama aşaması kullanılmıştır. Kullanılan Dönüştürücü Ağlarda çok başlıklı öz dikkat ve artık değer mekanizması aktif edilerek önem değerleri hesaplanmıştır.

Şekil 3.8’de mimarisi gösterilen önem atfetme adımı kullanılarak kullanılan Dönüştürücü Ağlarda, öncelikle 4 başlıklı öz dikkat mekanizması ile dikkat değerleri hesaplanmıştır. Daha sonra sonuçlar düğüm atma katmanından geçirilmiş ve artık değerler hesaplanmıştır. Bu artık değerler ilk gömme değerlerine eklenmiştir. Son aşama olarak sırasıyla normalizasyon, iki katman halinde tam bağlı

ağ, normalizasyon, düğüm atma ve normalizasyon işlemlerinden geçirilerek özellik vektöründeki değerlerin ağırlıklandırılmış hali elde edilmiştir. Tablo 3.4’te dönüştürücü ağın parametre değerleri verilmiştir. Parametre isimleri PyTorch kütüphanesinde belirlenen isimler ile ifade edilmiştir.



Şekil 3-8: Önem atfetme aşamasının mimarisi

Tablo 3-4: Önem atfetme aşamasında kullanılan modellerin parametre değerleri. Parametre isimleri PyTorch kütüphanesinde kullanılan isimler ile ifade edilmiştir.

Model	Parametre
MultiheadAttention	embed_dim: 128 num_head: 4 Diğer parametreler varsayılan değerdir.
Dropout (Her iki katman için)	p: 0,5 inplace: False
LayerNorm (Her iki katman için)	normalized_shape: 128 Diğer parametreler varsayılan değerdir.
Linear (Birinci Katman)	in_features: Birinci Katman128, İkinci Katman 256 out_features: Birinci Katman 256, İkinci Katman 128 bias: True

3.5.4 Tahmin Başlığı

Tahmin katmanında nesne türüne göre özelleştirilmiş üç ayrı model bulunmaktadır. İlk üç aşama sonucu elde edilen veriler kullanılarak tahmin katmanında sırasıyla tam bağlı katman, LeakyReLU aktivasyon fonksiyonu ve normalizasyon işlemi uygulanmıştır. Her bir nesne için nesnenin farklı kenar bağlantıları üzerinden elde edilen sonuçlar birleştirilerek nesnelerin konumundaki değişimler tahmin edilmiştir. Böylece eş zamanlı olarak tüm nesneler için belirlenen tahmin zaman aralığı boyunca birer adet rota değişim dizisi oluşturulmuştur. Daha sonra nesnenin gözlem sürecinde son bulunduğu noktanın koordinat değerleri üzerine tahmin sonuçları eklenerek

nihai konumlar elde edilmiştir. Tahmin başlığında kullanılan tam bağlı katmanın parametre değerleri Tablo 3.5’te gösterilmiştir. Parametre isimleri PyTorch kütüphanesinde olduğu gibi kullanılmıştır.

Tablo 3-5: Tahmin katmanında kullanılan modelin parametre değerleri. Parametre isimleri PyTorch kütüphanesinde kullanılan isimler ile ifade edilmiştir.

Model	Parametre
Linear	in_features: 128
	out_features: 256
	bias: True
LayerNorm	normalized_shape: 128
	Diğer parametreler varsayılan değerdir.

4. BULGULAR

Bu tez kapsamında otonom araçlar için trafikteki hareketli nesnelerin gelecekteki rotasını tahmin etmek için geliştirilen sistemler ile iki ayrı deney gerçekleştirilmiştir. Birincisi tez kapsamında geliştirilen ROTA modelinin farklı veri setleri ve parametrelerle eğitilmesi ve test edilmesidir. İkinci deney ise LiDAR verisinin dönüşümü, nesne algılama, takip ve tahmin süreçlerinin peş peşe işletildiği ve önceden eğitilmiş modellerin kullanıldığı uçtan uca rota tahmini yapan Uçtan-Uca Rota yönteminin eğitilmesi ve test edilmesidir.

Tez kapsamında geliştirilen ROTA modeli iki ayrı veri seti üzerinde denenmiştir. Öncelikle model, ihtiyaç duyulan verilerin hazır olarak bulunduğu WAYMO veri seti üzerinde 1 ve 2 sn gözlem süresi ile 1, 2, 3, 4, 5 ve 8 sn tahmin süreleriyle ayrı ayrı eğitilmiş ve en iyi ADE sonucunun elde edildiği durumda model parametreleri kaydedilmiştir. Daha sonra model, Argoverse veri seti kullanılarak eğitilmiş ve en iyi ADE değerinin elde edildiği durumda model parametreleri kaydedilmiştir.

Uçtan-Uca Rota yönteminin eğitiminde algılama ve takip aşamasındaki modeller yazarlar tarafından yayınlanan şekliyle kullanılmış ve ROTA modeli bu aşamalar ile elde edilen verilerle eğitime devam edilmiştir.

Elde edilen bulguların ışığında, geliştirilen ROTA modelinin ve uçtan uca rota tahmin yönteminin son dönem çalışmalarına göre daha az hata değerleriyle sonuç ürettiği gözlemlenmiştir.

4.1 ÇALIŞMA ORTAMI KONFİGÜRASYONU, PARAMETRELER ve KISITLAR

Rota tahmin modelinin geliştirilmesi ve eğitimi esnasında 128 GB Ram ve Nvidia GeForce RTX 4070 Ti ekran kartına sahip bir bilgisayar kullanılmıştır. Model, Deep Graph Library (DGL) [121] kütüphanesi ve Python 3.10 kullanılarak geliştirilmiştir. Optimizasyon tekniği olarak Adam Optimizasyon yöntemi 0,001 öğrenme oranı ile kullanılmıştır. Kayıp fonksiyonu olarak Ortalama Kare Hatası fonksiyonu kullanılmıştır. Eğitim döngü sayısı olarak 70 kullanılıp 256 çizgede bir geri yayılım yapılmıştır. İzlem zaman aralığı 1 ve 2 sn; tahmin zaman aralığı 1, 2, 3, 4, 5 ve 8sn olarak belirlenmiş ve her saniye için 10 çerçeve kullanılmıştır. Rota tahmin aşaması için değerlendirme kriteri olarak ADE ve FDE kullanılmıştır. Her iki değer sonucları metre cinsinden sunulmuştur. WAYMO ve Argoverse veri setinden ön işlemler sonucu sırasıyla eğitim için 62470

ve 75061, doğrulama için 15021 ve 16020'ser adet çizge elde edilmiştir. Ezberlemeyi engellemek amaçlı, eğitim esnasında her bir eğitim döngüsünde çizgeler karışık sırada modele gönderilmiştir.

Modelin eğitimi aşamasında aşağıdaki kısıtlamalar ve durumlar göz önünde bulundurulmuştur:

- Oluşturulan çizgeler o anda tek bir nesne türü barındırıp homojen bir yapı sergileyebileceği gibi birden fazla nesne türünü barındırıp heterojen yapıda da olabilmektedir. Çizgeler arasında herhangi bir heterojenlik derecesine odaklanılmamış tüm çizgeler kullanılmıştır.
- Modelin sonuç üretmesi aşamasında ortamda bulunan tüm nesneler dikkate alınırken, kayıp değeri hesaplanıp model parametreleri güncellenirken $(\text{gözlem süresi} + \text{tahmin süresi}) \times 10$ adet çerçeve boyunca ortamda bulunan nesneler dikkate alınmıştır.
- Girdi verileri oluşturulurken herhangi bir özel senaryoya odaklanılmamış veri setlerindeki tüm dosyalar ve dosyalardaki tüm çerçeveler kullanılmıştır.
- Veri setlerinde ortak olarak bulunan veriler veya bu verilerden hesaplanabilecek bilgiler çizge oluşturma esnasında düğüm özellik vektörüne eklenmiştir.
- Veri setlerinde genel olarak araçlar ve yayalar açısından yanlılık bulunmaktadır [127-128]. Bu nedenle modelin bisikletliler açısından eğitimi için yeterli veri bulunmamaktadır. Bu durum bisikletliler açısından hatanın artmasına sebep olmaktadır.
- Hava durumu (yağmur, güneş, sis, kar vb.) ve gün içerisindeki zaman dilimi (sabah, öğle, akşam, gece, alacakaranlık vb.) bilgileri her iki veri setinde sunulmadığı için modelde bu bilgiler parametre olarak kullanılmamış ve bu bilgilere bağlı olarak herhangi bir özelleştirilmiş senaryo geliştirilmemiştir. Argoverse veri setinde herhangi bir detay bilgiye yer verilmeyip farklı hava koşullarında ve ışık ortamlarında bilginin elde edildiği ifade edilmiştir. WAYMO veri setinde veriler ağırlıklı olarak güneşli günlerde ve gündüz saatlerinde elde edilmiştir [127]. Model bu bilgiler üzerinde herhangi bir ayrıştırma yapılmadan tüm veriler ile eğitilmiştir.
- Heterojen çizgelerde düğüm türü ve kenar türü arttıkça modelin karmaşıklığı ve cevap verme süresi polinomsal bir şekilde artmaktadır. Bu nedenle düğüm türü sayısı 3 ile sınırlandırılmıştır.

4.2 KARŞILAŞTIRMA İÇİN KULLANILAN ÇALIŞMALAR

Model geliştirilirken algılama ve takip süreçlerinin de işletileceği bir süreç planlandığı için karşılaştırma çalışması olarak WAYMO veya Argoverse veri setindeki lidar verilerini kullanan çalışmalar seçilmiştir. Bu veri setini kullanan ve karşılaştırma için kullanılan çalışmalar aşağıdaki gibidir:

Casas vd. [25] tarafından geliştirilen ve araçlara odaklanan DeTra çalışmasında LiDAR verilerinden öncelikle nesneler algılanmış, nesnelerin izleri oluşturulmuş ve rota tahmini gerçekleştirilmiştir. Çalışmada gözlem süresi olarak 0,5 sn, tahmin süresi olarak 5 sn kullanılmış ve sadece araçlar için rota tahmini yapılmıştır. Bu çalışmada sonuçlar hem WAYMO hem de Argoverse veri seti için yayınlanmıştır.

Z. Wang vd. [129] tarafından yayınlanan teknik raporda hazır modeller kullanılarak nesne algılama ve rota tahmin süreçleri uçtan uca işletilmiştir. 2 sn gözlem ile gelecekteki 3 sn için rota tahmini Argoverse veri seti üzerinde gerçekleştirilmiştir. Rota tahmin aşaması için Y. Hu vd. [130] tarafından geliştirilen UniAD algoritması kullanılmıştır.

Y. Xu vd. [23] tarafından geliştirilen çalışmada 2 sn gözlem ile gelecekteki 3 sn için nesne algılama, takip ve rota tahmin işlemleri Argoverse veri setinde gerçekleştirilmiştir. Rota tahmin aşaması için S. Shi vd. [131] tarafından geliştirilen MTR algoritması kullanılmış ve model Argoverse veri setinde eğitmeye devam edilerek uyumlama işlemi gerçekleştirilmiştir.

Najibi vd. [24] tarafından geliştirilen çalışmada WAYMO veri setindeki LiDAR verileri kullanılarak denetimsiz nesne algılama, nesnelerin izlerinin oluşturulması ve rota tahmini gerçekleştirmişlerdir. Bu çalışmanın insan etiketlemeleri kullanılarak elde edilen ve daha az hata oranlarına sahip sonuçları karşılaştırma verisi olarak ele alınmıştır. Çalışmada gözlem süresi olarak 1 sn, tahmin süresi olarak 8 sn kullanılmıştır.

Zhang vd. [73] tarafından geliştirilen STINet çalışmasında LiDAR verileri üzerinden öncelikle yayalar tespit edilmiş ve tespit edilen yayalar için rota tahmini gerçekleştirilmiştir. Çalışmada WAYMO veri seti üzerinde 1 sn gözlem ile gelecekteki 3 sn için sadece yayalar açısından rota tahmini yapılmıştır.

Çalışmalarda rota tahmin sonuçları elde edilirken algılama ve takip aşamaları sonucunda doğru olarak etiketlenen nesnelerin gelecekteki rotaları tahmin edilip raporlanmıştır. Karşılaştırma esnasında yazarlar tarafından yayınlanan sonuçlar kullanılmıştır.

4.3 ROTA MODELİNİN SONUÇLARI

ROTA modeli üç ayrı deney seti ile eğitilip sonuçlar elde edilmiştir.

- **Deney Seti 1:** WAYMO veri setinde ön kameranın görüş açısında ve otonom araca uzaklığı 50 metre ve daha yakın olan nesneler dikkate alınarak çizgeler elde edilmiş ve Deney Seti 1 oluşturulmuştur.
- **Deney Seti 2:** WAYMO veri setinde ön kameranın göremeyeceği açılardaki nesnelerinde ele alınması için LiDAR sensörüyle algılanan 50 metre ve daha yakın olan tüm nesneler kullanılarak çizgeler oluşturulmuş ve Deney Seti 2 elde edilmiştir.
- **Deney Seti 3:** Argoverse veri setinde otonom araca 50 metre ve daha yakın olan ve LiDAR sensörüyle algılanan nesneler kullanılarak Deney Seti 3 elde edilmiştir.

Modelin ilk geliştirilme aşamasında Conv1D bir katmana sahip olacak şekilde tasarlanmış ve 1 sn gözlem verileri kullanılarak farklı tahmin süreleri için Deney Seti 1 üzerinde ayrı ayrı eğitilmiştir. Elde edilen ilk sonuçlar Tablo 4.1’de gösterilmektedir.

Tablo 4-1: ROTA modelinin tek katman Conv1D ve Deney Seti 1 kullanıldığında 1 sn gözlemle elde ettiği sonuçlar

Model	Tahmin Zaman Aralığı	Genel ADE/FDE	Araç ADE/FDE	Yaya ADE/FDE	Bisikletli ADE/FDE
ROTA Covn1D 1 Katman	1sn	0,21/0,39	0,25/0,47	0,13/0,25	0,45/0,71
	2sn	0,38/0,80	0,45/0,94	0,24/0,52	1,04/2,02
	3sn	0,50/1,01	0,59/1,16	0,31/0,69	1,65/3,08
	4sn	0,72/1,54	0,85/1,80	0,44/0,99	2,22/4,42
	5sn	0,83/1,84	0,94/2,09	0,57/1,31	2,85/5,55
	8sn	1,49/3,55	1,70/4,18	1,04/2,30	4,88/8,07

Hata oranları tahmin süresi arttıkça beklenildiği üzere artmaktadır. Bu nedenle hataları azaltmaya yönelik olarak Conv1D iki katman halinde kullanılarak eğitim süreçleri Deney Seti 1 üzerinde tekrarlanmış ve elde edilen sonuçlar Tablo 4.2’de gösterilmiştir. Sonuçlar incelendiğinde iki katman halinde uygulama gerçekleştirildiğinde hata oranlarının düştüğü görülmüştür. Daha fazla katman ile elde edilen sonuçlar ile hareket karakteristiğinin daha iyi öğrenildiği tespit edilmiştir. Ancak bisikletliler için 8sn tahminde süre uzamasına rağmen hata oranında düşüş görülmüştür. Bunun nedenin, iki katman halinde kullanılan Conv1D’nin rota örüntülerine daha çok odaklanması ve 8 sn için kullanılan rota izlerinde ortamda 90 kare boyunca bulunmayan nesnelerin elenmesiyle örüntüyü bozan rotaların elenme olasılığı olabileceği düşünülmektedir.

Gelecekteki 3 sn zaman aralığında yayalar için tahminde bulunan STINet uygulaması ile karşılaştırıldığında geliştirilen modelin ADE ve FDE açısından sırasıyla %32 ve %7 kazanım sağladığı görülmektedir.

Tablo 4-2: Conv1D iki katman halinde kullanıldığında ROTA modeliyle Deney Seti1 üzerinde elde edilen sonuçlar. En iyi değerler kalın ve italik olarak ifade edilmiştir.

Model	Tahmin Zaman Aralığı	ADE/FDE	Araç ADE/FDE	Yaya ADE/FDE	Bisikletli ADE/FDE
ROTA Covn1D 2 Katman	1sn	0,12/0,26	0,15/0,33	0,06/0,12	0,47/0,92
	2sn	0,28/0,59	0,34/0,69	0,14/0,36	1,04/2,16
	3sn	0,40/0,86	0,47/0,99	0,23/0,54	1,86/3,82
	4sn	0,56/1,18	0,64/1,34	0,36/0,78	2,73/5,36
	5sn	0,73/1,60	0,83/1,84	0,46/1,03	3,67/7,37
	8sn	1,38/3,28	1,62/3,96	0,90/1,98	3,56/6,04
Zhang vd. [73]	3sn			0,34/0,58	

5 ve 8. saniyelerdeki hata oranlarını düşürmek adına zaman serisi modeli iki katman halindeki LSTM ve GRU için Deney Seti 1 üzerinde ayrı ayrı eğitilmiştir. Tablo 4.3'te görüleceği üzere tekrarlayan sinir ağları daha az hata ile sonuç üretebilmektedir. Conv1D ile karşılaştırıldığında; LSTM ADE ve FDE açısından sırasıyla %16 ve %11, GRU ise ADE ve FDE açısından sırasıyla %13 ve %10 kazanç sağlamıştır. Bu da hala kısa süreli gözlemlerin yapıldığı durumlarda LSTM ve GRU'nun zaman serilerini işlemede daha etkili olduğunu göstermektedir.

Tablo 4-3: ROTA modelinin farklı zaman serisi işleme yöntemlerine göre Deney Seti 1 üzerinde gelecekteki 5 ve 8 sn için rota tahmin sonuçları. En iyi değerler kalın ve italik olarak ifade edilmiştir.

Tahmin Zaman Aralığı	Model	Zaman Serisi Modeli	ADE/FDE	Araç ADE/FDE	Yaya ADE/FDE	Bisikletli ADE/FDE
5sn	Casas vd. [25]			1,87/4,29		
	ROTA	Conv1D	0,73/1,60	0,83/1,84	0,46/1,03	3,67/7,37
		LSTM	0,61/1,42	0,66/1,57	0,45/1,02	3,27/6,64
		GRU	0,63/1,43	0,69/1,61	0,44/1,00	3,09/6,30
8sn	Najibi vd. [24]		1,89/4,79			
	ROTA	Covn1D	1,38/3,28	1,62/3,96	0,90/1,98	3,56/6,04
		LSTM	1,34/3,25	1,53/3,81	0,94/2,16	3,53/6,38
		GRU	1,35/3,18	1,55/3,73	0,90/2,05	4,98/8,64

Tablo 4.3'te karşılaştırma için kullanılan çalışmaların sonuçları orijinal makalelerinden alınmıştır. Bu makalelerde hata oranları, algılama ve takip aşamaları sonucunda doğru olarak

işaretlenmiş nesnelerin rotaları kullanılarak hesaplanmaktadır. ROTA sonuçları ise veri setinde hazır bulunan LiDAR sensör etiket verileri denetiminde elde edilen sonuçlardır. Bu bilgiler ışında aşağıdaki sonuçlar elde edilmiştir:

Casas vd. tarafından yapılan çalışmada sonuçlar sadece araçlar açısından 0,5 sn gözlemle gelecekteki 5 sn için tahmine dayalı yayınlamıştır. Buna göre ROTA modeli 1 sn gözlemle Conv1D, GRU ve LSTM kullanım durumları için ADE açısından sırasıyla %55, %63 ve %64; FDE açısından sırasıyla %57, %62 ve %63 kazanç sağlamıştır, bu sonuçların yüksek olmasında gözlem süresindeki fark ve etiket verilerinin kullanılması etkilidir.

Najibi vd. tarafından yapılan çalışmada sonuçlar 1 sn gözlemle gelecekteki 8 sn tahmin için yayınlanmıştır. Buna göre ROTA modeli 1 sn gözlemle Conv1D, GRU ve LSTM kullanım durumları için ADE açısından sırasıyla %26, %28 ve %29; FDE açısından sırasıyla %31, %33 ve %32 kazanç sağlamıştır. Bu sonuçların yüksek olmasında etiket verilerinin kullanılması etkilidir.

Modelin gözlem süresi 2 sn olarak değiştirilerek gözlem süresinin modelin sonucuna etkileri izlenmiştir. Bu amaçla model ile 2 sn (20 çerçeve) için geçmiş bilgisi kullanılarak gelecekteki 5 sn (50 çerçeve) için rota tahmini yapılmıştır. Tablo 4.4'te de görülebileceği üzere en yavaş hareket karakteristiğine sahip yayalar için gözlem süresinin sadece 1 sn arttırılmasının bir katkısı olmadığı gözlemlenmiştir. En yüksek kazanım orta hızlı hareket eden bisikletliler için sağlanmıştır, sürenin 1 sn artması hareket karakteristiğini anlamak için önemli katkı sunmuştur. Bisikletlilerde Conv1D, LSTM ve GRU için sırasıyla ADE açısından %29, %9 ve %15; FDE açısından %26, %9 ve %16 kazanç sağlanmıştır. Doğası gereği hızlı hareket eden araçlar için yapılan tahminlere de gözlem süresinin arttırılması önemli katkı sunmamış ve hareket karakteristiğinin kısa gözlemle de elde edilebileceği görülmüştür. Araçlar için kazanç sadece Conv1D kullanılan modelde ADE ve FDE açısından sırayla %12 ve %8 olmuştur. Genel olarak bakıldığında en büyük kazançlar örüntü takibine odaklanan Conv1D tabanlı modelde elde edilmiştir.

Tablo 4-4: ROTA modelinin 2 sn gözlem ile gelecekteki 5 sn için Deney Seti 1 üzerindeki tahmin sonuçları. En iyi değerler kalın ve italik olarak ifade edilmiştir.

Gözlem Süresi	Zaman Serisi Modeli	ADE/FDE	Araç ADE/FDE	Yaya ADE/FDE	Bisikletli ADE/FDE
1 sn	Conv1D	0,73/1,60	0,83/1,84	0,46/1,03	3,67/7,37
	LSTM	0,61/1,42	0,66/1,57	0,45/1,02	3,27/6,64
	GRU	0,63/1,43	0,69/1,61	0,44/1,00	3,09/6,30
2 sn	Conv1D	0,65/1,49	0,73/1,69	0,46/1,06	2,60/5,40
	LSTM	0,63/1,43	0,69/1,59	0,47/1,04	2,98/6,02
	GRU	0,61/1,46	0,68/1,64	0,46/1,05	2,61/5,26

ROTA modeli, otonom aracın sadece önündeki nesnelerin değil yanındaki ve arkasındaki nesnelerin de nasıl bir etki yaratacağını belirlemek için Deney Seti 2 üzerinde farklı tahmin sürelerinde eğitilmiştir. Tablo 4.5'te de görülebileceği üzere heterojen çizgelerin oluşturulmasında aracın etrafında bulunan 50 metre mesafe içerisindeki tüm nesnelerin kullanılması ROTA modelinin daha iyi sonuç üretmesini sağlamıştır. Böylece aracın yanında veya arkasında kalsa bile tüm nesnelerin birbiriyle etkileşimi dikkate alınabilmektedir. Deney Seti 2 kullanarak eğitilen ROTA modelleri gelecekteki 5 sn için sırasıyla ADE ve FDE açısından Conv1D kullanıldığında %10 ve %13, LSTM kullanıldığında %26 ve %23, GRU kullanıldığında %13 ve %25; gelecekteki 8 sn için sırasıyla ADE ve FDE açısından Conv1D kullanıldığında %21 ve %19, LSTM kullanıldığında %29 ve %25, GRU kullanıldığında %29 ve %24 kazanç ile sonuç üretmiştir. Deney seti 2 üzerinde eğitilen modeller uçtan uca yöntemde kullanılmak üzere kaydedilmiştir.

Tablo 4-5: ROTA modelinin Deney Seti 2 üzerinde 1 sn gözlem ile gelecekteki 5 ve 8 sn için rota tahmin sonuçları. En iyi değerler kalın ve italik olarak ifade edilmiştir.

Tahmin Süresi	Zaman Serisi Modeli	Sadece Ön Kameraya Yansıyan Nesneler için ADE/FDE	Tüm Nesneler için ADE/FDE
5 sn	Conv1D	0,73/1,60	<i>0,65/1,38</i>
	LSTM	0,61/1,42	<i>0,45/1,09</i>
	GRU	0,63/1,43	<i>0,45/1,07</i>
8 sn	Conv1D	1,38/3,28	<i>1,08/2,65</i>
	LSTM	1,34/3,25	<i>0,94/2,42</i>
	GRU	1,35/3,18	<i>0,95/2,39</i>

Deney Seti 2 ile elde edilen 8 sn sonrası için tahmin sonuçları Najibi vd. tarafından yapılan çalışma ile karşılaştırıldığında ROTA modelinin ADE ve FDE açısından sırayla Conv1D ile %42 ve %44, LSTM ile %50 ve %49, GRU ile %49 ve %50 kazanç sağladığı görülmüştür.

Ayrıca ROTA modeli Deney Seti 2 üzerinde 5 sn tahmin süresi için araçlar özelinde ADE ve FDE açısından Conv1D ile 0,75 ve 1,54 m; LSTM ile 0,48 ve 1,16 m; GRU için 0,48 ve 1,16 m hata ile sonuç üretmiştir. Bu sonuçlar ile Casas vd. tarafından 5 sn sonrası için rapor edilen tahminler karşılaştırıldığında ROTA modelinin araçlar için ADE ve FDE açısından Conv1D ile %59 ve %64; LSTM ve GRU ile %74 ve %72 kazanç sağladığı görülmüştür. Sonuçların yüksek olmasında ROTA modelinin 1 sn gözlem ve etiket verileri ile tahmin yapmasının etkisi de bulunmaktadır.

Geliştirilen yapay zekâ modelinin farklı veri setlerine uygulanabilirliğinin test edilmesi için ROTA modeli, yapısı ve karakteristiği farklı olan ikinci bir veri setinde de denenmiştir. Bu amaçla

Argoverse veri seti üzerinden oluşturulan Deney Seti 3 kullanılmıştır. Eğitim ve test işlemlerinde karşılaştırma çalışmalarına uyum sağlamak açısından 1 sn gözlem ile gelecekteki 5 sn ve 2 sn gözlem ile gelecekteki 3 sn için rota tahmini yapılmıştır. Tablo 4.6’da görülebileceği gibi Casas vd. tarafından yapılan çalışma ile karşılaştırıldığında araçlar açısından model, ADE ve FDE için sırasıyla Conv1D kullanıldığında %40 ve %35; LSTM kullanıldığında %42 ve %37; GRU kullanıldığında %41 ve %36 kazanç ile sonuç üretmiştir. 2 sn gözlem ile gelecekteki 3 sn için tahmin durumunda Xu Y. vd. tarafından rapor edilen sonuçlarla kıyaslandığında ROTA modelinin tüm nesneler için ADE ve FDE açısından sırayla Conv1D kullanıldığında %81 ve %63; LSTM kullanıldığında %81 ve %65; GRU kullanıldığında %82 ve %65 kazanç sağladığı görülmüştür. Kazanç oranlarının yüksek olmasında ROTA modeli eğitilirken Argoverse veri setindeki etiket bilgilerinin kullanılması etkili olmuştur.

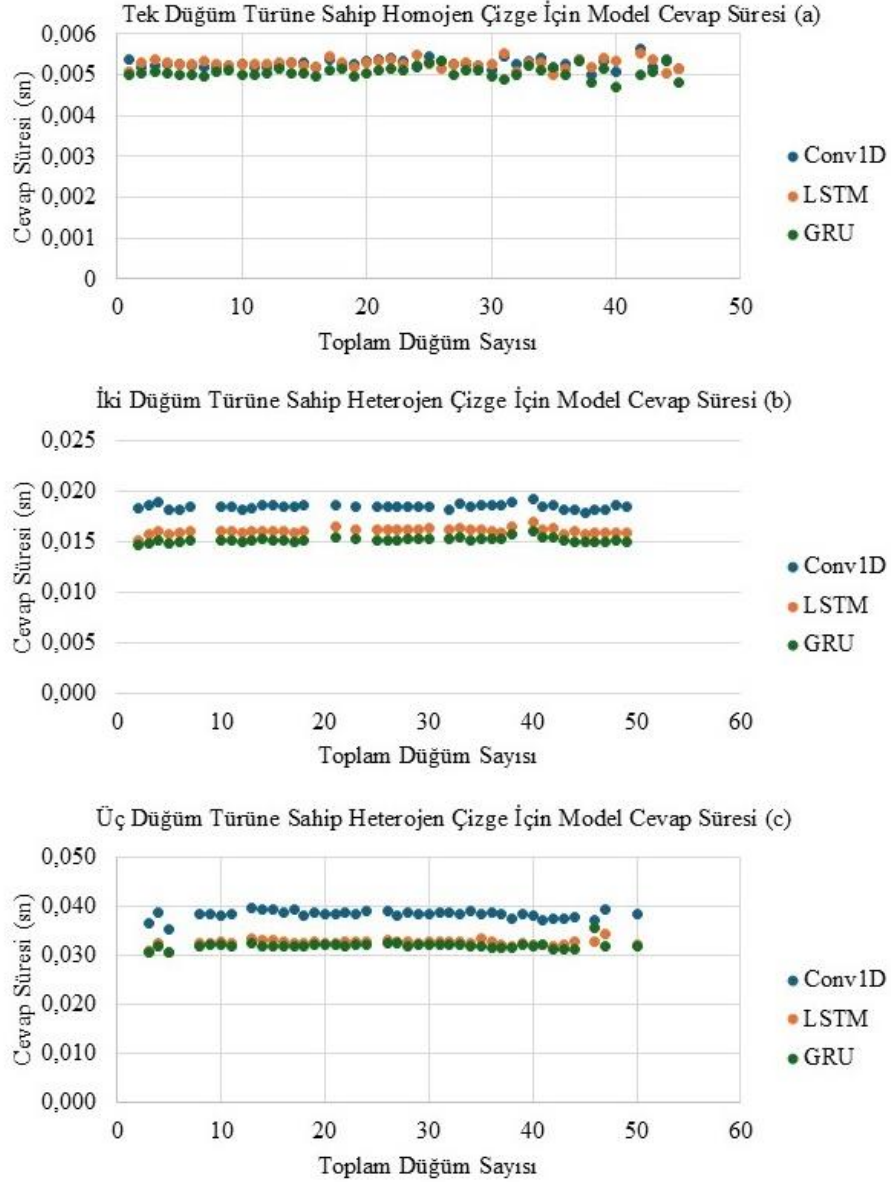
Tablo 4-6: ROTA modelinin Argoverse veri seti (Deney veri seti 3) üzerinde 1 sn gözlem ile gelecekteki 5 sn ve 2 sn gözlem ile gelecekteki 3 sn için rota tahmin sonuçları. En iyi değerler kalın ve italik olarak ifade edilmiştir.

Gözlem Süresi	Tahmin Süresi	Model	Zaman Serisi Modeli	ADE/FDE	Araç ADE/FDE	Yaya ADE/FDE	Bisikletli ADE/FDE
0,5 sn	5 sn	Casas vd. [25]			1,54/3,59		
1 sn	5 sn	ROTA	Conv1D	0,84/2,06	0,92/2,32	0,51/1,15	3,16/6,20
			LSTM	0,82/2,00	0,89/2,23	0,54/1,16	3,38/6,46
			GRU	0,82/2,02	0,90/2,28	0,50/1,11	3,18/6,34
2 sn	3 sn	Wang Z. vd. [129]		3,22/3,76			
		Xu Y. vd. [23]		2,14/2,43			
		ROTA	Conv1D	0,40/0,89	0,46/1,01	0,21/0,49	1,72/3,20
			LSTM	0,39/0,85	0,44/0,95	0,23/0,51	1,43/2,75
			GRU	0,38/0,84	0,42/0,94	0,21/0,48	1,57/3,06

Modeller için hata oranlarının yanı sıra önemli olan bir başka kriter de modelin sonuç üretme süresidir. Trafik ortamında o anda içerisinde bulunan ortamda nesne dağılımı açısından homojen ve heterojen durumlar oluşabilmektedir. ROTA modeli bu durumu karşılayabilmekte ve kendisine gönderilen homojen veya heterojen yapıdaki çizgeleri işleyebilmektedir. Tablo 4.7’de görülebileceği gibi ROTA modelinin sonuç üretme süresi homojen çizgeler için ortalama 0,005, iki düğüm türüne sahip heterojen çizgeler için 0,016 ve üç düğüm türüne de sahip heterojen çizgeler için ortalama 0,03 sn’dir.

Tablo 4-7: Düzüm çeşidi sayısına göre ROTA modelinin farklı zaman serisi işleme yöntemlerine göre Deney Seti 2 üzerinde ortalama cevap üretme süreleri

Düzüm Türü Sayısı	Conv1D (sn)	LSTM (sn)	GRU (sn)
1	0,005270687	0,005287748	0,005060211
2	0,017806486	0,015430597	0,014517461
3	0,03794762	0,030876098	0,032483108



Şekil 4-1: Düzüm türü sayısına göre ROTA modelinin cevap üretme süresi. a) Homojen çizge b) İki tür düğüm olan heterojen çizge c) Üç tür düğüm olan heterojen çizge

Şekil 4.1'deki grafiklerden görüleceği üzere süreler en çok ortamdaki düğüm ve kenar türü sayısına bağlı olarak artmaktadır. Çünkü ROTA modelinde; homojen çizgelerde tek, iki tür düğüm olduğunda 4 ve üç düğüm türü olduğunda 9 adet komşuluk matrisi kullanılarak düğüm gömmeleri oluşturulmaktadır. Heterojenlik hesaplama karmaşıklığını arttırarak sürelerin uzamasına sebep olmaktadır. Ortamdaki düğüm sayısının önemli bir etkisi düğüm sayıları az olduğu için önemli bir etkiye sebep olmamıştır. Heterojen bir çizgede hesap karmaşıklığı V düğüm türleri kümesi, t bir düğüm türü, $|V_t|$ t türündeki düğümlerin sayısı, E kenar türleri kümesi, k bir kenar türü, $|E_k|$ ilgili kenar türündeki kenar sayısı, H dikkat başlığı sayısı, $f_{giriş}$ modelin girdi boyutu ve $f_{çıkış}$ modelin çıktı boyutu olmak üzere Çizge Dikkat Ağı için (4.1) ve Çizge Evrişimli ağ için (4.2) numaralı denklem ile hesaplanmaktadır:

$$O(\sum_{t \in V} |V_t| \cdot H \cdot f_{çıkış} + \sum_{k \in E} |E_k| \cdot H \cdot f_{çıkış}) \quad (4.1)$$

$$O(\sum_{t \in V} |V_t| \cdot f_{giriş} \cdot f_{çıkış} + \sum_{k \in E} |E_k| \cdot f_{giriş}) \quad (4.2)$$

4.4 UÇTAN-UCA ROTA YÖNTEMİNİN SONUÇLARI

Bu tez kapsamında, LiDAR verisinin okunması, bu veriden nesnelerin algılanması, algılanan nesnelerin LiDAR çerçeveleri arasında takip edilip kimlik numarası ve izlerin oluşturulması ile nesneler için rota tahmin işleminin yapılmasını kapsayan Uçtan-Uca Rota yöntemi önerilmiştir. Bu yöntem algılama ve takip aşamalarında kullanılan modellere uygunluğu açısından WAYMO veri setinden oluşturulan Deney seti 2 üzerinde denenmiştir. Gözlem süresi olarak 1 sn, tahmin süresi olarak 5 ve 8 sn zaman aralıkları kullanılmıştır.

Yöntemde LiDAR verisinin okunması ve nesnelerin algılanması aşamasında kullanılan CenterPoint uygulamasının açık kaynak kodları kullanılarak öncelikle WAYMO veri setinin eğitim ve doğrulama setlerindeki dosyalar ön işleminden geçirilmiş ve CenterPoint için girdi verisi hazırlanmıştır. Daha sonra hazırlanan bu veriler kullanılarak önceden eğitilmiş CenterPoint modeli aracılığıyla bulut kümesi işlenmiş ve 3B nesne algılama işlemleri gerçekleştirilmiştir. Algılanan 3B nesne çevreleyen kutu bilgileri, takip aşaması için kullanılan MCTrack modelinin açık kaynak kodları kullanılarak ön işleminden geçirilmiş ve modelin istediği standart veri formatına dönüştürülmüştür. Standart formata dönüştürülen veriler kullanılarak önceden eğitilmiş MCTrack modeli aracılığıyla nesneler takip edilerek izleri oluşturulmuş ve her birine kimlik numarası atanmıştır.

Algılama ve takip sonucunda elde edilen verilerden Bölüm 3.4’te açıklandığı şekliyle heterojen çizgeler elde edilmiş ve bu çizgeler kullanılarak ROTA modeli uyumlama öğrenim tekniği ile 20 eğitim döngüsü için eğitime devam edilmiştir. Uçtan-Uca Rota yönteminde zaman serisi işleme açısından LSTM ve GRU kullanılan versiyonlar tercih edilmiştir.

Conv1D ile geliştirilen versiyonda hata oranları gerçek hayat uygulamaları açısından kullanılamayacak seviyede gerçekleşmiştir. Hatanın yüksek çıkmasında: Nesne algılamada nesnenin çevreleyen kutu bilgisinin gerçek veriyle belirli oranda çakışmasıyla doğru kabul edilmesi nedeniyle nesnenin merkez noktasının doğru olarak tespit edilememesi, hız ve ivmelenme bilgilerinin de buna bağlı olarak belirli bir karakteristik gösterememesi ve örüntü yapılarına odaklanan Conv1D’nin ihtiyaç duyduğu verinin oluşmaması etkili olmuştur.

Tablo 4.8’de de görüleceği üzere, tez kapsamında geliştirilen Uçtan-Uca Rota yöntemi ADE ve FDE açısından LSTM kullanıldığında gelecekteki 5 sn için 0,92 m ve 2,02 m ve 8 sn için 1,65 m ve 3,97 m; GRU kullanıldığında 5 sn için 0,88 m ve 1,96 m ve 8 sn için 1,64 m ve 3,87 m hata miktarıyla rota tahmini gerçekleştirmiştir.

Gelecekteki 5 sn zaman diliminde araçlar için elde edilen sonuçlar Casas vd. tarafından yayınlanan sonuçlar ile karşılaştırıldığında ADE ve FDE açısından sırasıyla LSTM kullanıldığında %51 ve %52; GRU kullanıldığında %53 ve %54 kazanç sağladığı görülmüştür.

Gelecekteki 8 sn zaman diliminde tüm nesneler için elde edilen genel sonuçlar Najibi vd. tarafından yayınlanan sonuçlarla karşılaştırıldığında Uçtan-Uca Rota yönteminin ADE ve FDE açısından sırasıyla LSTM kullanıldığında %12 ve %17; GRU kullanıldığında %13 ve %19 kazanç sağladığı görülmüştür.

Tablo 4-8: Uçtan-Uca Rota yönteminin Deney seti 2 üzerinde gelecekteki 5 ve 8 sn için tahmin sonuçları. En iyi değerler kalın ve italik olarak ifade edilmiştir.

Tahmin Zaman Aralığı	Model	Zaman Serisi Modeli	ADE/FDE	Araç ADE/FDE	Yaya ADE/FDE	Bisikletli ADE/FDE
5sn	Casas vd. [25]			1,87/4,29		
	ROTA	LSTM	0,92/2,02	0,91/2,02	0,80/1,58	3,28/6,35
		GRU	0,88/1,96	0,87/1,96	0,76/1,52	3,29/6,29
8sn	Najibi vd. [24]		1,89/4,79			
	ROTA	LSTM	1,65/3,97	1,62/3,95	1,49/3,11	6,62/13,36
		GRU	1,64/3,87	1,62/3,86	1,36/2,80	6,44/12,52

ROTA modeline kıyasla Uçtan-Uca Rota yönteminin genel hata oranlarının arttığı gözlenmiştir. Bu artışa nesne algılama ve takip aşamalarındaki hataların, nesnenin merkez

konumunun tespitindeki sapmaların ve bu sapmaların hız ve ivmelenme hesaplarına yansımalarının etkisi olmuştur.

Elde edilen sonuçlar ışığında, Uçtan-Uca Rota yönteminin son zaman çalışmalarına kıyasla daha yüksek doğrulukla sonuç üretebildiği ve bu açıdan gerçek hayat uygulamalarında kullanılmaya aday olabileceği gösterilmiştir. Bununla birlikte, daha hassas sonuç üretebilen nesne algılama ve takip aşamaları gerçekleştirildiğinde, ROTA modelinin veri setlerindeki etiket verileriyle elde edebildiği sonuçlara ulaşılacaktır.



5. TARTIŞMA VE SONUÇ

Bu tez kapsamında, otonom sürüş teknolojilerinde kritik öneme sahip olan nesne algılama, takip ve rota tahmini süreçlerini uçtan uca bütünleştiren Uçtan-Uca Rota adında özgün bir yöntem önerilmiştir. Özellikle rota tahmini aşamasında, zamansal ve mekânsal etkileşimleri modelleyebilen yapay zekâ tabanlı ROTA adında yeni bir model geliştirilmiştir. Bu model, heterojen çizge temsili ile derin öğrenme bileşenlerinin entegrasyonu yoluyla hayata geçirilmiştir. Nesne algılama ve takip aşamaları, açık erişimli ve önceden eğitilmiş derin öğrenme modelleri aracılığıyla gerçekleştirilmiş olup, bu modellerin eğitimi çalışmanın kapsamı dışında bırakılmıştır.

Geliştirilen Uçtan-Uca Rota yöntemi, WAYMO veri seti üzerinde yapılan testlerde 8 saniyelik gelecek öngörülerini için ortalama 1,65 m ADE ve 3,92 m FDE değerleriyle rota tahmini yapabilmış, ADE ve FDE açısından sırasıyla %12,5 ve %18 kazanç sağlayarak mevcut yaklaşımlara kıyasla üstün başarı göstermiştir. Elde edilen bu başarı, yöntemin son aşamasında kullanılan ROTA modelinin yüksek temsil yeteneğine sahip dinamik heterojen çizgeleri derin grafik öğrenimi yöntemleriyle işleyerek nesnelerin hareket karakteristiğini zamansal-mekânsal ilişkide elde etmesine ve elde edilen bu verilerin rota tahminine etkisinin ağırlıklandırılarak kullanıldığı yapısına bağlıdır. Farklı zaman serisi modellemeleri ile desteklenen ROTA modeli, 0,03 sn cevap verme süresi ve 8 saniyelik gelecek öngörülerini için ortalama 0,99 m ADE ve 1,75 m FDE değerleri ile hem doğruluk hem de işlem süresi açısından önemli kazanımlar sağlamıştır. Model, özellikle WAYMO veri seti üzerinde yapılan testlerde, LiDAR sensörünün etiketli verileri kullanılarak gerçekleştirilen rota tahmininde hem ADE hem de FDE metriklerinde ortalama %47 oranında iyileşme sağlamıştır.

Modelin heterojen çizge yapısında 0,03 saniye gibi kısa sürede tahmin üretebilmesi, gerçek zamanlı uygulamalarda kullanılabilirliğini göstermesi açısından dikkat çekicidir. Ancak bu süre, yüksek performanslı özel donanımda ölçülmüş olup, gömülü sistemlerdeki karşılığı henüz değerlendirilmemiştir. Dolayısıyla, modelin enerji tüketimi, işlemci gücü ve bellek kullanımı gibi donanım kısıtlarını dikkate alarak optimize edilmesi, endüstriyel geçerliliğin artırılması için gereklidir.

Önerilen yaklaşımın heterojen çizge yapısı, nesneler arası etkileşimleri öğrenme açısından güçlü bir yapı sunmaktadır. Bununla birlikte, düğüm ve kenar çeşitliliğinin artmasıyla sistemin karmaşıklığı ve işlem süresi polinomsal olarak olumsuz yönde etkilenmektedir. Bu nedenle heterojen çizgelerin öz nitelik çıkarımı işlemlerinde kullanılan derin öğrenme yöntemlerinin

geliştirilmesine ihtiyaç duyulmaktadır. Buradaki kısıt nedeniyle düğüm türü sayısı üç ile sınırlandırılmıştır.

Modelin başarımı, büyük ölçüde kullanılan sensör verilerinin kalitesine bağlıdır. Gürültülü veya eksik LiDAR verileri, özellikle karmaşık ve yoğun trafik senaryolarında modelin doğruluğunu önemli ölçüde düşürebilmektedir. Bu nedenle, veri kalitesine duyarlı ön işleme adımlarının modele entegre edilmesi, kararlılığı ve genel başarımı artıracaktır.

Gelecek çalışmalar kapsamında, rota tahminindeki hataların azaltılması için nesne algılama, takip ve rota tahmini aşamalarının tek bir bütünleşik eğitim sürecine dahil edilmesi önerilebilir. Ayrıca, farklı aktivasyon ve kayıp fonksiyonlarının denenmesi, model katmanlarının yeniden yapılandırılması ve hiper parametrelerin optimizasyonu, tahmin doğruluğunu artıracak potansiyel araştırma alanları olarak değerlendirilmektedir. Bunun yanı sıra, işlem maliyetlerinin düşürülmesi amacıyla model mimarisinde sadeleştirme çalışmaları yapılmalı ve cevap üretim süresi daha da kısaltılmalıdır.

Veri setleri açısından değerlendirildiğinde, mevcut veri setlerinin ağırlıklı olarak araç ve yayaları içermesi, modellerin nadir nesne türleri (örneğin bisikletli, hayvan vb.) üzerindeki genelleme yeteneğini sınırlamaktadır. Bu durum, tez kapsamında en yüksek hata oranlarının bisikletlilerde gözlemlenmesine neden olmuştur. Bu bağlamda, veri setlerinin daha dengeli ve çeşitliliği yüksek örnekler içerecek şekilde yeniden düzenlenmesi önem arz etmektedir. Ayrıca veri setleri oluşturulurken hava şartları ve ortamdaki ışık düzeyleri açısından da homojen bir dağılım bulunmamaktadır. Örneğin WAYMO veri seti genellikle güneşli havalarda ve gün ışığı olduğu zaman dilimlerindeki verileri kapsamaktadır. Bu durum geliştirilen modellerin her türlü çevresel şartları genellemesini kısıtlamaktadır.

Ayrıca, LiDAR sensörünün yüksek maliyetinden dolayı daha ekonomik çözümler sunabilecek stereo kamera tabanlı sistemlerin araştırılması, alternatif yönelimler arasında değerlendirilebilir. Bunun için, stereo kameralarla elde edilen derinlik bilgilerini içeren geniş ölçekli ve etiketli veri kümelerinin oluşturulmasına ihtiyaç duyulmaktadır.

Sonuç olarak, bu tez çalışması, otonom araçlar için güvenli sürüş kararlarını destekleyebilecek, uçtan uca çalışan özgün bir rota tahmin yöntemi sunmaktadır. Literatürdeki öncü yaklaşımlarla yapılan karşılaştırmalar, geliştirilen modelin hem doğruluk hem de işlem süresi açısından rekabetçi bir performans sergilediğini göstermektedir.

6. KAYNAKLAR

- [1] http://www.odd.org.tr/folders/2837/categorial1docs/2528/otomotiv_tuketicileri_arastirmas_i.pdf (Erişim tarihi: 23.12. 2024)
- [2] Badue, C., Guidolini, R., Carneiro, R. V., Azevedo, P., Cardoso, V. B., Forechi, A., Jesus, L., Berriel, R., Paixão, T. M., Mutz, F., de Paula Veronese, L., Oliveira-Santos, T., & De Souza, A.F. (2021). Self-driving cars: A survey. *Expert Syst Appl*, 165,113816. Doi: 10.1016/j.eswa.2020.113816.
- [3] Othman, K. (2021). Public acceptance and perception of autonomous vehicles: a comprehensive review. *AI and Ethics*, 1(3), 355-387. Doi: 10.1007/s43681-021-00041-8.
- [4] Rahman, Md. M., & Thill, J.C. (2023). What Drives People's Willingness to Adopt Autonomous Vehicles? A Review of Internal and External Factors. *Sustainability*, 15 (15), 11541. Doi: 10.3390/su151511541.
- [5] Brown, B. (2017). The Social Life of Autonomous Cars. *Computer* (Long Beach Calif), 50(2), 92-96. Doi: 10.1109/MC.2017.59.
- [6] Martinez-Ríos, E. A., Bustamante-Bello, M. R., & Arce-Sáenz, L. A. (2022). A Review of Road Surface Anomaly Detection and Classification Systems Based on Vibration-Based Techniques. *Applied Sciences*, 12(19), 9413. Doi: 10.3390/app12199413.
- [7] Gautam, S., & Kumar, A. (2023). Image-based automatic traffic lights detection system for autonomous cars: a review. *Multimed Tools Appl*, 82(17), 26135-26182. Doi: 10.1007/s11042-023-14340-1.
- [8] Triki, N., Karray, M., & Ksantini, M. (2024). A Comprehensive Survey and Analysis of Traffic Sign Recognition Systems With Hardware Implementation. *IEEE Access*, 12, 144069-144081. Doi: 10.1109/ACCESS.2024.3459708.
- [9] Kumar, D., & Muhammad, N. (2023). A Survey on Localization for Autonomous Vehicles. *IEEE Access*, 11, 115865-115883. Doi: 10.1109/ACCESS.2023.3326069.
- [10] Reda, M., Onsy, A., Haikal, A. Y., & Ghanbari, A. (2024). Path planning algorithms in the autonomous driving system: A comprehensive review. *Rob Auton Syst*, 174, 104630. Doi: 10.1016/j.robot.2024.104630.
- [11] Durlik, I., Miller, T., Kostecka, E., Zwierzewicz, Z., & Łobodzińska, A. (2024). Cybersecurity in Autonomous Vehicles—Are We Ready for the Challenge?. *Electronics* (Basel), 13(13), 2654. Doi: 10.3390/electronics13132654.

- [12] Alaba, S. Y., Gurbuz, A. C., & Ball, J. E. (2024). Emerging Trends in Autonomous Vehicle Perception: Multimodal Fusion for 3D Object Detection. *World Electric Vehicle Journal*, 15(1), 20. Doi: 10.3390/wevj15010020.
- [13] Gulzar, M., Muhammad, Y., & Muhammad, N. (2021). A Survey on Motion Prediction of Pedestrians and Vehicles for Autonomous Driving. *IEEE Access*, 9, 137957-137969. Doi: 10.1109/ACCESS.2021.3118224.
- [14] Chen, L., Wu, P., Chitta, K., Jaeger, B., Geiger, A., & Li, H. (2023). End-to-end Autonomous Driving: Challenges and Frontiers. *Arxiv*: <http://arxiv.org/abs/2306.16927>
- [15] Hecht, J. (2018). Lidar for Self-Driving Cars. *Optics and Photonics News*, 29(1), 28-33. Doi: 10.1364/OPN.29.1.000026.
- [16] Aijazi, A., Checchin, P., & Trassoudaine, L. (2013). Segmentation Based Classification of 3D Urban Point Clouds: A Super-Voxel Based Approach with Evaluation. *Remote Sens (Basel)*, 5(4), 1624-1650. Doi: 10.3390/rs5041624.
- [17] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436-444. Doi: 10.1038/nature14539.
- [18] Bera, A., Kim, S., Randhavane, T., Pratapa, S., & Manocha, D. (2016). GLMP- realtime pedestrian path prediction using global and local movement patterns. 2016 IEEE International Conference on Robotics and Automation (ICRA), 5528-5535. Doi: 10.1109/ICRA.2016.7487768.
- [19] Keller, C. G., & Gavrila, D. M. (2014). Will the Pedestrian Cross? A Study on Pedestrian Path Prediction. *IEEE Transactions on Intelligent Transportation Systems*, 15(2), 494-506. Doi: 10.1109/TITS.2013.2280766.
- [20] Hariyono, J., Shahbaz, A., & Jo, K.-H. (2015). Estimation of walking direction for pedestrian path prediction from moving vehicle. 2015 IEEE/SICE International Symposium on System Integration (SII), 750-753. Doi: 10.1109/SII.2015.7405073.
- [21] Alahi, A., Goel, K., Ramanathan, V., Robicquet, A., Fei-Fei, L., & Savarese, S. (2016). Social LSTM: Human Trajectory Prediction in Crowded Spaces. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 961-971. Doi: 10.1109/CVPR.2016.110.
- [22] Elman, J. (1990). Finding structure in time. *Cogn Sci*, 14(2), 179-211. Doi: 10.1016/0364-0213(90)90002-E.

- [23] Xu, Y., Zablocki, É., Boulch, A., Puy, G., Chen, M., Bartoccioni, F., Samet, N., Siméoni, O., Gidaris, S., Vu, T.-H., Bursuc, A., Valle, E., Marlet, R., & Cord, M. (2024). Valeo4Cast: A Modular Approach to End-to-End Forecasting. Arxiv: <http://arxiv.org/abs/2406.08113>
- [24] Najibi, M., Ji, J., Zhou, Y., Qi, C. R., Yan, X., Ettinger, S., & Anguelov, D. (2022). Motion Inspired Unsupervised Perception and Prediction in Autonomous Driving. Arxiv: <http://arxiv.org/abs/2210.08061>
- [25] Casas, S., Agro, B., Mao, J., Gilles, T., Cui, A., Li, T., & Urtasun, R. (2024). DeTra: A Unified Model for Object Detection and Trajectory Forecasting. Arxiv: <http://arxiv.org/abs/2406.04426>
- [26] Mao, J., Shi, S., Wang, X., & Li, H. (2023). 3D Object Detection for Autonomous Driving: A Comprehensive Survey. *Int J Comput Vis*, 131(8), 1909-1963. Doi: 10.1007/s11263-023-01790-1.
- [27] Zhang, P., Li, X., He, L., & Lin, X. (2023). 3D Multiple Object Tracking on Autonomous Driving: A Literature Review. Arxiv: <http://arxiv.org/abs/2309.15411>
- [28] Madjid, N. A., Ahmad, A., Mebrahtu, M., Babaa, Y., Nasser, A., Malik, S., Hassan, B., Werghi, N., Dias, J., & Khonji, M. (2025). Trajectory Prediction for Autonomous Driving: Progress, Limitations, and Future Directions. Arxiv: <http://arxiv.org/abs/2503.03262>
- [29] Yang, B., Luo, W., & Urtasun, R. (2018). PIXOR: Real-time 3D Object Detection from Point Clouds. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 7652-7660. Doi: 10.1109/CVPR.2018.00798.
- [30] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 770-778. Doi: 10.1109/CVPR.2016.90.
- [31] <https://www.vision.rwth-aachen.de/page/mots> (Erişim Tarihi: 21.12.2024)
- [32] Meyer, G. P., Laddha, A., Kee, E., Vallespi-Gonzalez, C., & Wellington, C. K. (2019). LaserNet: An Efficient Probabilistic 3D Object Detector for Autonomous Driving. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 12669-12678. Doi: 10.1109/CVPR.2019.01296.
- [33] Yin, T., Zhou, X., & Krahenbuhl, P. (2021). Center-based 3D Object Detection and Tracking. 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 11779-11788. Doi: 10.1109/CVPR46437.2021.01161.

- [34] Zhou, Y., & Tuzel, O. (2018). VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 4490-4499. Doi: 10.1109/CVPR.2018.00472.
- [35] Duan, K., Bai, S., Xie, L., Qi, H., Huang, Q., & Tian, Q. (2019). CenterNet: Keypoint Triplets for Object Detection. 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 6568-6577. Doi: 10.1109/ICCV.2019.00667.
- [36] Caesar, H., Bankiti, V., Lang, A. H., Vora, S., Liong, V. E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., & Beijbom, O. (2019). nuScenes: A multimodal dataset for autonomous driving. Arxiv: <http://arxiv.org/abs/1903.11027>
- [37] Sun, P., Kretzschmar, H., Dotiwalla, X., Chouard, A., Patnaik, V., Tsui, P., Guo, J., Zhou, Y., Chai, Y., Caine, B., Vasudevan, V., Han, W., Ngiam, J., Zhao, H., Timofeev, A., Ettinger, S., Krivokon, M., Gao, A., Joshi, A., Zhao, S., Cheng, S., Zhang, Y., Shlens, J., Chen, Z., & Anguelov, D. (2020). Scalability in Perception for Autonomous Driving: Waymo Open Dataset. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2443-2451. Doi: 10.1109/CVPR42600.2020.00252.
- [38] Fan, L., Xiong, X., Wang, F., Wang, N., & Zhang, Z. (2021). RangeDet: In Defense of Range View for LiDAR-based 3D Object Detection. 2021 IEEE/CVF International Conference on Computer Vision (ICCV), 2898-2907. Doi: 10.1109/ICCV48922.2021.00291.
- [39] Lin, T.-Y., Dollar, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. (2017). Feature Pyramid Networks for Object Detection. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 936-944. Doi: 10.1109/CVPR.2017.106.
- [40] <https://waymo.com/waymo-driver/> (Erişim Tarihi: 15.12.2023)
- [41] Sun, P., Wang, W., Chai, Y., Elsayed, G., Bewley, A., Zhang, X., Sminchisescu, C., & Anguelov, D. (2021). RSN: Range Sparse Net for Efficient, Accurate LiDAR 3D Object Detection. 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 5721-5730. Doi: 10.1109/CVPR46437.2021.00567.
- [42] Hu, J. S. K., Kuai, T., & Waslander, S. L. (2022). Point Density-Aware Voxels for LiDAR 3D Object Detection. 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 8459-8468. Doi : 10.1109/CVPR52688.2022.00828.
- [43] Girshick, R. (2015). Fast R-CNN. Proceedings of the IEEE International Conference on Computer Vision (ICCV), Arxiv: <http://arxiv.org/abs/1504.08083>

- [44] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention Is All You Need. Arxiv: <http://arxiv.org/abs/1706.03762>
- [45] Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, 323(6088), 533-536. Doi: 10.1038/323533a0.
- [46] Huang, Y., Zhou, J., Li, X., Dong, Z., Xiao, J., Wang, S., & Zhang, H. (2023). MENet: Map-enhanced 3D object detection in bird's-eye view for LiDAR point clouds. *International Journal of Applied Earth Observation and Geoinformation*, 120, 103337. Doi: 10.1016/j.jag.2023.103337.
- [47] Chen, Y., Liu, J., Zhang, X., Qi, X., & Jia, J. (2023). VoxelNeXt: Fully Sparse VoxelNet for 3D Object Detection and Tracking. 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 21674-21683. Doi: 10.1109/CVPR52729.2023.02076.
- [48] Wang, J., Liu, Y., Zhu, Y., Wang, D., & Zhang, Y. (2024). 3D Point Cloud Object Detection Method Based on Multi-Scale Dynamic Sparse Voxelization. *Sensors*, 24(6), 1804. Doi: 10.3390/s24061804.
- [49] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., & Berg, A. C. (2015). SSD: Single Shot MultiBox Detector. *Computer Vision- ECCV 2016*, 21-37. Doi: 10.1007/978-3-319-46448-0_2.
- [50] Qian, R., Garg, D., Wang, Y., You, Y., Belongie, S., Hariharan, B., Campbell, M., Weinberger, K. Q., & Chao, W.L. (2020). End-to-End Pseudo-LiDAR for Image-Based 3D Object Detection. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 5880-5889. Doi: 10.1109/CVPR42600.2020.00592.
- [51] You, Y., Wang, Y., Chao, W.-L., Garg, D., Pleiss, G., Hariharan, B., Campbell, M., & Weinberger, K. Q. (2019). Pseudo-LiDAR++: Accurate Depth for 3D Object Detection in Autonomous Driving. Arxiv: <http://arxiv.org/abs/1906.06310>
- [52] Shi, S., Wang, X., & Li, H. (2019). PointRCNN: 3D Object Proposal Generation and Detection From Point Cloud. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 770-779. Doi: 10.1109/CVPR.2019.00086.
- [53] Luo, C., Yang, X., & Yuille, A. (2021). Exploring Simple 3D Multi-Object Tracking for Autonomous Driving. 2021 IEEE/CVF International Conference on Computer Vision (ICCV), 10468-10477. Doi: 10.1109/ICCV48922.2021.01032.
- [54] Kim, A., Osep, A., & Leal-Taixe, L. (2021). EagerMOT: 3D Multi-Object Tracking via Sensor Fusion. 2021 IEEE International Conference on Robotics and Automation (ICRA), 11315-11321. Doi: 10.1109/ICRA48506.2021.9562072.

- [55] He, K., Gkioxari, G., Dollar, P., & Girshick, R. (2017). Mask R-CNN. 2017 IEEE International Conference on Computer Vision (ICCV), 2980-2988. Doi: 10.1109/ICCV.2017.322.
- [56] Kalman, R. E. (1960). A New Approach to Linear Filtering and Prediction Problems. *Journal of Basic Engineering*, 828(1), 35-45. Doi: 10.1115/1.3662552.
- [57] Jungnickel, D. (1999). The Greedy Algorithm. *Graphs, Networks and Algorithms*, 129-153. Doi: 10.1007/978-3-662-03822-2_5.
- [58] Chiu, H.-K., Li, J., Ambrus, R., & Bohg, J. (2021). Probabilistic 3D Multi-Modal, Multi-Object Tracking for Autonomous Driving. 2021 IEEE International Conference on Robotics and Automation (ICRA), 14227-14233. Doi: 10.1109/ICRA48506.2021.9561754.
- [59] Mahalanobis, P. C. (1936). On the Generalized Distance in Statistics. *Proceedings of the National Institute of Science of India*, 49-55.
- [60] Zhang, C., Zhang, C., Guo, Y., Chen, L., & Happold, M. (2023). MotionTrack: End-to-End Transformer-based Multi-Object Tracking with LiDAR-Camera Fusion. 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 151-160. Doi: 10.1109/CVPRW59228.2023.00020.
- [61] Wang, X., Qi, S., Zhao, J., Zhou, H., Zhang, S., Wang, G., Tu, K., Guo, S., Zhao, J., Li, J., & Yang, M. (2024). MCTrack: A Unified 3D Multi-Object Tracking Framework for Autonomous Driving. Arxiv: <http://arxiv.org/abs/2409.16149>
- [62] Kuhn, H. W. (1955). The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1-2), 83-97. Doi: 10.1002/nav.3800020109.
- [63] Ma, Y., Zhu, X., Zhang, S., Yang, R., Wang, W., & Manocha, D. (2019). TrafficPredict: Trajectory Prediction for Heterogeneous Traffic-Agents. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(1), 6120-6127. Doi: 10.1609/aaai.v33i01.33016120.
- [64] Chandra, R., Bhattacharya, U., Bera, A., & Manocha, D. (2019). TraPHic: Trajectory Prediction in Dense and Heterogeneous Traffic Using Weighted Interactions. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 8475-8484. Doi: 10.1109/CVPR.2019.00868.
- [65] Jain, A., Casas, S., Liao, R., Xiong, Y., Feng, S., Segal, S., & Urtasun, R. (2019). Discrete Residual Flow for Probabilistic Pedestrian Behavior Prediction. Arxiv: <https://arxiv.org/abs/2409.16149>

- [66] Li, X., Ying, X., & Chuah, M. C. (2019). GRIP++: Enhanced Graph-based Interaction-aware Trajectory Prediction for Autonomous Driving. Arxiv: <http://arxiv.org/abs/1907.07792>
- [67] Kipf, T. N., & Welling, M. (2016). Semi-Supervised Classification with Graph Convolutional Networks. Arxiv: <http://arxiv.org/abs/1609.02907>
- [68] Bai, S., Kolter, J. Z., & Koltun, V. (2018). An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling. Arxiv: <http://arxiv.org/abs/1803.01271>
- [69] Cho, K., Merrienboer, B. V., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. Arxiv: <http://arxiv.org/abs/1406.1078>
- [70] Cheng, H., Liao, W., Tang, X., Yang, M. Y., Sester, M., & Rosenhahn, B. (2020). Exploring Dynamic Context for Multi-path Trajectory Prediction. Arxiv: <http://arxiv.org/abs/2010.16267>
- [71] Sohn, K., Yan, X., & Lee, H. (2015). Learning structured output representation using deep conditional generative models. *Proceedings of the 28th International Conference on Neural Information Processing Systems*, 2, 3483-3491.
- [72] Bock, J., Krajewski, R., Moers, T., Runde, S., Vater, L., & Eckstein, L. (2020). The inD Dataset: A Drone Dataset of Naturalistic Road User Trajectories at German Intersections. 2020 IEEE Intelligent Vehicles Symposium (IV), 1929-1934. Doi: 10.1109/IV47402.2020.9304839.
- [73] Zhang, Z., Gao, J., Mao, J., Liu, Y., Anguelov, D., & Li, C. (2020). STINet: Spatio-Temporal-Interactive Network for Pedestrian Detection and Trajectory Prediction. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 11343-11352. Doi: 10.1109/CVPR42600.2020.01136.
- [74] Lang, A. H., Vora, S., Caesar, H., Zhou, L., Yang, J., & Beijbom, O. (2019). PointPillars: Fast Encoders for Object Detection From Point Clouds. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 12689-12697. Doi: 10.1109/CVPR.2019.01298.
- [75] Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. Arxiv: <http://arxiv.org/abs/1505.04597>

- [76] Sardoğan, M., Özen, Y., & Tuncer, A. (2020). Detection of Apple Leaf Diseases using Faster R-CNN. *Düzce Üniversitesi Bilim ve Teknoloji Dergisi*, 8(1), 1110-1117. Doi: 10.29130/dubited.648387.
- [77] Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., & Bengio, Y. (2017). Graph Attention Networks. Arxiv: <http://arxiv.org/abs/1710.10903>
- [78] Zhang, K., Feng, X., Wu, L., & He, Z. (2022). Trajectory Prediction for Autonomous Driving Using Spatial-Temporal Graph Attention Transformer. *IEEE Transactions on Intelligent Transportation Systems*, 23(11), 22343-22353. Doi: 10.1109/TITS.2022.3164450.
- [79] Houston, J., Zuidhof, G., Bergamini, L., Ye, Y., Chen, L., Jain, A., Omari, S., Iglovikov, V., & Ondruska, P. (2020). One Thousand and One Hours: Self-driving Motion Prediction Dataset. Arxiv: <http://arxiv.org/abs/2006.14480>
- [80] Mo, X., Huang, Z., Xing, Y., & Lv, C. (2022). Multi-Agent Trajectory Prediction With Heterogeneous Edge-Enhanced Graph Attention Network. *IEEE Transactions on Intelligent Transportation Systems*, 23(7), 9554-9567. Doi: 10.1109/TITS.2022.3146300.
- [81] Zhan, W., Sun, L., Wang, D., Shi, H., Clausse, A., Naumann, M., Kummerle, J., Konigshof, H., Stiller, C., Fortelle, A. D. L., & Tomizuka, M. (2019). INTERACTION Dataset: An INTERnational, Adversarial and Cooperative moTION Dataset in Interactive Driving Scenarios with Semantic Maps. Arxiv: <http://arxiv.org/abs/1910.03088>
- [82] Li, X., Pontes, J. K., & Lucey, S. (2021). Neural Scene Flow Prior. Arxiv: <http://arxiv.org/abs/2111.01253>
- [83] Varadarajan, B., Hefny, A., Srivastava, A., Refaat, K. S., Nayakanti, N., Cornman, A., Chen, K., Douillard, B., Lam, C. P., Anguelov, D., & Sapp, B. (2021). MultiPath++: Efficient Information Fusion and Trajectory Aggregation for Behavior Prediction. Arxiv: <http://arxiv.org/abs/2111.14973>
- [84] Zhang, K., Zhao, L., Dong, C., Wu, L., & Zheng L. (2023). AI-TP: Attention-Based Interaction-Aware Trajectory Prediction for Autonomous Driving. *IEEE Transactions on Intelligent Vehicles*, 8(1), 73-83. Doi: 10.1109/TIV.2022.3155236.
- [85] Xu, D., Shang, X., Peng, H., & Li, H. (2023). MVHGN: Multi-View Adaptive Hierarchical Spatial Graph Convolution Network Based Trajectory Prediction for Heterogeneous Traffic-Agents. *IEEE Transactions on Intelligent Transportation Systems*, 24(6), 6217-6226. Doi: 10.1109/TITS.2023.3248090.

- [86] Jia, X., Wu, P., Chen, L., Liu, Y., Li, H., & Yan, J. (2023). HDGT: Heterogeneous Driving Graph Transformer for Multi-Agent Trajectory Prediction via Scene Encoding. *IEEE Trans Pattern Anal Mach Intell*, 45(11), 13860-13875. Doi: 10.1109/TPAMI.2023.3298301.
- [87] Hu, Z., Dong, Y., Wang, K., & Sun, Y. (2020). Heterogeneous Graph Transformer. Arxiv: <http://arxiv.org/abs/2003.01332>
- [88] Ettinger, S., Cheng, S., Caine, B., Liu, C., Zhao, H., Pradhan, S., Chai, Y., Sapp, B., Qi, C., Zhou, Y., Yang, Z., Chouard, A., Sun, P., Ngiam, J., Vasudevan, V., McCauley, A., Shlens, J., & Anguelov, D. (2021). Large Scale Interactive Motion Forecasting for Autonomous Driving : The Waymo Open Motion Dataset. 2021 IEEE/CVF International Conference on Computer Vision (ICCV), 9690-9699. Doi: 10.1109/ICCV48922.2021.00957.
- [89] Zeng, X., Gao, M., He, Z., & Yang, Y. (2023). Trajectory Prediction for Surrounding Traffic Participants via Local Perception and Attentive Map Encoding. 2023 IEEE International Conference on Robotics and Biomimetics (ROBIO), 1-6. Doi: 10.1109/ROBIO58561.2023.10354998.
- [90] Tang, L., Yan, F., Zou, B., Li, W., Lv, C., & Wang, K. (2023). Trajectory prediction for autonomous driving based on multiscale spatial-temporal graph. *IET Intelligent Transport Systems*, 17(2), 386-399. Doi: 10.1049/itr2.12265.
- [91] Amin, F., Gharami, K., & Sen, B. (2024). TrajectoFormer: Transformer-Based Trajectory Prediction of Autonomous Vehicles with Spatio-temporal Neighborhood Considerations. *International Journal of Computational Intelligence Systems*, 17(1), 87. Doi: 10.1007/s44196-024-00410-1.
- [92] <https://datahub.transportation.gov/stories/s/Next-Generation-Simulation-NGSIM-Open-Data/i5zb-xe34/> (Erişim Tarihi: 22.11.2024)
- [93] Awan, M., Shin, J., & Whangbo, T. K. (2024). Trajectory Prediction of Heterogeneous Traffic Agents With Collision Vigilance and Avoidance. *IEEE Transactions on Intelligent Vehicles*, 9(1), 93-102. Doi: 10.1109/TIV.2023.3293088.
- [94] Schlichtkrull, M., Kipf, T. N., Bloem, P., van den Berg, R., Titov, I., & Welling, M. (2017). Modeling Relational Data with Graph Convolutional Networks. Arxiv: <http://arxiv.org/abs/1703.06103>
- [95] Watkins, C. (1989). Learning from delayed rewards. Doktora Tezi, King's College, Londra.
- [96] Wilson, B., Qi, W., Agarwal, T., Lambert, J., Singh, J., Khandelwal, S., Pan, B., Kumar, R., Hartnett, A., Pontes, J. K., Ramanan, D., Carr, P., & Hays, J. (2023). Argoverse 2: Next

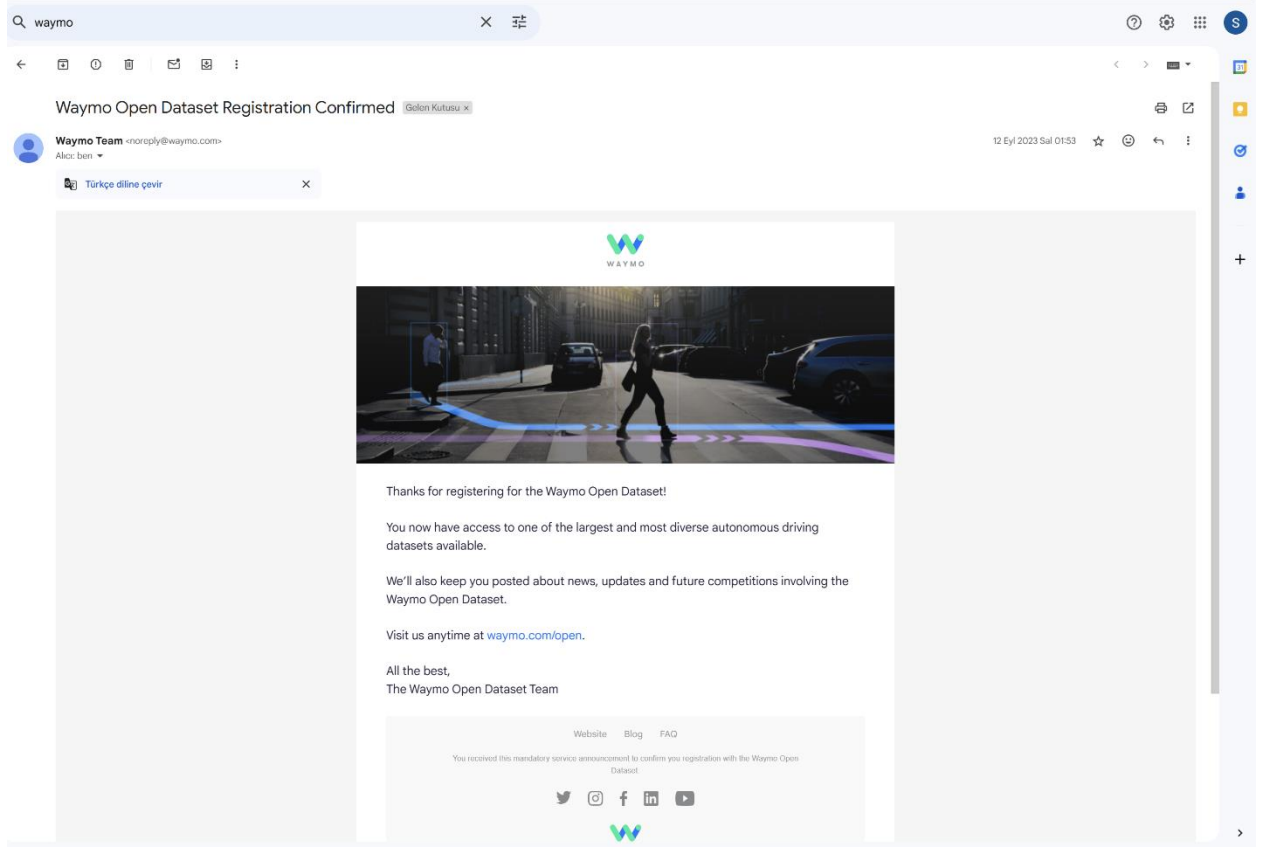
- Generation Datasets for Self-Driving Perception and Forecasting. Arxiv: <http://arxiv.org/abs/2301.00493>
- [97] <https://www.autopilotreview.com/cars-with-autopilot-self-driving/> (Erişim Tarihi: 19.11.2024)
- [98] <https://www.webtekno.com/otonom-araclar-neden-cok-fazla-sensore-ihtiyac-duyarlar-h31657.html> (Erişim Tarihi: 14.12.2024)
- [99] Fawole, O. A., & Rawat, D. B. (2024). Recent Advances in 3D Object Detection for Self-Driving Vehicles: A Survey. *AI*, 5(3), 1255-1285. Doi: 10.3390/ai5030061.
- [100] Royo, S., & Ballesta-Garcia, M. (2019). An Overview of Lidar Imaging Systems for Autonomous Vehicle. *Applied Sciences*, 9(19), 4093. Doi: 10.3390/app9194093.
- [101] <https://erc-bpgc.github.io/handbook/electronics/Sensors/lidar/> (Erişim Tarihi: 05.05.2024)
- [102] <https://www.intelligent-mobility-xperience.com/lidar--future-technology-for-autonomous-vehicles-d-43215/> (Erişim Tarihi: 14.12.2024)
- [103] <https://waymo.com/blog/2022/09/informing-smarter-lidar-solutions-> (Erişim Tarihi: 14.12.2023)
- [104] <https://www.fiercееlectronics.com/components/lidar-vs-radar> (Erişim Tarihi: 15.12.2023)
- [105] Liu, Li., Ouyang, W., Wang, X., Fieguth, P., Chen, Jie., Liu, X., & Pietikäinen, M. (2020). Deep Learning for Generic Object Detection: A Survey. *Int J Comput Vis*, 128(2), 261-318. Doi: 10.1007/s11263-019-01247-4.
- [106] Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2013). Rich feature hierarchies for accurate object detection and semantic segmentation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Arxiv: <http://arxiv.org/abs/1311.2524>
- [107] Ren, S., He, K., Girshick, R., & Sun, J. (2017). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans Pattern Anal Mach Intell*, 39(6), 1137-1149. Doi: 10.1109/TPAMI.2016.2577031.
- [108] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2026). You Only Look Once: Unified, Real-Time Object Detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 779-788.
- [109] Simon, M., Milz, S., Amende, K., & Gross, H.-M. (2018). Complex-YOLO: Real-time 3D Object Detection on Point Clouds. Arxiv: <http://arxiv.org/abs/1803.06199>

- [110] Liu, S., & Deng, W. (2015). Very deep convolutional neural network based image classification using small training sample size. 2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR), 730-734. Doi: 10.1109/ACPR.2015.7486599.
- [111] Law, H., & Deng, J. (2018). CornerNet: Detecting Objects as Paired Keypoints. Proceedings of the European Conference on Computer Vision (ECCV), 734-750.
- [112] Newell, A., Yang, K., & Deng, J. (2016). Stacked Hourglass Networks for Human Pose Estimation. Arxiv: <http://arxiv.org/abs/1603.06937>
- [113] Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., & Zagoruyko, S. (2020). End-to-End Object Detection with Transformers. Arxiv: <http://arxiv.org/abs/2005.12872>
- [114] Zhu, Y., Xu, R., Tao, C., An, H., Wang, H., Sun, Z., & Lu, K. (2024). DS-Trans: A 3D Object Detection Method Based on a Deformable Spatiotemporal Transformer for Autonomous Vehicles. Remote Sens (Basel), 16(9), 1621. Doi: 10.3390/rs16091621.
- [115] Bernardin, K., & Stiefelhagen, R. (2008). Evaluating Multiple Object Tracking Performance: The CLEAR MOT Metrics. EURASIP J Image Video Process, 2008,1-10. Doi: 10.1155/2008/246309.
- [116] Ristani, E., Solera, F., Zou, R. S., Cucchiara, R., & Tomasi, C. (2016). Performance Measures and a Data Set for Multi-Target, Multi-Camera Tracking. Arxiv: <https://arxiv.org/abs/1609.01775>
- [117] Bewley, A., Ge, Z., Ott, L., Ramos, F., & Upcroft, B. (2016). Simple online and realtime tracking. 2016 IEEE International Conference on Image Processing (ICIP), 3464-3468. Doi: 10.1109/ICIP.2016.7533003.
- [118] Rudenko, A., Palmieri, L., Herman, M., Kitani, K. M., Gavrila, D. M., & Arras, K. O. (2019). Human Motion Trajectory Prediction: A Survey. The International Journal of Robotics Research, 39(8), 895-935. Doi: 10.1177/0278364920917446.
- [119] Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. Neural Computation, 9(8), 1735-1780. Doi: 10.1162/neco.1997.9.8.1735.
- [120] Shi, C., Wang, X., & Yu, P. S. (2022). Heterogeneous Graph Representation Learning and Applications. Artificial Intelligence: Foundations, Theory, and Algorithms. Doi: 10.1007/978-981-16-6166-2.
- [121] Wang, M., Zheng, D., Ye, Z., Gan, Q., Li, M., Song, X., Zhou, J., Ma, C., Yu, L., Gai, Y., Xiao, T., He, T., Karypis, G., Li, J., & Zhang, Z. (2019). Deep Graph Library: A Graph-Centric, Highly-Performant Package for Graph Neural Networks. Arxiv: <http://arxiv.org/abs/1909.01315>

- [122] Stamile, C., Marzullo, A., & Deusebio, E. (2021). Graph Machine Learning. Packt Publishing Ltd. Birmingham.
- [123] Ju, W., Fang, Z., Gu, Y., Liu, Z., Long, Q., Qiao, Z., Qin, Y., Shen, J., Sun, F., Xiao, Z., Yang, J., Yuan, J., Zhao, Y., Wang, Y., Luo, X., & Zhang, M. (2024). A Comprehensive Survey on Deep Graph Representation Learning. *Neural Networks*, 173, 106207. doi: 10.1016/j.neunet.2024.106207.
- [124] Yang, Z., Zhang, G., Wu, J., Yang, J., Sheng, Q. Z., Xue, S., Zhou, C., Aggarwal, C., Peng, H., Hu, W., Hancock, E., & Pietro, L. (2023). State of the Art and Potentialities of Graph-level Learning. Arxiv: <http://arxiv.org/abs/2301.05860>
- [125] Tektaş, M., & Tektaş, N. (2018). Meslek Matematiği. Hiperyayın. İstanbul.
- [126] Kingma, D. P., & Ba, J. (2014). Adam: A Method for Stochastic Optimization. Arxiv: <http://arxiv.org/abs/1412.6980>
- [127] <https://github.com/kittyschulz/Exploring-Waymo-Open-Dataset?tab=readme-ov-file> (Erişim Tarihi: 15.05.2024)
- [128] <https://argoverse.github.io/user-guide/datasets/sensor.html#sensor-dataset-splits> (Erişim Tarihi: 05.04.2025)
- [129] Wang, Z., Chen, F., Lertniphonphan, K., Chen, S., Bao, J., Zheng, P., Zhang, J., Huang, K., & Zhang, T. (2023). Technical Report for Argoverse Challenges on Unified Sensor-based Detection, Tracking, and Forecastin. Arxiv: <http://arxiv.org/abs/2311.15615>
- [130] Hu, Y., Yang, J., Chen, L., Li, K., Sima, C., Zhu, X., Chai, S., Du, S., Lin, T., Wang, W., Lu, L., Jia, X., Liu, Q., Dai, J., Qiao, Y., & Li, H. (2022). Planning-oriented Autonomous Driving. Arxiv: <http://arxiv.org/abs/2212.10156>
- [131] Shi, S., Jiang, L., Dai, D., & Schiele, B (2022). Motion Transformer with Global Intention Localization and Local Movement Refinement. Arxiv: <http://arxiv.org/abs/2209.13508>

7. EKLER

EK 1. WAYMO Kullanıcı Aktifleştirme E-postası



ÖZGEÇMİŞ

Kişisel Bilgiler		
Adı Soyadı	Sevcan TURAN	
Doğum Yeri		
Doğum Tarihi		
Uyruğu		
Telefon		
E-Posta Adresi	sevcanturan@comu.edu.tr	
Web Adresi		

foto

	Eğitim Bilgileri
	Lisans
Üniversite	Bahçeşehir Üniversitesi
Fakülte	Mühendislik Fakültesi
Bölümü	Bilgisayar Mühendisliği

	Yüksek Lisans
Üniversite	Çanakkale Onsekiz Mart Üniversitesi
Enstitü Adı	Fen Bilimleri Enstitüsü
Anabilim Dalı	Bilgisayar Mühendisliği
Programı	Bilgisayar Mühendisliği

Makale ve Bildiriler
AdaBelief Optimizasyon Tekniğinin Derin Öğrenmeye Dayalı Yaya Rotası Tahmin Uygulamalarına Etkisinin “Yakınsama” Açısından İncelenmesi. Turan, S., & Temurtaş, F. (2024). Karamanoğlu Mehmetbey Üniversitesi Mühendislik ve Doğa Bilimleri Dergisi, 6(1), 1-10. https://doi.org/10.55213/kmujens.1418280
Different Application Areas of Object Detection With Deep Learning. Turan, S., Milani, B., & Temurtaş, F. (2021). Akıllı Ulaşım Sistemleri ve Uygulamaları Dergisi, 4(2), 148-164. https://doi.org/10.51513/jitsa.957371
A Hadoop Application for Urban Computing in Smart City. Turan, S., & Öztekin, H. (2020). Electronic Letters on Science and Engineering, 16(2), 193-201.
Elektrikli bisiklet paylaşımı: Çanakkale Onsekiz Mart Üniversitesi Terzioğlu Yerleşkesi örneği. Turan, S., & Efe, S. B. (2020). Akıllı Ulaşım Sistemleri ve Uygulamaları Dergisi, 3(2), 112-122.