

**T.C.
SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**AKILLI GÖRSEL GÖZETLEME SİSTEMİNE SAHİP OTOMATİK
YÖNLENDİRMELİ HASTANE ARACI**

YÜKSEK LİSANS TEZİ

Şerife ÇELİKBAŞ

Enstitü Anabilim Dalı : BİYOMEDİKAL MÜHENDİSLİĞİ
Tez Danışmanı : Dr. Öğr. Üyesi Ayla EREN ÖZDEMİR

Ağustos 2020

**T.C.
SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**AKILLI GÖRSEL GÖZETLEME SİSTEMİNE SAHİP OTOMATİK
YÖNLENDİRMELİ HASTANE ARACI**

YÜKSEK LİSANS TEZİ

Şerife ÇELİKBAŞ

Enstitü Anabilim Dalı : BİYOMEDİKAL MÜHENDİSLİĞİ

Bu tez 14.08.2020 tarihinde aşağıdaki jüri tarafından oybirliği / oyçokluğu ile kabul edilmiştir.

**Dr. Öğr. Üyesi Ayla
EREN ÖZDEMİR
Jüri Başkanı**

**Dr. Öğr. Üyesi Gökçen
ÇETİNEL
Üye**

**Doç. Dr. Mustafa Zahid
YILDIZ
Üye**

BEYAN

Tez içindeki tüm verilerin akademik kurallar çerçevesinde tarafımdan elde edildiğini, görsel ve yazılı tüm bilgi ve sonuçların akademik ve etik kurallara uygun şekilde sunulduğunu, kullanılan verilerde herhangi bir tahrifat yapılmadığını, başkalarının eserlerinden yararlanılması durumunda bilimsel normlara uygun olarak atıfta bulunulduğunu, tezde yer alan verilerin bu üniversite veya başka bir üniversitede herhangi bir tez çalışmasında kullanılmadığını beyan ederim.

Şerife ÇELİKBAŞ

14.08.2020

TEŞEKKÜR

Danışman hocam Dr.Öğr. Üyesi Ayla Eren ÖZDEMİR'e çalışmalarım için gösterdiği destek ve yardımları için teşekkür ederim. Yüksek lisans eğitimim boyunca değerli bilgi ve deneyimlerinden yararlandığım, her konuda bilgi ve desteğini almaktan çekinmediğim, araştırmanın planlanmasından yazılmasına kadar tüm aşamalarında yardımlarını esirgemeyen, teşvik eden, aynı titizlikte beni yönlendiren değerli hocam Öğr. Üyesi Ali Furkan KAMANLI'ya teşekkürlerimi sunarım.

Malzeme teminatı konusunda anlayış ve yardımlarını esirgemeyen başta Optimum Süreç Tasarımları Ltd. Şti. patronu Tansel Cavit KULAK ve proje sorumlusu Merve DURAK olmak üzere tüm çalışanlarına teşekkür ederim.

Maddi, manevi, dolaylı ya da doğrudan yardımlarından hep yararlandığım aynı kaderi paylaştığım laboratuvar arkadaşlarım Ebru AKSOY, Büşra GÜNOĞLU, Bahar PEHLİVANÖZ ve Enes TOPRAK'a teşekkür ederim. Ayrıca başta Fatih YILMAZ ve Bilal GÜREVİN olmak üzere tüm AGV ekibi arkadaşlarıma tüm yardımları ve emekleri için çok teşekkür ederim. Değerli İlkokul öğretmenim Yaşar ERTÖRÜN'e, lise öğrenimimde bana çok destek olan değerli hocam Derya İLHAN'a, lisans eğitimimden bu yana her yardımına koşan maddi manevi destek olan arkadaşım Cihat ŞAMAN'a tüm emekleri için çok teşekkür ederim.

Doğduğum günden beri maddi manevi desteğini hiç üzerimden çekmeyen, annem Ayşe Hanım ÇELİKBAŞ, babam Emin ÇELİKBAŞ, ablam Arife YAZICI, eniştem İsmail YAZICI ve kardeşim Büşra ÇELİKBAŞ'a ayrı ayrı çok teşekkür ederim.

İÇİNDEKİLER

TEŞEKKÜR	i
İÇİNDEKİLER.....	ii
SİMGELER VE KISALTMALAR LİSTESİ.....	v
ŞEKİLLER LİSTESİ	vi
TABLolar LİSTESİ	ix
ÖZET	x
SUMMARY.....	xi
BÖLÜM 1.	
GİRİŞ	1
BÖLÜM 2.	
KAYNAK ARAŞTIRMASI	5
2.1. Otomatik Yönlendirmeli Araç.....	5
2.1.1. Hastane ortamında kullanılan OYA	6
2.1.2. OYA'ların avantajları.....	7
2.1.3. OYA'ların dezavantajları	7
2.1.4. OYA rehber sistemler	7
2.1.4.1. Jiroskoplu yönlendirme sistemleri	8
2.1.4.2. Kablo yönlendirme sistemleri.....	8
2.1.4.3. Optik (Işık) yönlendirme sistemleri	8
2.1.4.4. Lazer yönlendirme sistemleri	8
2.1.4.5. Yol hesabı yönlendirme sistemi.....	9
2.1.4.6. Spot (Beacon) yönlendirme sistemleri	9
2.2. Oya Çeşitleri	9
2.3. Robot İşletim Sistemi	11

2.3.1. RİS mimarisi.....	12
2.3.2. RİS işlevsel yapısı.....	14
2.3.3. 3B modelleme ve benzetim	15
2.3.3.1. Modelleme	15
2.3.3.2. Simülasyon	16
2.4. Yapay Zekâ.....	19
2.4.1. Yapay zekâ teknikleri	19
2.4.1.1. Yapay sinir ağları	20
2.4.2. Derin öğrenme	29
2.4.2.1. Derin öğrenme mimarileri	32
2.5. İnsan Hareket Tespiti.....	39
2.5.1. Hareket sezimi	40
2.5.1.1. Hareket bölütleme	40
2.5.1.2. Nesne tanımlama.....	41
2.5.2. Hareket izleme.....	41
2.5.2.1. Model tabanlı	41
2.5.2.2. Görünüş tabanlı.....	41
2.5.3. Hareket anlama.....	42
2.5.3.1. Aktivite tanıma ve tanımlama.....	42

BÖLÜM 3.

MATERYAL VE YÖNTEM.....	43
3.1. Materyal.....	43
3.2. Yöntem	44
3.2.1. Kullanılan araç-gereçler	44
3.2.2. Kullanılan programlama dilleri	45
3.2.3. Simulasyon çalışmaları	45
3.2.3.1. RİS kurulum işlemleri	47
3.2.3.2. Cihazın modellenmesi	47
3.2.3.3. Simülasyon ortamının oluşturulması.....	48
3.2.3.4. RPLİDAR ile haritalama işlemi.....	48
3.2.3.5. Gazebo simülasyon ortamının oluşturulması	49

3.2.3.6. OYA'nın hareketi.....	49
3.2.3.7. Gerçek zamanlı davranış tespit etme.....	50
3.2.4. Gerçek ortamda yapılan çalışmalar	55
3.2.4.1. Sensör düğümlerinden bilgi akışı sağlanması	56
3.2.4.2. Cihazın hareketi	57
3.2.4.3. Gerçek zamanlı davranış tespit etme.....	57
3.2.5. Merkeze anormal durumların bilgisinin gönderilmesi.....	58
BÖLÜM 4.	
ARAŞTIRMA BULGULARI.....	61
4.1. Davranış Tespiti Başarı Oranları.....	61
4.1.1. Matematiksel yöntem.....	61
4.1.2. Yapay zeka yöntemi	61
4.2. Simülasyon Ortamı ve Gerçek Ortam Kıyaslaması	63
BÖLÜM 5.	
TARTIŞMA VE SONUÇ	65
KAYNAKLAR	68
ÖZGEÇMİŞ.....	74

SİMGELER VE KISALTMALAR LİSTESİ

BSD	: Berkeley Software Distribution
CNN	: Convolutional Neural Network
DAE	: Deep Autoencoder
DBN	: Deep Belief Network
GPU	: Grafik Processing Unit
LSTM	: Long Short-Term Memory
MNIST	: Modified National Institute of Standards and Technology
OMT	: Otomatik Mal Transferi
OYA	: Otomatik Yönlendirmeli Araç
PR	: Personel Robot
RBM	: Restricted Boltzmann Machines
ReLU	: Rectified Linear Unit
RNN	: Recurrent Neural Network
RİS	: Robot İşletim Sistemi
SLAM	: Simultaneous Localization And Mapping
STAIR	: STanford AI Robot
TRS	: Transfer Ray Sistemi
VNC	: Virtual Network Computing

ŞEKİLLER LİSTESİ

Şekil 1.1. Tez çalışması akış diyagramı.....	3
Şekil 2.1. Hastane Otomatik Yönlendirmeli Araçları.....	6
Şekil 2.2. Çatallı OYA'nın paletizer işlemi sonunda paleti alması.....	10
Şekil 2.3. Konveyörlü OYA.....	10
Şekil 2.4. Kaldıraçlı OYA.....	10
Şekil 2.5. Konteyner taşıyıcı OYA.....	11
Şekil 2.6. Çekici OYA.....	11
Şekil 2.7. RİS dosya yapısı.....	13
Şekil 2.8. RİS İşlevsel Yapısı.....	14
Şekil 2.9. Gazebonun ilk kez açıldığında görülen ekran.....	17
Şekil 2.10. Rviz ilk kez açıldığında görülen ekran.....	18
Şekil 2.11. Nöron Hücresi.....	21
Şekil 2.12. Sinir Hücresinin matematiksel modeli.....	22
Şekil 2.13. Tek Kutuplu Sigmoid Grafiği.....	24
Şekil 2.14. Çok Kutuplu Sigmoid Grafiği.....	24
Şekil 2.15. Hiperbolik TanH Fonksiyonu.....	24
Şekil 2.16. ReLU Fonksiyon Grafiği.....	25
Şekil 2.17. Softmax fonksiyonu işlevi gösterimi.....	26
Şekil 2.18. İleri Beslemeli Yapay Sinir Ağı Modeli.....	26
Şekil 2.19. Çok Katmanlı Yapay Sinir Ağı Modeli.....	27
Şekil 2.20. Hopfield Geri Beslemeli Yapay Sinir Ağı Modeli.....	28
Şekil 2.21. Danışmanlı Öğrenme Yapısı [37].....	29
Şekil 2.22. İlk derin ağ mimarisi ve Ivakhnenko [41].....	31
Şekil 2.23. İnsan Görme Sistemi.....	33
Şekil 2.24. CNN Modeli Ağ Yapısı.....	34
Şekil 2.25. Tekrarlayan Sinir Ağı Yapısı.....	35

Şekil 2.26. RNN ağ yapısı	36
Şekil 2.27. LTSM ağ yapısı	36
Şekil 2.28. Sınırlı Boltzmann Makineleri Ağ Yapısı	37
Şekil 2.29. Derin İnanç Ağları (DBN).....	38
Şekil 2.30. Derin Oto-Kodlayıcı Ağ Yapısı	38
Şekil 2.31. İnsan hareketi tespiti aşamaları	39
Şekil 3.1. SUBÜ Deneysel OYA.....	43
Şekil 3.2. Simülasyon çalışmaları sistem diyagramı.....	44
Şekil 3.3. Gerçek ortam sistem diyagramı.....	44
Şekil 3.4. Hastane koridorlarını temsil eden simülasyon ortamı RViz haritası ve Gazebo çizimi.....	46
Şekil 3.5. Simülasyon çalışmaları akış diyagramı	46
Şekil 3.6. Gerçek Cihaz ve Simülasyonda kullanılan model.....	47
Şekil 3.7. RPLIDAR 360 ° ile haritalama	48
Şekil 3.8. Çıkarılan haritanın RViz’de gösterimi.....	48
Şekil 3.9. Gazebo ortamında alınan haritanın çizimi.....	49
Şekil 3.10. Test simülasyon ortamının son hali	49
Şekil 3.11. RPLİDAR 360 Lazer Tarayıcı.....	50
Şekil 3.12. Gines Hidalgo ve Hanbyul Joo tarafından Openpose Algoritmasının Gösterimi.....	51
Şekil 3.13. OpenPose iskelet eklem bölgeleri	51
Şekil 3.14. Matematiksel Yöntem Referans İşlemleri	51
Şekil 3.15. Gazebo ortamında matematiksel modelin test edilmesi işlemi	52
Şekil 3.16. Verisetinde bulunan düşme, koşma, yürüme, oturma, kavga, eylem, bekleme eylemleri.....	53
Şekil 3.17. Derin Sinir Ağı Eğitim Modeli.....	54
Şekil 3.18. Derin Sinir Ağı yöntemi ile oluşturulan algoritmanın Gazebo ortamında test edilmesi	55
Şekil 3.19. Gerçek ortam çalışmaları akış diyagramı.....	55
Şekil 3.20. Çok kişinin birden koşması durumunun gösterimi.....	59
Şekil 3.21. Birinin düşmesi durumunun simülasyonda gösterimi	59
Şekil 3.22. Kavga ortamının gerçek ortamda gösterilmesi.....	60

Şekil 4.1. Simülasyon ortamında matematiksel işlemin testi	61
Şekil 4.2. Eğitim modeli doğruluk ve kayıp grafikleri.....	62
Şekil 4.3. Karışıklık Matrisi (Confusion Matrix).....	62

TABLolar LİSTESİ

Tablo 2.1. rosmg ve rosservice işlemleri için bazı komutlar	14
Tablo 3.1. Verilerin yapay sinir ağına geçmeden önce düzenlenmiş hali	53
Tablo 3.2. Jetson nano ile rapberry pi B + fps değerleri	58
Tablo 4.1. Matematiksel yöntem ve yapay zekâ yönteminin davranışları tespit etme oranları grafiğı	63
Tablo 4.2. Gerçek ortam ve Simülasyon ortamının davranış tespiti için kıyaslanması	64

ÖZET

Anahtar kelimeler: Robot işletim sistemi, derin öğrenme, otomatik yönlendirmeli araç, davranış tespiti

Otomatik yönlendirmeli araçlar (OYA); hasta ihtiyaçlarını daha hızlı gidermek, hasta bakım ve verimliliğini arttırmak, akıllı hastaneler oluşturmak gibi amaçlarla hastaneler içinde kullanılmaktadır.

Çalışmada, OYA'ların hastane ortamındaki kişilere ve nesnelere zarar vermeden yol alması sağlanmaktadır. Çevresinde bulunan kişilerin davranış bilgisi gerçek zamanlı olarak yapay zeka ve görüntü işleme yöntemleriyle merkeze bildirilmektedir. Sorun oluşturma ihtimali olan; çok kişinin aynı anda koşması, herhangi birinin düşmesi, kişilerin kavga etmesi gibi durumlara göre OYA'nın hareket edebilmesi sağlanmaktadır. Davranış tespitinde, matematiksel yöntem ve gönüllü deneklerin yıldız iskelet modellerinden elde edilen data'lara göre eğitilmiş bir derin öğrenme yöntemi kullanılmaktadır. Tüm çalışmalar öncelikle belirlenen deneysel ortamın haritasının alınmasının ardından robot işletim sistemi (RİS) içerisinde simüle edilmiştir. Daha sonrasında deneysel bir OYA ile gerçek ortamda test edilmiştir.

Yapılan çalışmalar sonunda, derin öğrenme yönteminin matematiksel yöntemle kıyasla davranış tespiti için daha iyi sonuçlar verdiği görülmektedir. Ayrıca, tez çalışması hastane ortamında malzeme taşıma işlemi yapan OYA'lar için yeni bir alan önerisi sunarak, görsel gözetleme sistemlerinde kullanılabileceğini göstermiştir. Görsel gözetleme, hastane ortamı gibi insanların toplu halde bulunduğu ortamlarda önem kazanmaktadır. Bu anlamda kullanılan hibrit OYA'lar görsel gözetleme sistemleri içinde çalışan operatörler için iş kolaylaştırıcı olmaktadır.

AUTOMATIC GUIDED HOSPITAL VEHICLE WITH INTELLIGENT VISUAL SURVEILLANCE SYSTEM

SUMMARY

Keywords: Robot operating system, deep learning, auto guided vehicle, behavior detection

Automated guided vehicles (AGV); It is used in hospitals for purposes such as meeting patient needs faster, increasing patient care and efficiency and creating smart hospitals.

In practice, it is ensured that AGV's travel without harming the people and objects in the hospital environment. Behavioral information of the people around is reported to the center in real-time with artificial intelligence and image processing methods. Possible problem; It is ensured that AGV can act according to situations where many people run at the same time, everyone falls and fights. In determining the behavior, a deep learning method trained according to the mathematical method and data obtained from the star skeleton models of volunteer subjects are used. All studies were simulated in the robot operating system (ROS) after mapping the experimental environment determined first. It was then tested with an experimental OGV in a real environment.

As a result of the studies, it is seen that the deep learning method gives better results in behavior detection than the mathematical method. In addition, the thesis study has shown that it can be used in visual surveillance systems by offering a new field proposal for AGV's carrying materials in a hospital setting. Visual surveillance gains importance in environments where people are in groups such as the hospital environment. In this sense, hybrid AGV's are business facilitators for operators working in visual surveillance systems.

BÖLÜM 1. GİRİŞ

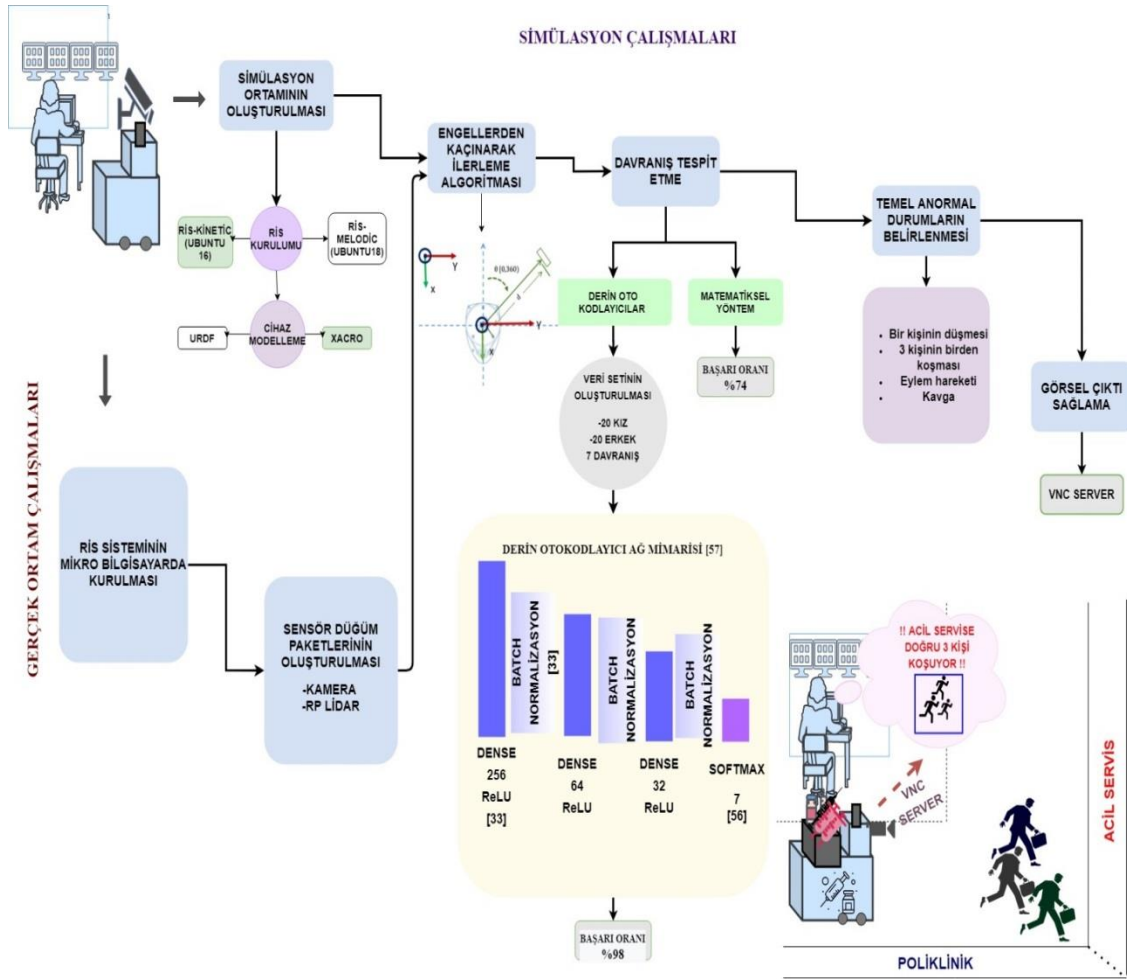
Otomatik güdümlü ya da Otomatik yönlendirmeli araçlar olarak adlandırılan OYA'lar; merkezi bir kontrol ünitesi tarafından kontrol edilen, rehber yolları takip eden bir tür taşıma birimidir. Ürünleri, farklı istasyonlara taşıma, istifleme gibi işlemleri otonom olarak gerçekleştirmektedirler [1,3]. 1953 yılında icat edilen OYA'lar otomotiv endüstrisi başlangıçta olmak üzere çeşitli endüstrilerde benimsenmiştir. 1980'li yıllara gelindiğinde ise hastane ortamı gibi insanların toplu halde bulunduğu ortamlarda hastane lojistiği içinde kullanılmaya başlanmıştır. OYA'lar iç dağıtım görevlerinde büyük bir gelişmenin öncüsü olmuştur [3]. İlk kullanılmaya başladıkları zamanlardan bu yana birçok yeni özellik eklenerek giderek daha kullanışlı, daha güvenilir ve ergonomik hale gelmişlerdir. Ayrıca içerisinde çok sayıda yüksek teknoloji düzeyinde kontrol işlemini birlikte barındırmaktadırlar. Bu kapsamda OYA gelişimi endüstri 4.0'a oldukça uygun olarak ilerlemektedir [4].

Tarihte gerçekleşen son sanayi devrimi olarak ifade edilen Endüstri 4.0, insan gücüyle yapılan işlerin makine gücüne dönüştürülerek tüm üretim ve hizmet sektöründeki süreçlerin kendiliğinden yönetilebilir hale gelmesi olarak tanımlanabilmektedir [5]. OYA'lar, Endüstri 4.0 kapsamında önem kazanan görüntü işleme, makine öğrenmesi, nesnelere interneti (IoT), yapay zekâ gibi konulardan etkilenmiştir. Örneğin, bir şirket için OYA filosu satın almanın uygunluğunun ve finansal fizibilitesinin belirlenerek oluşturulmasının, yol düzenlerinin oluşturulmasının, yönlendirme tabloları, operasyon görevleri ve ağ akışı ile birlikte düşünülerek esnek bir üretim sisteminin tasarımı oldukça karmaşık ve çok parametrelidir. Bu nedenle bulanık kararlı nitel karşılaştırmalı analiz aracılığıyla, yönetsel karar vericilerin memnuniyet düzeyini ölçme çalışması yapılmıştır [6,7]. Endüstri 4.0 a göre OYA, değişen durumlarda otomatik planlama yapmalı, sağlamlığını koruyabilmeli ve her an nakliyelenebilirliğe açık olmalıdır. Bu açıdan

OYA'lar akıllı olmalı ve faydalı bilgiler toplayabilmelidir. Bu kapsamda, oldukça çok çalışma yapılmıştır. Askeri ekipmanlarda pil kullanımını ve performansını optimize etmek için yapay zeka kullanımını önererek araştırmanın sonucunda pil yönetim sisteminde yapay zeka ile yapılan çalışmadan verimli sonuçların elde edildiği çalışma bunlardan birisidir. [8].

Endüstri 4.0 alanındaki gelişmeler, yapay zeka uygulamalarının yaygınlaşması ve teknolojik gelişimin hız kanarak artması robot – insan etkileşiminin önemini de arttırmıştır. Bu kapsamda OYA'lar insana yardımcı olarak insanların görevlerinde tamamlayıcı rolüne girmişlerdir. Malzeme taşınmasında kısmi otomasyonu için insan kullanılabilir. Bu sistemde OYA hedefine operatörü takip ederek ulaşmaktadır. Bu sistemde operatör üstünde bulunan referans bir etiket ile hangi operatörü takip edeceği belirlenmektedir [11]. İnsan hareketi algılayarak ve bu algılanan harekete göre OYA cihazının kontrolünün sağlanması gibi çalışmalar da yapılmaktadır [12]. Yine yapay zeka yöntemleri kullanılarak operatörün yaptığı hatalar OYA üzerinde bulunan kinect kamera vasıtasıyla alınan eklem bilgileri referans alınarak görüntü işleme yöntemleriyle tespit edilebilmektedir [13].

Yapılan tez çalışmasında, OYA'nın yapay zekâ ile entegrasyonu konusundaki çalışmalarına farklı bir açıdan yaklaşarak insanların toplu halde bulunduğu ortamlarda kullanılan hastane OYA'ları için akıllı ve gezgin bir görsel gözetleme sistemi önerilmektedir. Bu ortamlarda görsel gözetleme sistemleri, toplum huzuru, acil müdahale ya da çeşitli güvenlik problemleri için oldukça önemli bir yere sahiptir. Tez çalışmasında da hastane koridorlarında taşıma için kullanılan OYA, engellere çarpmadan ilerlerken, insanların hareketlerini analiz ederek belirli temel anormal olarak kabul edilecek durumların tespit edilmesini sağlamaktadır. Bu şekilde oluşturulan hibrit OYA ile gezgin bir görsel gözetleme sisteminin oluşturulması; malzeme taşımının yanısıra, standart görsel gözetleme sistemlerine göre kör noktaları yakalayabilme imkânı, operatörün iş yükünün azaltılması ve bulunulan ortamda fenalaşan ya da tedirginlik oluşturan genel durumların kısa sürede haber merkezine iletmesi gibi imkanları sunmaktadır.



Şekil 1.1. Tez çalışması akış diyagramı

Bu tez çalışmasında, Şekil 1.1.'de verilen akış diyagramında olduğu gibi işleyiş gerçekleştirilmiştir. Öncelikle simülasyon çalışmaları Robot işletim sistemi (RİS) platformu içerisinde tamamlanmıştır. Ardından simülasyon çalışmaları ışığında belirlenen algoritmalar gerçek ortam çalışmaları için de test edilerek kullanılmıştır. RİS içerisinde kamera ve ingilizcede Rp-Light Detection and Ranging (RPLIDAR) denilen 360° tarayıcı sensör paketleri oluşturularak bu sensörlerden alınan yayınlara göre kontrol sağlanmıştır. OYA, lidardan alınan verilere göre karşısına çıkan engellerden kaçınarak ve en uygun yolu seçerek ilerlemektedir. Bununla eş zamanlı olarak da üzerinde bulunan bir kamera ile etrafındaki kişilerin davranışlarını, matematiksel yöntem ve derin-oto kodlayıcılar derin öğrenme mimarisi olmak üzere iki farklı yöntem kullanılarak tespit etmektedir. Bekleme, eylem, kavga, oturma, yürüme, koşma ve düşme olmak üzere 7 farklı davranış, yapay zekâ eğitimi

sonucunda %98 oranında başarılı tespit edilirken, matematiksel yöntem ile bu davranışlar %74 oranında doğruluk göstermektedir. Çalışmalar simülasyon ortamında test edildikten sonra gerçek cihaza entegre edilmiştir. Cihaz tespit ettiği anormal durumlarda kendini durdurmakta ve ortamı bir süreliğine kayıt altında tutmaktadır. Ve bu vasiyle, malzeme taşıma göreviyle kullanılan OYA'ya farklı bakış açısı kazandırılarak, hastane ortamında güvenlik önlemlerinin daha hızlı alınması ve de çeşitli alarm durumlarında merkeze verilen bilginin daha kısa sürede ulaşması amaçlanmıştır.

BÖLÜM 2. KAYNAK ARAŞTIRMASI

2.1. Otomatik Yönlendirmeli Araç

Otomatik Yönlendirmeli Araç (OYA)'lar manuel yönlendirme olmaksızın üzerlerine monte edilen gömülü bilgisayarlar vasıtasıyla hareket edebilen ve kendisine verilen görevleri gömülü bilgisayar sistemi ve kullanılan çeşitli sensörler sayesinde otomatik olarak gerçekleştiren araçlardır. Sensor çıktıları ve gömülü bilgisayarda oluşturulan programlar sayesinde motor sürme devrelerinin kontrolü sağlanmaktadır. Otomotiv, imalat, dağıtım ve nükleer, patlayıcı madde taşınması gibi birçok alanda kullanılmaktadır. Duyusal ve düzenleyici teknolojinin hızlı gelişimi ile mikroelektronikteki erken gelişmeler OYA'ların yolunu açmıştır. Bugün neredeyse bütün OYA'lar gömülü sistemler (sensörler, tamponlar, kameralar, vb.) ile donatılmıştır [14].

OYA, iç lojistik süreçlerinde verimlilik artırıcı şeffaflığa yol açmaktadır. Ürünlerin OYA'lar ile taşınma aşamaları önceden hesaplanabilmekte ve planlanabilmektedir. Rezerv stoklama ve bekleme süreleri asgari düzeylere indirilebilmektedir. Taşıma için kullanılacak personel sayısında azalma meydana gelmektedir. Bu da personel için harcanan maliyeti oldukça düşürmektedir. Nakliye hasarları ve takip maliyetleri azalmaktadır. Sipariş edilen çalışma adımları, temiz ve sessiz taşıma koşulları içerisinde gerçekleşeceğinden güvenli ve keyifli çalışma koşulları oluşmakta ve böylece çalışma standartları yükseltilmektedir. Yükleme ve trafik alanlarında çoklu kullanımları da mümkündür. Ayrıca hem iç hem dış mekanlarda kullanılabilirler [2]. Bütün bunların yanısıra çeşitli ek fonksiyonlar eklemek de mümkündür. Sıralama, karar verme, veri aktarımı, veri toplama, mal tartma, prosedürleri düzenleme, depo yönetimi, depolama alanlarını yönetme, mal tanıma, çeşitli düzenlere hâkim olma, palet bulma, kamyon yükleme, akıllı güvenlik, akıllı

durumsal tepkiler (yangın alarmı şalteri, çeşitli devreye alma modları), kapalı zamanlarda daha hızlı ve daha karmaşık çalışma, akıllı yükleme stratejileri, mobil robotlar, devreye alma işlevi vb. gibi fonksiyonlar eklenebilmektedir. Bütün bu avantajlar; OYA'ların çeşitli alanlarda çalışanların iş yükünü azaltmanın yanı sıra birçok farklı açıdan iş kolaylaştırıcı ve kullanışa uygun yapıda olduğunu göstermektedir. Bu açıdan ilaç, gıda endüstrisi, hastane lojistiği gibi çeşitli dallarda modern intralojistik uygulamalar için güçlü ve gerekli bir araçtır.

2.1.1. Hastane ortamında kullanılan OYA



Şekil 2.1. Hastane Otomatik Yönlendirmeli Araçları

Birçok alanda kullanılan OYA'lar, 1980'li yıllara gelindiğinde ise hastane ortamı gibi insanların toplu halde bulunduğu ortamlarda hastane lojistiği içinde kullanılmaya başlanmıştır (Şekil 2.1.). Büyük kliniklerde (OMT) otomatik mal transfer ekipmanı veya Tavan Ray Sistemi (TRS), olarak adlandırılan sistemler uzun zamandır kullanılmaktadır. TRS sistemleri, her biri tavanın altına monte edilmiş bir ray boyunca bir konteyneri taşıyan, elektronik olarak kontrol edilen ve elektrikle çalışan tek raylardır. Hastane OYA sistemlerinde; tavan montajı olmadığından kurulum aşamaları kolaylaşmıştır. Esnek kullanım, yeniden programlama olanağı sağlamaktadır. Böylece kullanım yolları isteğe göre farklı yollar ve tesislere göre revize edilebilmektedir. OYA'lar, TRS'ler karşısında bu gibi avantajları gösterdiğinden, dünya çapında pek çok hastanede TRS ekipmanları OYA'lar ile değiştirilmeye başlanmıştır. Hastane OYA'ları genellikle tekerlekli konteynerlerde

taşınan klinik lojistikteki ana mal akışlarına odaklanır; bunlar yiyecek, çamaşırhane, sterilize edilmiş ürünler, tedarik mağazaları ve atıklardır [1].

2.1.2. OYA'ların avantajları

- İşçi gücü gereksinimini azaltmaktadır
- Yapılan iş için ihtiyaç duyulan alan azalmaktadır.
- Malzeme akışı, verimi ve üretim hızında artış ve kolaylık sağlamaktadır.
- Malzeme taşınmasında ürünlerin zarar görmesi engellenmektedir.

2.1.3. OYA'ların dezavantajları

- OYA'lar birçok farklı sistemin birleşiminden oluştuğundan yazılım işlemlerinde karmaşıklık fazladır.
- OYA yeni teknolojilerle oluşturulmuş bir araç olduğundan, cihaz hakkında gerekli bilgilere hâkim olma konusunda operatöre düşen sorumluluk artmaktadır.

2.1.4. OYA rehber sistemler

Rehber sistem; diğer bir deyişle yönlendirme sistemleri, OYA'nın bulunduğu bir konumdan ayrılarak, planlanan konuma giderken hangi tipte yörüngeyi takip edeceğini belirtmektedir. Jiroskoplu yönlendirmeli, kablo yönlendirmeli (wire-guided), optik yönlendirmeli, lazer yönlendirmeli (laser guided), yol hesabı yönlendirmeli ve spot yönlendirmeli sistemler olmak üzere çeşitli rehber sistemler bulunmaktadır. Bu rehber sistemler, birbirlerine göre çeşitli avantaj ve dezavantajlara sahiptirler. Bazıları güvenlik açısından yetersiz kalabilirken daha güvenilir olanları ise üretim sahasının kullanılışında kısıtlamalara sebep olabilmektedir [15,16].

2.1.4.1. Jiroskoplu yönlendirme sistemleri

Modern teknoloji alanının birçok dalında (Pusulalarda, uzay aracında, bilgisayar işaret aygıtlarında) kullanılan jiroskop, yönelim ve dönme hesaplanmasına olanak sağladığı için OYA'larda da kullanılmaktadır. OYA ivmesi ile farklı yönde olan kuvvetlerin tespit edilmesine yardımcı olmaktadır. İvme kuvvetleri jiroskop eksenini dik kesecektir ve iki defa integral alınmasıyla konum sapmasının tespitini de sağlayacaktır [17].

2.1.4.2. Kablo yönlendirme sistemleri

OYA'nın hareket edeceği bölgedeki zeminin altına kablo yerleştirilmektedir. Kablo üzerinden geçen akım ile birlikte elektromanyetik dalgalar yaymaktadır. OYA üzerinde bulunan tarama bobinleri bu elektromanyetik dalgaları algılamaktadır ve böylece OYA belirlenen rotaya göre yol almaktadır. Kullanımı, farklı çalışma ortamlarına kolayca taşınmadığından ve kurulum aşaması zorlayıcı olduğundan esnek değildir ve sistem engebeli ve kirli ortamlarda düzgün çalışmamaktadır. Bunların yanında uygun koşullar sağlandığında çalışması oldukça güvenilir ve hata oranı düşüktür.

2.1.4.3. Optik (Işık) yönlendirme sistemleri

Üzerinde bulunan fotoseller sayesinde, zemindeki florasan, boya veya banttan yansıyan ışığı algılamaktadır. Rotasının farklı çalışma ortamlarına kolay entegre edilebilmesi açısından esnek yapıdadır fakat zemindeki bandların kirlenmesi ve zarar görmesi nedeni ile ışığın şeklinin kalitesinin bozulması güvenlik problemlerine neden olabilmektedir.

2.1.4.4. Lazer yönlendirme sistemleri

Reflektör (yansıtıcı) hedef ve lazer tarayıcılar ile OYA'nın bulunduğu yerin tam koordinatının hesaplanması sağlanmaktadır. Referanslar duvara karşılıklı olarak

yerleştirilmektedir. OYA üzerinde bulunan lazer tarayıcı sensör ile hedeflerin uzaklık açısı bilgileri alınmakta ve bu şekilde koordinat tespiti yapılmaktadır [18]. Yol veya kablolama işlemi gerektirmediği ve duvarlara yerleştirilen reflektörler sayesinde zeminin engebeli, çatlak, kirliliği vs. gibi sorunlardan bağımsız olarak çalışabilmektedir.

2.1.4.5. Yol hesabı yönlendirme sistemi

Kodlayıcı, OYA tekerleğinin dönme açısı hesaplanmaktadır. Denetleyici ise eskiden kayıtlı olan ortam haritası ve OYA'nın başlangıç pozisyonu ile kodlayıcıdan gelen bilgileri birleştirerek cihaz konumunun belirlenmesini sağlamaktadır. Bu yönlendirme sistemlerinde patinaj problemleri görülmektedir. Bu yüzden zeminin yapısına bağımlı olarak sorunlar yaşanabilmektedir.

2.1.4.6. Spot (Beacon) yönlendirme sistemleri

Lazer yönlendirme sistemleri ile aynı mantık ile çalışmaktadır. Spot yönlendirme sisteminde lazerden farklı olarak ışık kullanılmaktadır. Bu sistem diğerlerine nazaran daha az maliyetli ve lazer yönlendirme sistemlerinde olduğu gibi kullanımı esnekler. Uygulama aşamasında, ekonomik ve kolay kullanımından dolayı, yönlendirme sistemi olarak optik yönlendirme ve lazer yönlendirme sistemi kullanılmıştır.

2.2. Oya Çeşitleri

OYA'lar birçok farklı endüstride çeşitli kullanım amaçlarına hizmet etiklerinden genellikle projeye özgü tasarımlar yapılmaktadır. Fakat taşınan malzemelerin cinsine, boyutuna, ağırlığına ya da kullanım alanlarına uygun olarak çeşitli tiplerde sıklıkla kullanılan OYA çeşitleri ise; Çatallı, Yüksek Çatallı, Konveyörlü, Kaldıraçlı, Konteyner Taşıyıcı, Çekici OYA şeklinde sıralanabilmektedir.



Şekil 2.2. Çatallı OYA'nın paletizer işlemi sonunda paleti alması

Çatallı OYA, özellikle paletlerin ve çatala yerleştirilebilecek yapıda olan ağır malzemelerin bir yerden bir yere taşınmasında kullanılmaktadır. Çatallar Şekil 2.2.'de gösterildiği gibi ikili ya da çoklu bulunabilmektedir. Taşıyacağı ağırlığa göre araç boyutları değişiklik göstermektedir [18]. Yüksek çatallı OYA ise çatallı OYA'lara göre daha yüksekte bulunan ürünleri almaya uygun olarak tasarlanmıştır.



Şekil 2.3. Konveyörlü OYA



Şekil 2.4. Kaldıraçlı OYA

Üzerinde bulunan bir rulo konveyör ile (Şekil 2.3.) malzemelerin bir konveyörden diğerine aktarımında kullanılan OYA'lar ise konveyörlü OYA olarak

adlandırılmaktadır. Şekil 2.4.'de gösterildiği şekilde OYA üzerine kaldıraçlar yerleştirilebilmektedir. Bu kaldıraçlar taşınacak malzemenin altına rahatça girerek kaldırmakta ve yükü ulaştırmak istediği hedefe getirdiğinde tekrar alçalarak ürünü güvenli olarak bırakabilmektedir. Bu tür OYA'lar Kaldıraçlı OYA olarak adlandırılmaktadır.



Şekil 2.5. Konteyner taşıyıcı OYA



Şekil 2.6. Çekici OYA

Genellikle liman gibi konteyner tipinde malzemenin daha çok bulunduğu yerlerde (Şekil 2.5.) konteyner taşıyıcı OYA kullanılmaktadır. Oldukça ağır yüklerin taşınması için tasarlanmışlardır. Şekil 2.6.'da gösterilen OYA ise arkasına takılan malzemenin istenilen yere götürülmesini sağlamaktadır.

2.3. Robot İşletim Sistemi

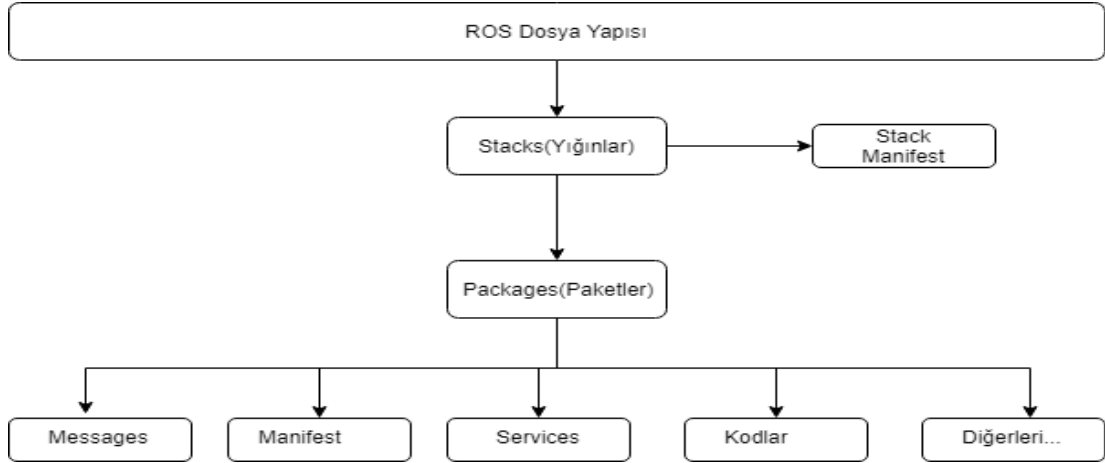
Robot İşletim Sistemi (RİS); bir bilgisayar aracılığı ile OYA bileşenlerinin kontrolünü sağlayan BSD (Berkeley Software Distribution) lisansına sahip açık kaynaklı bir yazılım sistemidir. Robot dünyasında, standardizasyon gerekliliğinin sonucunda ortaya çıkmıştır ve en çok kullanılan açık kaynaklı platformlardan biri haline gelmiştir. Stanford AI Robot (STAIR) ve Kişisel Robotlar (PR) olmak üzere iki önemli program ile RİS'in öncülüğünü 2007 yılında stanford yapay zekâ laboratuvarında Stanford Üniversitesi gerçekleştirmiştir. Stanford yapay zekâ laboratuvarı yapay zekâ ile makine öğrenimi, navigasyon, planlama vb. tekniklerini entegre etmek için bu robot platformu tasarlamıştır [19,20]. Bu laboratuvarında yapılan çalışmanın ardından 2008 yılında Willow Garage, yirmiden fazla farklı kurumlarla

RİS'i geliştirmeye devam etmiştir. Bu kurumlar tarafından RİS'e birçok donanım, iyileştirmeler getirmiştir. RİS için mevcut olan robotlara paralel olarak sensörler gibi çeşitli donanımsal malzemeler de üretilmiştir [21]. RİS, Linux ortamında rahatlıkla kullanılmaktadır, özellikle Ubuntu ile daha iyi çalışmaktadır. Windows ve MacOS X ortamlarıyla da çalışır; ancak, sadece deneyseldir ve verimli bir gerçek zamanlı kullanım için Linux üzerinde çalışmak gerekmektedir.

Paket yönetimi, işlevselliği gerçekleştirme, prosesler arasında mesajlaşma gibi işlemleri gerçekleştirebilmektedir. Haberleşmesi peer-to-peer ağ yapısı şeklindedir. Yavaşlıktan kaynaklanan veri kayıplarını önlemektedir. Haberleşmenin ihtiyaç duyduğu sistemlerin birbirleriyle iletişimlerini gösteren tablo için RİS'te XML-RPC'i içeren master adında bir yapı kullanılmaktadır. RİS, OpenCV/görüntü işleme, yapay zekâ uygulamaları, navigasyon sistemi, çeşitli filtreler gibi birçok çeşitli kodu tekrar tekrar kullanabilmektedir. Yapısal olarak dört başlık altında sınıflandırılabilir bunlar; düğüm, mesaj, topik ve servistir. İşleyişin gerçekleştiği yer düğüm olarak adlandırılmaktadır. Düğümler üzerindeki iletişim mesajlar üzerinden sağlanmaktadır. Mesajlar ise topikler üzerinden akmaktadır. Topikler publish/subscribe adı verilen yapılar ile bağlantı kurmakta yani haberleşmektedir. RİS, geliştiriciler aracılığı ile yazılmış kütüphaneler ve sürücülerini desteklemesinin yanı sıra tek seferde yazılan bir kodun desteklenen tüm aygıtlarda çalıştırılabilmesi imkânı, sensörlerden alınan verilerin işlenerek sonrasında bu verilerle robotlara rahatlıkla görev atama imkânı sağlaması gibi daha birçok avantajla günden güne kullanım alanını arttırmaktadır [20]. Bu proje de kontrol yazılımları RİS uyumlu olacak şekilde geliştirilmiş ve simülasyon çalışmaları RİS ortamında gerçekleştirilmiştir.

2.3.1. RİS mimarisi

Prosesler arasındaki haberleşme asenkron (zaman ayarsız) ve senkron (zaman ayarlı) olarak iki tipe gerçekleştirilebilmektedir. İşlemler gerçek zamanlı olarak görüntülenebilmektedir. Bu avantajları daha iyi anlayabilmek için RİS'in dosya ve işlevsel yapısı olarak iki grupta incelenen RİS mimarisi konusunun iyi anlaşılması gerekmektedir.



Şekil 2.7. RİS dosya yapısı

RİS dosya yapısı Şekil 2.7.'de gösterildiği gibi paketler ve yığınlardan oluşmaktadır. Aynı amaca sahip birden fazla paketin bir araya gelmesiyle yığınlar oluşmaktadır.

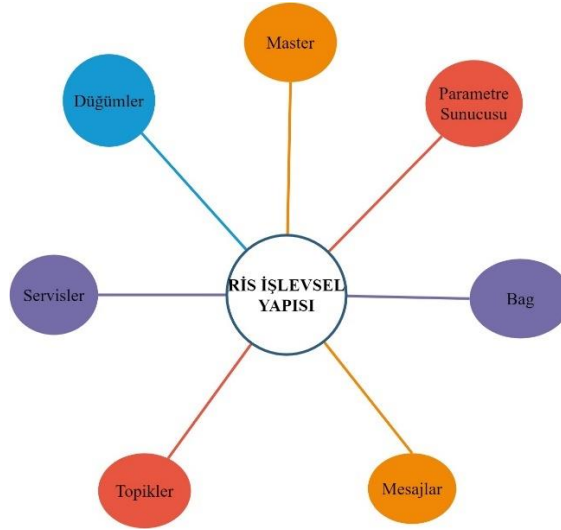
Paketler (Packages): Bir RİS programı geliştirilmek üzere paket oluşturulduğunda bu paketin içinde bin, include, msg, scripts, src, srv, CMakeList.txt, manifest.xml dosyaları bulunması gerekmektedir. Derleme ile elde edilen dosyalar bin/ klasöründe tutulmaktadır. Paket içinde ihtiyaç duyulan kütüphanelerin baş dosyalarının tutulduğu dosya include/ dosyasıdır. Tanımlanan mesaj tiplerinin bulunduğu dosya msg/, standart servislerin dışında tanımlanan servis tipleri için ise srv/ dosyaları kullanılmaktadır. Bash, Python gibi çalıştırılabilir script kodlarını scripts/ dosyası, kaynak C kodlarının bulunduğu kodlar ise src/ dosyasında bulunmaktadır. CmakeList.txt dosyası ise isminden de anlaşıldığı üzere CMake derleme dosyasıdır.

Mesaj (msg) Tipleri: Düğümler arasındaki verileri bir düğümden diğerine taşıma işlemi mesaj işlemi ile yapılmaktadır. Bu mesajlar genellikle bir standarda sahiptir. Eğer bu standartlar dışında bir mesaj tanımlanacak ise msg klasörünün içinde .msg uzantılı olacak şekilde bir mesaj tipi tanımlanabilmektedir. Bu özellikler srv uzantılı dosyalar içinde geçerlidir. Eğer standartlar dışında bir servis tanımlanacak ise srv klasörünün içinde .srv uzantılı olacak şekilde bir servis tipi tanımlanabilmektedir.

Tablo 2.1. rosmg ve rosservice işlemleri için bazı komutlar

rosmg komut	Görev	rosservice komut	Görev
rosmg show	Mesajı görüntülemek için kullanılır.	rosservice call/service args	Verilen argümanlarla hizmetleri listelemek için kullanılır.
rosmg list	Kullanılabilir tüm mesajları listeler.	rosservice find msg-type	Verilen mesaj türündeki hizmetleri bulmak için kullanılır.
rosmg package	Verilen paketteki tüm mesajları göstermek için kullanılır.	rosservice info/service	Hizmetle ilgili bilgileri gösterir.
rosmg packages	Belirtilen iletiyi içeren tüm paketleri verir.	rosservice list	Etkin hizmetleri yazdırır.
rosmg users	Verilen mesaj türüne sahip kod dosyalarını listelemek için kullanılır.	rosservice type/service	Hizmet türünü göstermek için kullanılır.

2.3.2. RİS işlevsel yapısı



Şekil 2.8. RİS İşlevsel Yapısı

RİS'deki birimler arasındaki haberleşme ve bunların kontrolü işlevsel yapıyı tanımlamaktadır. Yapı içerisindeki birimler Şekil 2.8.'de gösterilmiştir. Bu yapı içerisinde;

Düğümler (Nodes): Düğümler hesaplamaların, işlemlerin gerçekleştirildiği yürütülebilir parçalardır. Herhangi bir iletişim problemi olmaması için düğümlere

verilen isimler benzersiz yapıda olmalıdır. Düğümleri oluşturan programa dilleri ise Python ve C++ yazılımlarıdır. Rospy kütüphanesini python dili kullanırken, Roscpp kütüphanesini ise C ++ dili kullanmaktadır. Düğümlerle ilgili bilgi almak için ise rosnode kullanılmaktadır. RİS'in en iyi özelliklerinden birisi de kodu en baştan derlemek yerine konunun parametre ve adını değiştirerek hızlı çözüm elde etmektir. Farklı işlemler birçok farklı düğüm ile gerçekleştirildiğinde herşeyi yapan tek bir düğüme göre çok daha kullanışlı ve fonksiyoneldir. Düğümlerin birbiriyle haberleşmesi ise RİS ağı ile gerçekleşmektedir.

Master: Düğümleri yöneten kayıt servisi olarak adlandırılabilir. Haberleşmedeki mesaj bilgileri gibi bilgileri bünyesinde bulundurmaktadır. Düğümlerin birbirini bulması, mesaj alışverişi yapması ve servislerin rahatça çağırılması için master olmazsa olmaz bir yapıdır.

Parametre Sunucusu (Parameter Server): Düğüm parametrelerini merkezi ortamda tutmaktadır.

Topikler: Düğümlerin veri dönüşümü için kullandığı yapılardır. Bir çalışmada herhangi bir veriye abone olmak için onunla aynı mesaj türüne sahip olmak gerekmektedir.

Bags: Mesaj bilgilerinin kaydedilmesi ve tekrardan kullanılmasını sağlayan yapıdır. Algılayıcı bilgileri gibi bilgilerin test, geliştirme aşamalarında kaydedilmesi ve gerçek zamanlı olarak kullanılmasını sağlayan yapıdır.

2.3.3. 3B modelleme ve benzetim

2.3.3.1. Modelleme

Pratikte gerçekleşen olayları veya çeşitli ihtimalleri bilinen teorik bilgileri kullanarak basit ve anlaşılır şekilde gösterme işlemine modelleme denmektedir. OYA cihazı

üzerinde gerçekleştirilecek uygulamaların direk olarak gerçek ortamda denenmesi yerine işlemleri daha hızlı çözümlmek için de modelleme işlemleri yapılmaktadır.

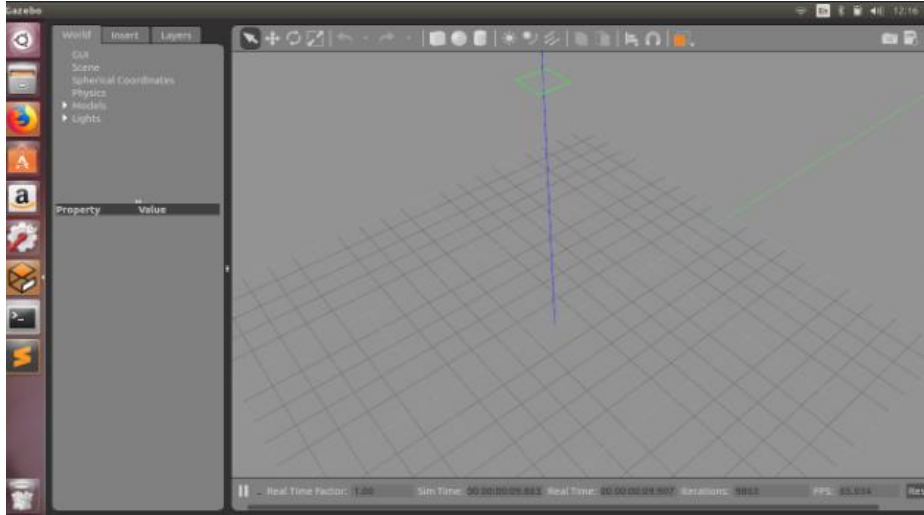
2.3.3.2. Simülasyon

Modellemede bir önceki başlıkta da anlatıldığı gibi bir sistemi incelemek üzere o sistemin basit bir numunesinin yapılması anlamı bulunmaktadır. Simülasyon kavramı ise bundan farklı olarak bir sisteme ait neden sonuç ilişkilerinin bilgisayar ortamına daha detaylı bir şekilde aktararak sistemin davranışlarının bilgisayarda izlenmesini sağlar. Bir tür modelleme tekniğidir. Kısaca özetlemek gerekirse; gerçek veya teorik bir fiziksel sistemin modelini tasarlama, modeli yürütme ve yürütme çıktısını analiz etme işlemidir [22]. Simülasyon robotik sistemler için de oldukça önemli ve gereklidir. OYA, dağınık yapıda yazılımlar tarafından kontrol edilen çok sayıda sensör, mikroişlemci, mikrobilgisayar gibi componentlerle donatılmış karmaşık bir donanım cihazıdır. Cihaz birçok farklı ortam için stabil şekilde çalışır yapıda olmalı görevlerini başarılı şekilde yerine getirmelidir. Farklı farklı test alanı oluşturmak çok sayıda koşulu defalarca denemek hem maliyetli hem de zaman alıcıdır. Bu açıdan OYA modelleme ve simülasyon işlemi, oldukça önemli bir yere sahiptir. Robot kontrolü için birçok yazılım platformu bulunmaktadır. Fakat RİS daha güvenilir robot kontrol ve navigasyon yazılımı hizmeti sunmaktadır. Ayrıca son yıllarda kullanımı giderek yaygınlaşan bir platformdur. Muadillerine kıyasla bütün bu avantajlar ve gereklilikler göz önüne alınarak tez çalışmasında RİS platformu kullanılmıştır. Ayrıca sistemi çevrimdışı platformlarda çalıştırma imkanı sunması ve matematiksel arka planına hakim olmadığımız başka bir robot üzerinde çalışma ve deneme yapma imkanı sunması da simülasyon ortamı için RİS sisteminin yani Gazebo ve RVIZ simülatörlerinin seçilmesinde önemli bir rol oynamıştır. RİS içerisinde yer alan Gazebo simülasyon ortamı, RİS'in RViz kütüphanesi ile birlikte, sonuçların gerçek robot donanımına doğrudan entegre edilebileceği simülasyonun oluşturulmasına yardımcı olmaktadır [23].

3.2.3.2.1. Gazebo

Gazebo, açık kaynaklı 3D simülatörüdür. Brian Gerkey, Richard Vaughan ve Andrew Howard tarafından Güney Kaliforniya Üniversitesi'nde,"Player" adlı bir projenin parçası olarak geliştirilmiştir. Nate Koenig ve Andrew Howard ise 2002 yılında Gazebo'yu geliştirmiştir. 2011'den ise geliştirici ekibi bağımsız olarak Gazebo geliştirmeye devam etmiştir. Ayrıca Gazebo, Willow Garage tarafından RİS ve PR2 robotunun geliştirilmesinde öncü olmuştur. Brian Gerkey ve Nate Koenig tarafından kurulan Açık Kaynak Robotik Vakfı, yakın zamanda Gazebo'nun geliştiriciliğini yapmaktadır. Gazebo robotların modellerinin oluşturulduğu yapı olmanın yanında derinlik kamerası veya lazer tarayıcı gibi gerçek sensör verilerini de simüle etme işlemleri için bir platform sağlar. Gazebo kendi robotunuzu oluşturabilmenin yanında açık kaynaklı başka robotları kendi çalışmanız için test etme imkanı da bulunmaktadır [24].

RİS indirildiğinde gazebo otomatik olarak inmektedir. Gazebonun çalışmasını test etmek için terminal ekranına “gazebo” yazmak yeterli olmaktadır.



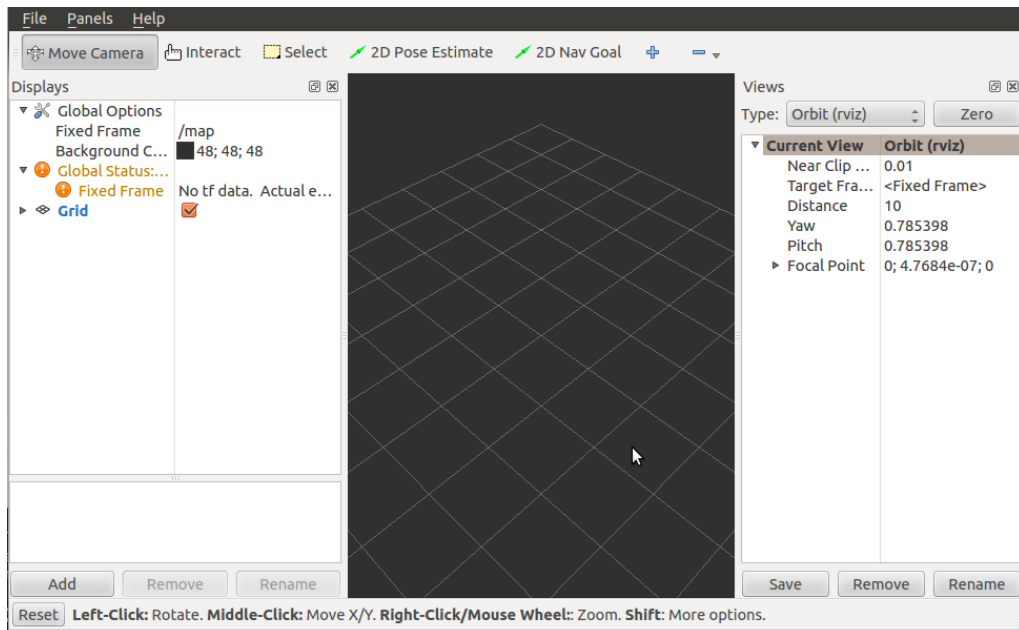
Şekil 2.9. Gazebonun ilk kez açıldığında görülen ekran

Gazebo, diğer robot simülasyon ortamlarına kıyasla; RİS işletim sisteminin sunduğu avantajların yanı sıra farklı yazılım dillerini desteklemesi, bağımsız bir şekilde çalışabilmesi veya başka bir Gazebo ya da Rviz ile eş zamanlı iletişim kurması gibi

avantajları göz önünde bulundurularak gelecek vadeden bir simülasyon ortamı sunmaktadır [10].

3.2.3.2.2. RVIZ

Rviz, RİS platformunda kullanılan bir başka önemli simülatördür. Rviz in Gazebo'dan farkı; arayüzü çoğunlukla sensörler ve çıkışları ile alakalıdır ve kullanıcıyı sensör çıkışları hakkında bilgilendirmekle ilgilenmektedir. Rviz'in daha iyi anlaşılması için RİS Rviz kullanıcı klavuzu oluşturmuştur [25]. RİS yüklemesi gerçekleştiğinde Rviz ve Gazebo otomatik olarak yüklenmektedir. Yukarı da belirtildiği gibi Gazebo'yu test etmek için terminal ekranına; "gazebo" yazmak yeterli olmakta iken Rviz açmak için ise terminale; "roslaunch rviz rviz" yazılması gerekmektedir.



Şekil 2.10. Rviz ilk kez açıldığında görülen ekran

Şekil 2.10.'da Rvizin ilk açıldığında ızgaralı orta alan 3B görünümü gösterilmektedir. Solda, yüklediğiniz ekranları gösteren Ekranlar listesi bulunmaktadır. Şu anda sadece daha sonra alacağım küresel seçenekleri ve zaman görünümünü içermektedir.

Gazebo ile entegre çalışabilen Rviz, Robot sistemleri arasında standardizasyon sağlamak ve haritalama işlemlerini gerçek ortam ile eş zamanlı yüksek doğrulukta takip etme gibi avantajlarıyla muadillerine kıyasla RİS sisteminin getirdiği birçok avantaj sunmaktadır [23, 25].

2.4. Yapay Zekâ

Bir insanın düşünmesi, akıl yürütmesi, algılaması, kavraması ve yargılaması gibi eylemlerine dayanarak bir sonuç çıkarması durumu zekâ olarak nitelendirilmektedir. Yapay zekâ ise kısaca; bir bilgisayarın insana ait olan bu zekâ kavramını yerine getirebilmesi ile oluşmaktadır.

İnsan beyninin fonksiyonları hakkında 1890'lı yıllarda doğru bilgilerin ortaya çıkmasının ardından 1943 yılında yapay zekâ konusunda çalışmalar başlamıştır [28]. Bilim adamlarının araştırmalarına göre yapay sinir hücreleri kullanılarak oluşturulan bu hesaplama modeli; fizyoloji, önermeler mantığı ve Turing'in hesaplama kuramına dayanmaktadır. Bu ağ yapılarının uygun şekilde tanımlanması durumunda öğrenme becerisi kazanabileceği gösterilmiştir.

2.4.1. Yapay zekâ teknikleri

Başlıca yapay zekâ teknikleri aşağıda gösterilmektedir;

- Yapay Sinir Ağları (YSA)
- Bulanık Mantık
- Sezgisel Algoritmalar (Genetik Algoritmalar, Tabu Arama, Tavlama Benzetimi, Karınca Algoritması gibi)
- Uzman Sistemler

Bulanık Mantık; normal mantıktan farklı olarak oluşmuş durumları derecelendirerek sunan bir mantık sistemidir. Örneğin bir cismin normal ağırlığı 50 kg olarak kabul edildiğinde normal mantığa göre 52 kg ağırlığındaki cisim ağır yani ağırlık kümesindeki 1 elemanını almaktadır. 50 kg dan küçük ağırlıkta olduğunda da hafif 0

elemanını almaktadır. Bulanık mantığa göre ise ağırlık kümesi 0 ve 1 aralığındaki tüm sonsuz değerleri almaktadır. Tıpta kullanılan kavramların büyük bir kısmı belirsiz ve karmaşıktır. Bu yüzden tıbbi problemler üzerinde bulanık mantık çalışmaları yapılmaktadır.

Sezgisel Algoritmalar; en iyi sonucu bulacağını garanti etmeden ve kanıtlama sorumluluğunda olmadan kısa sürede doğruya yakın bir sonuç elde etmeye yarayan algoritma türleridir. Bu algoritmalara örnek olarak; genetik algoritmalar, tabu arama, karınca algoritması, A* algoritması verilebilir.

Uzman sistemler ise; gerçek uzman kişilerden alınan bilgileri kullanarak zamanla kendisini geliştiren ve sonucunda uzmanın vardığı sonuca varabilen sistemlerdir. Örneğin; mesaj işleme ve sınıflandırma işlemleri için uzman sistemlerden yararlanılabilmektedir. Kurumlarda iletilmesi gereken birçok sayıda mesajı ayrı ayrı okuyarak sınıflandırmak ve doğru yere iletmek yerine, bu işlem uzman sistemler sayesinde kolayca ve çok daha hızlı yapılabilmektedir.

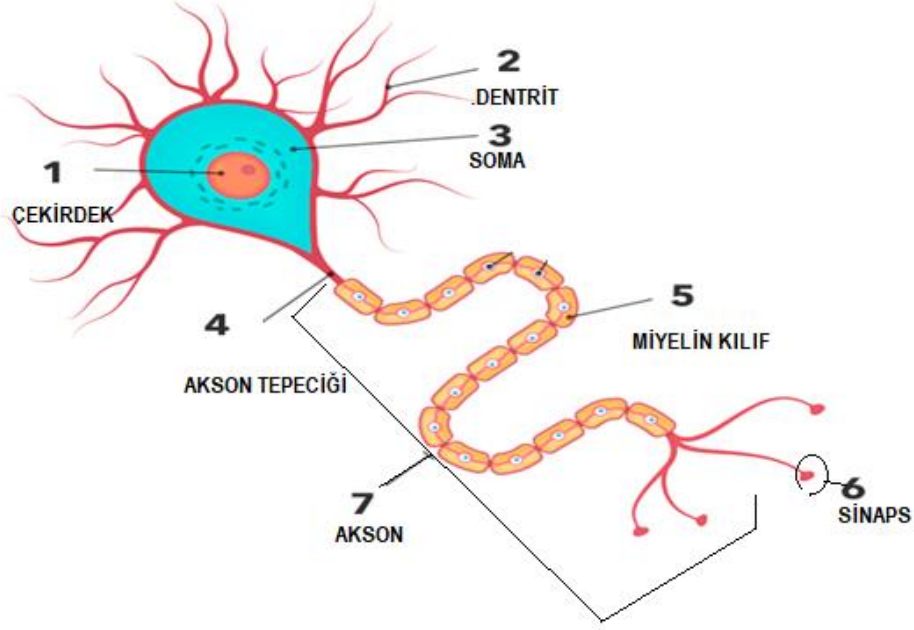
Tez çalışmasında YSA kullanılarak işlemler yapıldığından bu konuya daha ayrıntılı yer verilmektedir.

2.4.1.1. Yapay sinir ağları

Yapay sinir ağları: İnsan fizyolojik sinir sisteminin çalışma prensibini taklit ederek, çeşitli sistemlere öğrenme, hatırlama, genelleme veya çıkarım yapma gibi beceriler kazandırmayı amaçlayan bir yapay zekâ tekniğidir.

Biyolojik sinir sistemi basitçe; alıcı (receptör) sinirlerin organizma içerisinden ya da çevreden gelen uyarınları elektriksel sinyallere dönüştürerek beyne iletmesi ve beynin gelen bu uyarınlara göre tepki (effectör) sinirleriyle elektriksel olarak çıktı üretmesi ve bu çıktıları çeşitli tepkilere dönüştürmesi mantığında çalışmaktadır.

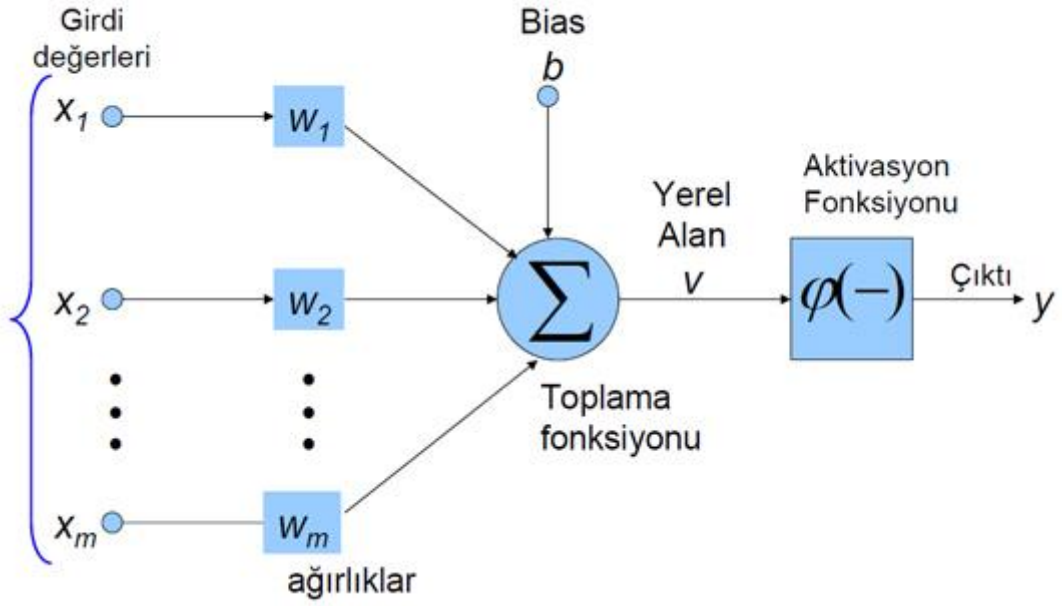
Beynimizde bulunan sinir hücreleri nöron olarak adlandırılmaktadır. Nöronun yapısı aşağıdaki Şekil 2.11.'de gösterilmektedir.



Şekil 2.11. Nöron Hücresi

Şekilde görülen bir nöron hücresi kendisi gibi milyonlarca nöron hücresi ile etkileşim halindedir. Dendrit; diğer nöronlardan gelen iletileri toplayan kollardır. Her nöron, dendritleri ile aldığı sinyalleri basitçe birleştiren bir tür mikro işleme birimi olarak nitendirilebilir. Akson; üzerinde bulunan sinyali diğer nöronlara ileten aktif gövdedir. Miyelin kılıf iletimin hızlanmasını sağlamaktadır. Çekirdek; periyodik olarak işaretlerin yeniden üretilmesini sağlar. Akson ucunda bulunan sinapslar ise diğer nöronların dendritleriyle ilgili nöronun birleşim yeridir. Burada bulunan iletim elektriksel uyarıyı kimyasal uyarıya dönüştüren nörotransmitter denilen kimyasal taşıyıcılar ile sağlanmaktadır. Bu taşıma işlemi temel hafıza sisteminde bulunan bilgiler ile birleştirilmektedir. Nöronlara gelen bu iletilerle uyumlu olarak nöronlar arasındaki sinaptik ilişkilerin değişmesi veya nöronlar arasında yeni sinaptik ilişkilerin kurulması “öğrenme” süreci olarak adlandırılmaktadır [29].

YSA modeli ise nöronların matematiksel modeli ele alınarak oluşturulmuştur. Aşağıda Şekil 2.12.'de nöronun matematiksel modeli gösterilmektedir.



Şekil 2.12. Sinir Hücresinin matematiksel modeli

Şekilde görülen yapı yapay sinir ağının en küçük parçası yani perceptron denen yapıdır. 1957 yılında Frank Rosenblatt'ın tanımladığı perceptron yapısı lineer bir fonksiyondur (Denklem 2.1.).

$$Y=W*X+b \quad (2.1)$$

Girdi değerleri (X), yapay sinir ağının öğrenmesini istediği bilgi için dışarıdan alınan bilgileri ifade eder. Biyolojik sinir hücresindeki karşılığı çevreden veya organizmanın kendi içerisinde gelen ve receptörlerle alınan uyarıcı sinyallerdir.

Ağırlıklar (W), ağa gelen bilginin önemini ve öğrenme üzerindeki etkisini göstermektedir. Fakat bu etki ağırlık değerinin çok olması ile artmamaktadır. Durumlara ve öğrenilecek bilgiye göre bazen ağırlığın çok düşük değerlerde olması çok daha avantajlı olabilmektedir. Ağırlıkların biyolojik sinir sisteminde karşılığı ise sinaps işlemidir.

Toplama Fonksiyonu (Birleştirme Fonksiyonu): Net girdinin hesaplandığı kısımdır. Bu kapsamda çeşitli fonksiyonlar kullanılmaktadır. Ağırlıklı toplam, bunların en çok

bilinenidir. Ağırlıklı toplamaya göre; her gelen bilgi kendi ağırlığı ile çarpılarak toplanmaktadır (Denklem 2.2).

$$(Ağırlıklı Toplam) \text{Net} = \sum_{k=1}^N X_k * W_k \quad (2.2)$$

Toplama fonksiyonunun biyolojik sinir sisteminde karşılığı ise dentritlerdir.

Aktivasyon Fonksiyonu: Toplama fonksiyonu ile hücreye giren net girdinin hesaplanmasının ardından bu gelen net girdiye göre üretilmesi gereken çıktının belirlendiği kısımdır. Bir nöronun çıkış genliğini sınırlamak, daha düzenli halde tutmak için kullanılmaktadırlar. Aktivasyon fonksiyonları; önemli olmayan çeşitli sorunların daha az sayıda düğüm ile hesap edilmesi için genellikle doğrusal yapıda olmazlar. Fakat bazı sistemlerde kullanımına da rastlanmaktadır (örneğin; regresyon problemleri) [30]. Geri beslemeli yapıda olan sinir ağlarının hızlı öğrenilmesi açısından türevi kolay bir şekilde hesaplanabilir bir fonksiyon olması gerekmektedir. Birçok farklı nonlinear yapıda aktivasyon fonksiyonu bulunmaktadır. Bunların başlıcaları;

- Sigmoid
- TanH / Hiperbolik Teğet
- ReLU (Doğrusallaştırılmış Doğrusal Ünite)
- Sızdıran ReLU
- Parametrik ReLU
- Softmax

Sigmoid; kelime anlamı olarak da düşünüldüğünde "S-şekilli" ve sigmoid haritaların lojistik formu olarak nitelendirilmektedir. Tek kutuplu ve iki kutuplu olarak iki adettir. tek kutuplu sigmoid (Denklem 2.3.);

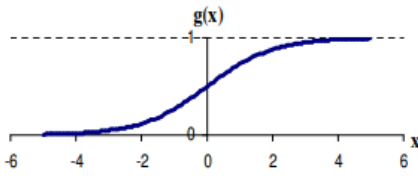
$$g(x) = \frac{1}{1 + e^{-x}} \quad (2.3)$$

Hesaplama kapasitesini oldukça düşürdüğünden ve ayırt etme kabiliyetini arttırdığından dolayı genellikle geri yayılma algoritmaları ile eğitilmiş ağlarda daha avantajlıdır.

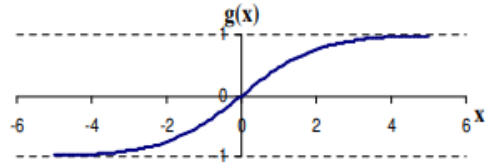
İki kutuplu sigmoid (Denklem 2.4.);

$$g(x) = \frac{1 - e^{-x}}{1 + e^{-x}} \quad (2.4)$$

Çıkış değerleri $[-1, 1]$ aralığında değer üreten uygulamalar için daha avantajlıdır [31].



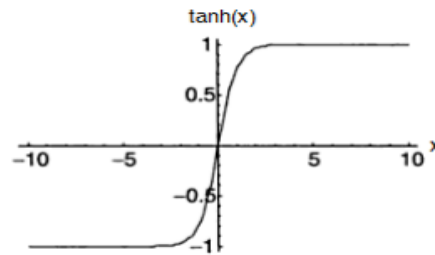
Şekil 2.13. Tek Kutuplu Sigmoid Grafiği



Şekil 2.14. Çok Kutuplu Sigmoid Grafiği

TanH / Hiperbolik Teğet: Bu aktivasyon fonksiyonu; Sinüs ile kosinüs fonksiyonları arasındaki oranı ifade etmektedir (Denklem 2.5.). İki kutuplu sigmoid fonksiyonuna benzer yapıdadır [25].

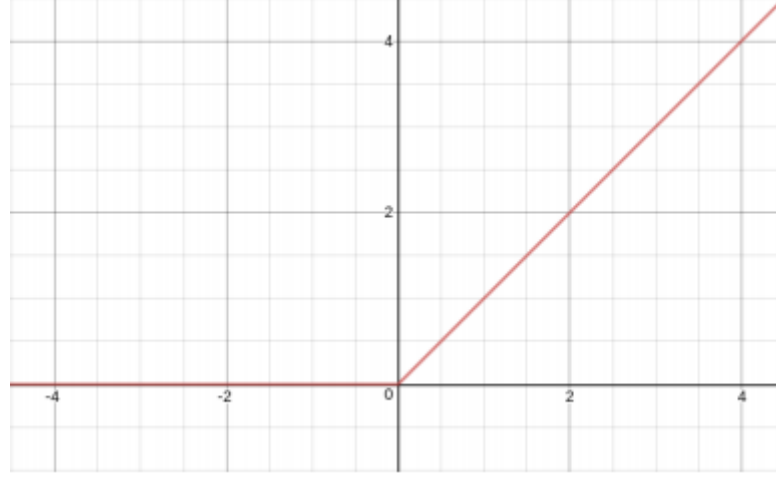
$$\tanh(x) = \frac{\sinh(x)}{\cosh(x)} = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (2.5)$$



Şekil 2.15. Hiperbolik TanH Fonksiyonu

ReLU (Doğrusallaştırılmış Doğrusal Ünite): $x < 0$ olduğunda 0 ve tersine, $x \geq 0$ olduğunda doğrusal bir fonksiyon çıkarmaktadır. Tam anlamı ile doğrusal olmasada

büyük ölçüde doğrusal yapılı bir fonksiyon olması; doğrusal sistemlerdeki avantajların büyük çoğunluğunu taşımasını sağlamaktadır [32].



Şekil 2.16. ReLU Fonksiyon Grafiği

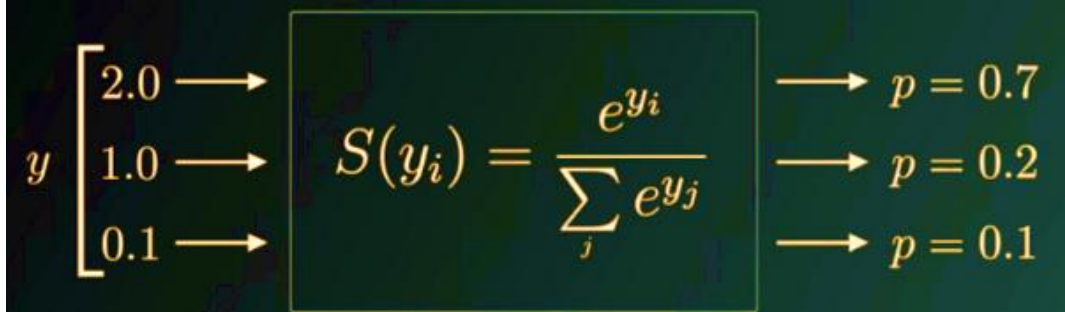
Bu avantajlar;

- Hesaplama yükü daha azdır.
- Diğer aktivasyon fonksiyonlarının aksine gerçek bir sıfır değeri üretebilmektedir. Bu duruma seyrek temsil denmektedir. Bu da öğrenmeyi hızlandırabilmekte ve modeli basitleştirebilmektedir.
- Büyük bir kısmı lineer yapıda olan bir fonksiyon olduğundan optimizasyonu daha kolay olmaktadır.
- Büyük veri kümeleri ile denetlenen işlemleri ön işlemlere gerek duymadan iyi bir performansla eğitmeyi sağlayabilmektedir. Bu da denetimli sinir ağlarının eğitimindeki zorluğu anlama ve çözme konusunda yardımcı olabileceğini göstermektedir [33].

Sızdıran ReLU: Küçük ve sıfır olmayan eğitime izin verme durumunda kullanılmaktadır [34].

Parametrik ReLU: Doğrultucunun parametrelerini uyarlamalı olarak öğrenen bir aktivasyon fonksiyonudur [35].

Softmax: Herhangi bir giriş değerini, toplam olarak 1'e eşit olan bir olasılıksal dağılıma dönüştüren lojistik regresyon şeklindedir.

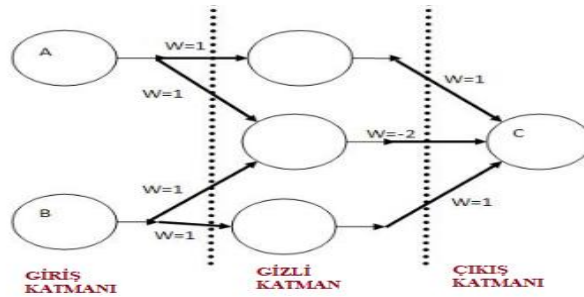


Şekil 2.17. Softmax fonksiyonu işlevi gösterimi

Sinir ağının son çıkış katmanı Softmax fonksiyonu ile işleme koyulmaktadır. Bu katman, sinir ağının karmaşık işlemleri sonrasında elde edilen anlaşılmasız güç değerlerinin anlaşılabilir seviyelerde olmasını sağlamaktadır.

Hücrenin Çıktısı: Değerin aktivasyon fonksiyonundan çıktıktan sonraki halidir. Ağın girdisi çok sayıda fakat çıktısı daha az sayıda ya da tek sayıda olabilmektedir. YSA'lar sinirler arasındaki bağlantıların yönlerine ve yapılarına göre ikiye ayrılmaktadır. Bunlar; ileri beslemeli yapay sinir ağı ve geri beslemeli yapay sinir ağıdır.

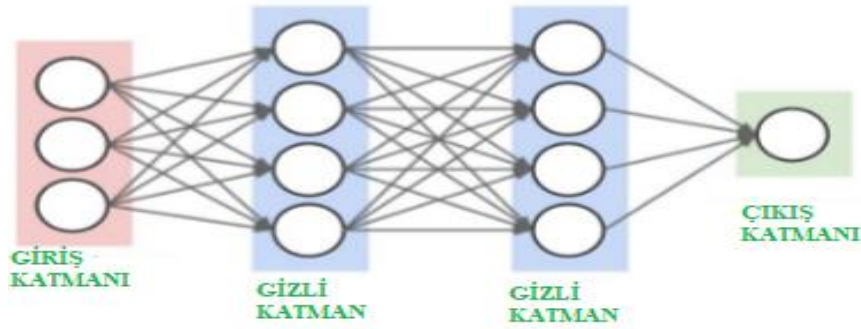
2.4.1.1.1. İleri beslemeli yapay sinir ağlar



Şekil 2.18. İleri Beslemeli Yapay Sinir Ağı Modeli

İleri Beslemeli Yapay Sinir Ağları, Şekil 2.18.'de de gösterildiği gibi; giriş katmanı, gizli katman ve çıkış katmanı yapılarından oluşmaktadır. Sistemde feedback mekanizması yoktur. Bir katmandaki çıkışlar diğer katmanın girişi olacak şekilde işlem yapılmaktadır. Giriş katmanında, alınan veriler değiştirilmeden gizli katmana verilmektedir. Bilginin işlenerek mantıklı çıkarımların oluşmasını sağlayan işleme aşamaları gizli ve çıkış katmanında gerçekleştirilmektedir.

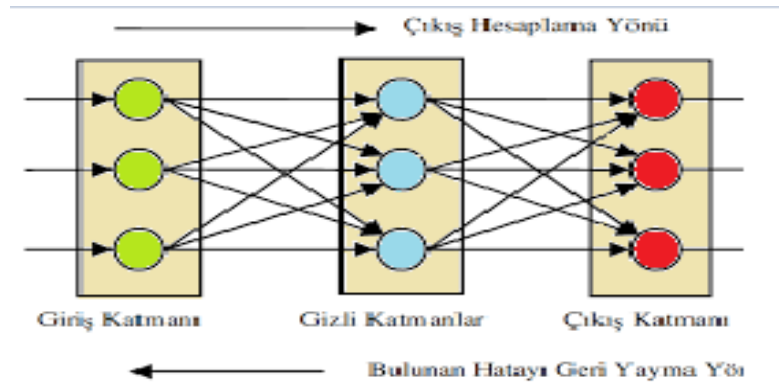
2.4.1.1.1. Çok katmanlı yapay sinir ağları



Şekil 2.19. Çok Katmanlı Yapay Sinir Ağı Modeli

Yapay zekâ ile yapılması gereken işlemler karmaşık doğrusal olmayan yapılarda olduğunda tek katmanlı ağ yapıları yetersiz kalmaktadır. Çok katmanlı ağ yapıları bu amaçla oluşturulmuştur. Birçok öğrenme algoritması bu sebeple çok katmanlı yapılardan oluşmaktadır. Şekil 2.19.'da çok katmanlı ağ yapısı için bir örnek bulunmaktadır. Bu ağ yapısı, giriş, çıkış ve birden çok gizli katmandan oluşmaktadır. Bu tür ağlarda bilgi akışı ileri yönde gerçekleşmektedir. Eleman sayısı yapılacak çalışmaya göre değişkenlik göstermektedir. Ara katman sayısı ya da ara katmandaki eleman sayısı için genl geçer bir kural bulunmamaktadır. Deneme yanılma yoluna göre en iyi doğruluk elde edilmesi ile oluşturulmaktadır.

2.4.1.1.2. Geri beslemeli yapay sinir ağı



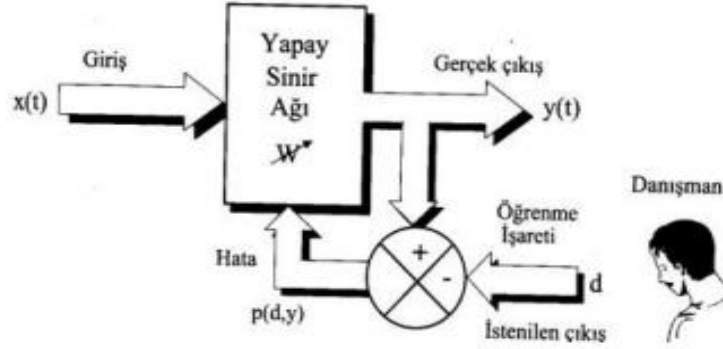
Şekil 2.20. Hopfield Geri Beslemeli Yapay Sinir Ağ Modeli

Bu ağ yapılarında; herhangi bir hücrenin çıkışı kendisine veya gerisindeki bir hücreye giriş olarak verilebilmektedir. Geri besleme ağın farklı yerlerinde olabilmektedir. Doğrusal değillerdir ve yapısal olarak dinamiklerdir. Genellikle danışmansız öğrenme kurallarının uygulandığı yapılarda kullanılmaktadır. Hopfield, Elman ve Jordan ağları bu ağlara örnek olarak verilebilmektedir [36]. Şekil 2.20.'de Hopfield ağ yapısı gösterilmektedir.

2.4.1.1.3. Öğrenme süreci

Öğrenme, yapay sinir ağlarındaki girdi-çıkış ilişkisini iyi bir şekilde anlamlandırabilecek en uygun ağırlıkların oluşturulması ile gerçekleşmektedir. Çalışma için elde bulunan datalardan yararlanarak bu ağırlıkların oluşumu gerçekleşmektedir. Yapay sinir ağlarının öğrenmesi danışmanlı (supervised), danışmansız (unsupervised) öğrenme ve takviyeli öğrenme olmak üzere üçe ayrılmaktadır.

Danışmanlı Öğrenme:



Şekil 2.21. Danışmanlı Öğrenme Yapısı [37]

Bu öğrenme metodunda sinir ağına datalar ve onların olası sonuçları verilmektedir ve eğitim gerçekleştirilmektedir. Bu verilen verilere göre işlem yapılması gereken veriler kıyaslanarak öğrenme ve hata oranları belirlenmektedir. Bu yöntemde öğrenme sonucunda yüksek oranlar elde edilene kadar denemeler yapılmaktadır. Ve doğruluk değeri istenen değere ulaştığında elde edilen ağırlık gerçek zamanlı kullanıma hazır hale gelmiş olmaktadır. Daha güvenilir ve yaygın olması sebebiyle en çok kullanılan öğrenme metodu danışmanlı öğrenme metodudur.

Danışmansız Öğrenme: Günümüzde merak uyandıran ve araştırma konusu olan bir konudur. Bilgisayarların insan yardımı olmaksızın öğrenebileceklerinin göstergesidir. Bu metotta öncesinde eğitim olmaksızın ağ kendi kurallarını oluşturarak sınıflandırmayı gerçekleştirmektedir.

Takviyeli öğrenme: Girişteki verinin ne olması gerektiğini, çıkış verisi bilemez. Girdilere göre bir çıkış üretilmelidir. Bu üretilen çıktı için ek takviyeler yapılır. Elde edilen çıktıya göre sonucun doğru ya da yanlış olduğu bilgisi verilir. Verilerle yanlış sonuçlar üretilirse ağırlıkların doğru sonucu üretmesi için işlemler güncellenir [38].

2.4.2. Derin öğrenme

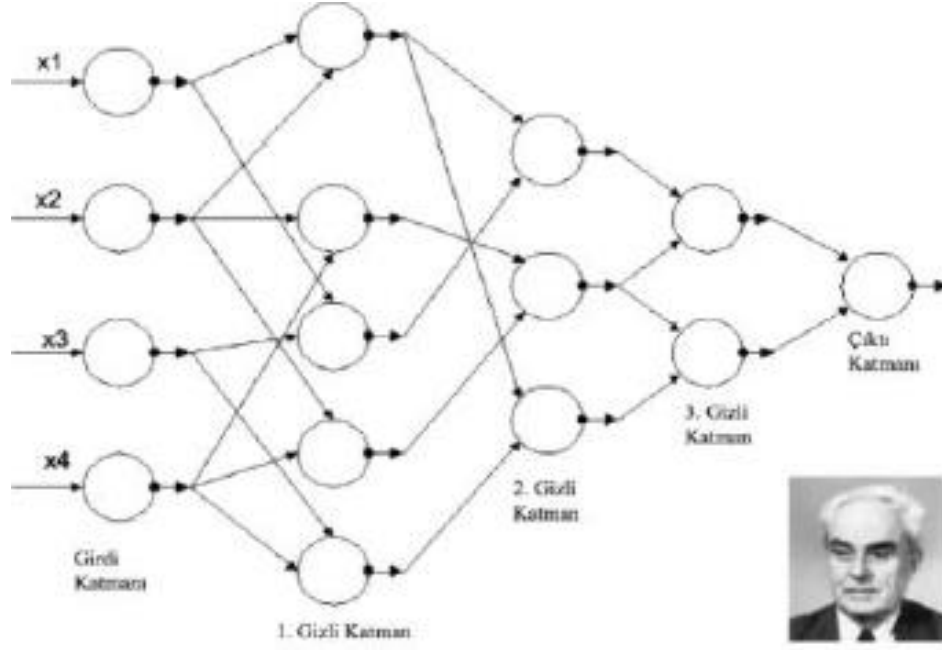
Yapay zekâ, makine öğrenmesi ve derin öğrenme kavramlarının tanımlanması ve sınıflandırılması konusu hala bir araştırma konusu olmaktadır. Genel manada hepsini

ayrı ayrı tanımlamak gerekirse, kullanılan datalarla eğitildiklerinde insanlar kadar öğrenebilen ya da elde edilen hatalara göre iyileştirme yapabilen sistemler yapay zekâ olarak tanımlanmaktadır. Makina öğrenmesi ise verilen datalara göre çeşitli benzetimlerle insanlardan daha iyi analizler yapan hatta kendi kendini eğiten sistemlerdir. Bunların yanında derin öğrenme büyük verileri çoklu katmanlar kullanarak makine öğreniminde kullanan ve tek seferde eğitim yapabilen sistemlerdir. Makine öğrenmesi kavramının alt dalı olarak gösterilmektedir.

Derin öğrenme sisteminde doğruluk değeri yüksek sonuçlar elde etmek için eğitimde kullanılacak olan özelliklerin seçimi büyük önem arz etmektedir. Bu özellikleri ortaya çıkarabilmek için çeşitli ön işlemler yapılmaktadır. Bunlar, boyut indirgeme, özellik seçme vb. olabilirler. Bu özelliklerin seçimi, maliyeti azaltmak, çalışma süresini kısaltmak ve iş zorluğunu hafifletmek açısından da önemli olmaktadır [39]. Özelliklerin temsili, örneğin veri bir görüntü ise piksel başına düşen yoğunluk değerlerinin bir vektörü, kenar kümeleri ya da özel şekiller gibi özellikleri göstermektedir. Derin öğrenmede özellik temsilleri ve özelliklerin evrelerinin öğrenilmesi için oluşturulmuş bir yapı bulunmaktadır. Bu yapı üst düzey özelliklerden yola çıkarak üretilen alt düzey özelliklere doğru hiyerarşik bir yapı oluşturmaktadır. Bu özellikler içinden veriyi en iyi temsil eden özelliklerin çıkarılması için etkin algoritmalar kullanılmaktadır. Bu da öğrenme doğruluğunun artması ve gelişmesi açısından oldukça önemlidir [40].

Derin öğrenme ağ yapısında, ardışık birçok katman kullanılmaktadır. Bu katmanlar doğrusal olmayan işlem birimleri içermektedir. Her bir katman kendisinden önceki katman çıktısını girdi olarak almaktadır. Bu algoritmalar denetimli (sınıflandırma) veya denetimsiz (desen analizi) şekilde olabilmektedir.

Denetimli derin beslemeli çok katmanlı yapay sinir ağları; bu öğrenme algoritması 1965 yılında Ivakhnenko ve Lapa tarafından yayınlanmıştır. Denetimli derin beslemeli çok katmanlı ilk öğrenme algoritmasıdır. Şekil 2.22.'de ağ yapısı gösterilmektedir.



Şekil 2.22. İlk derin ağ mimarisi ve Ivakhnenko [41].

Bu ağ mimarisi ile, en iyi özellikler istatistiksel yöntemlerle kullanarak çıkarılmakta ve bir sonraki katmana iletilmektedir. Geri yayılım işlemi kullanılmamaktadır. İki katman arasındaki veriler kullanılarak, gerçeğe yakın değerlere göre denklem oluşturmak üzere, bir çeşit regresyon işlemi olan en küçük kareler yöntemi uygulanmaktadır.

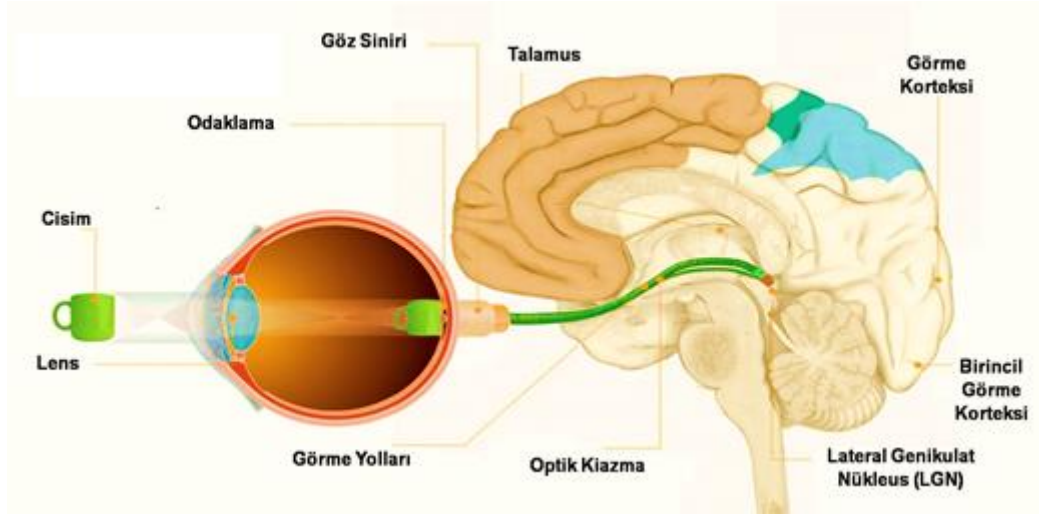
1979 yılında Fukushima, Omurgalı canlıların görsel sinir sistemlerini referans alarak ortaya attığı çalışmasında, "denetimsiz öğrenme" ile kendi kendini organize eden bir ağ geliştirilmiştir. Bu ağlar, günümüzdeki modern ağlara benzer yapıda çoklu bükülme ve havuz katmanları bulunmaktadır. Geri yayılım algoritmaları daha önceki yıllarda geliştirilmiştir. Fakat bu konuda ilk başarılı algoritmayı Yann LeCun ve arkadaşları posta kutusu yazıları üzerinde yaptıkları çalışma ile geliştirmişlerdir. Fakat bu çalışmadaki eğitim süreci 3 gün sürmektedir. Ardından Yann LeCun "LeNet" ağını kullanarak el yazısı rakamlarını (MNIST) sınıflandırmak için kıvrımlı ağlarla geri yayılımı birlikte uygulamıştır. Brendan Frey, Peter Dayan ve Geoffrey Hinton, uyanık-uyku (wake-sleep) algoritmasını geliştirmişlerdir. Yüzlerce gizli katman içeren ve 6'sı tamamen bağlı (fully connected), yapıda olan bir ağı eğitmeyi, başarmışlardır. Fakat bu eğitim 2 gün sürmüştür. 1997'de, Hochreiter ve Schmidhuber, daha önce ortaya attıkları, tekrarlayan sinir ağları (recurrent neural

networks) için Uzun Kısa Vadeli Bellek (long short-term memory) gibi konular üzerinde çalışmalar yapmışlardır. Anlatılan tüm bu çalışmalardan da görüldüğü gibi derin öğrenme çalışmalarında donanımsal yetersizlik ve hesaplama maliyetinin oldukça çok olması ve ağ mimari eğitimlerinin günlerce sürmesi sebebiyle 2000’li yıllara kadar destek vektör makinaları gibi yapılacak çalışmaya özgü daha basit yapıda çalışmalar tercih edilmiştir. 2000’li yıllara gelindiğinde ise grafik işlemci biriminin (GPU) hesaplamalarda kullanılması ile hesaplama hızı 1000 kat atmıştır. Ve makine öğrenim çalışmaları hız kazanmıştır [42]. 2000 yılında "Derin Öğrenme" ifadesi, Igor Aizenberg ve arkadaşları tarafından ilk kez ortaya atılmıştır [43]. Geoffrey Hinton ise 2006 yılında, çok katmanlı ileri beslemeli bir sinir ağının her iterasyonda bir katmanın eğitebildiğini göstermiştir. Her bir katmanı denetimsiz olarak kısıtlanmış bir Boltzmann makinesi ile eğitmiştir ve sonrasında denetimli geri yayılım yöntemi ile küçük önemli ayarlamalar yapılabileceğini göstermiştir [44]. 2012 yılında ağda ezberleme sorunun önüne geçen “dropout” adı verilen normalleştirme yöntemi kullanılmıştır. Dropout yöntemi ILSVRC-2012 ImageNet yarışmasında göstermiş olduğu başarı ile kendini kanıtlamaktadır.

2.4.2.1. Derin öğrenme mimarileri

Makine öğreniminin ilk ortaya çıktığı zamanlardan bu yana yapay zekâ ile yapılan çalışmalar ve konular çeşitlendikçe farklı konularla ilgili farklı yapay zekâ algoritmaları geliştirilmiştir. Bu da çeşitli derin öğrenme mimarinin oluşmasını sağlamıştır. Bu mimarilerin başlıcaları; konvolüsyon sinir ağı, tekrarlayan sinir ağı, uzun ve kısa vadeli hafıza ağları, sınırlı bozaltman makineleri, derin inanç ağları ve derin oto-kodlayıcılarıdır.

2.4.2.1.1. Konvolüsyon sinir ağı (Convolutional neural network (CNN))

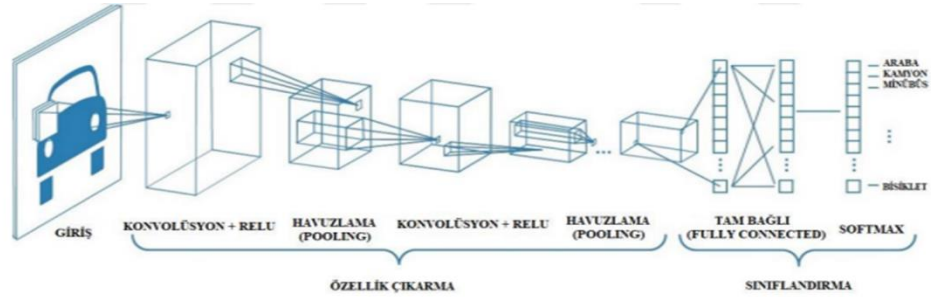


Şekil 2.23. İnsan Görme Sistemi

İnsanların görme sistemlerinde, birincil görme korteksi olarak da bilinen V1 beynin ilk alanıdır. Görüntüler, retinanın (göze ulaşan ışık ve gözün arkasındaki ışığa duyarlı doku) uyarılması sonucunda oluşmaktadır. Retinadaki nöronlar görüntünün basit bir şekilde bazı ön-işlemlerden geçmesini sağlamaktadır. Görüntü daha sonra optik sinirden ve lateral genikülasyon çekirdeği olarak adlandırılan beyin bölgesinden geçmektedir. CNN V1'in üç özelliğinden esinlenilerek oluşturulmuştur. Bunlar;

- V1, retinadaki görüntünün yapısını yansıtan iki boyutlu bir yapıya sahiptir. Konvolüsyonel ağlar bu özellikten faydalanarak özelliklerinin iki boyutlu haritalarla tanımlanmasını sağlamıştır.
- V1 birçok basit hücreden oluşmaktadır. Basit bir hücrenin aktivitesi; küçük, konumsal olarak belli bir yere yerleştirilmiş alıcı alandaki (receptive fields) görüntünün doğrusal bir fonksiyonu tarafından karakterize edilebilmektedir. Bir konvolüsyon ağının bulucu birimleri, basit hücrelerin bu özelliklerini taklit edecek şekilde tasarlanmıştır.
- V1 basit hücrelerin yanında karmaşık hücreler de içermektedir. Bu hücreler, basit hücreler tarafından tespit edilenlere benzer özelliklere cevap verir, ancak karmaşık hücre, özelliğin konumunda küçük kaymalar için değişmezdir.

Bu, konvolüsyon ağlarının havuzlama (pooling) katmanlarının oluşturulmasında etkili olmuştur.

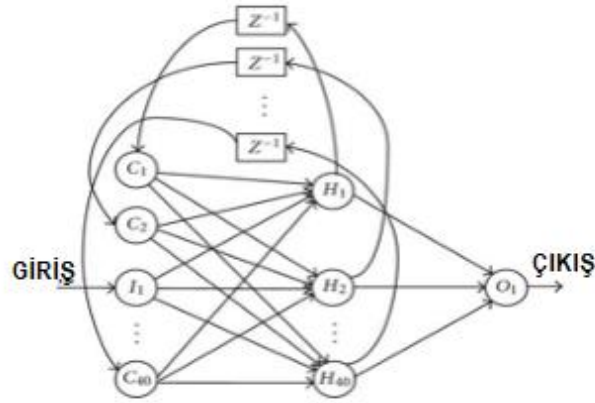


Şekil 2.24. CNN Modeli Ağ Yapısı

CNN, daha çok sınıflandırma ve tespit işlemlerinde kullanılmaktadır. Şekil 2.24.'de de gösterildiği gibi çok katmanlı sinir ağı içerisinde, konvolüsyon katmanları, tam bağlı katman, aktivasyon katmanı, sınıflandırıcı katman, havuzlama katmanı gibi katmanlar bulunmaktadır. 2012 ve 2014 yıllarında düzenlenen Büyük Ölçekli Görsel Tanıma yarışmasında en başarılı sonuçlar Konvolüsyonel sinir ağları ile elde edilmiştir [45].

2.4.2.1.2. Tekrarlayan sinir ağları (Recurrent neural network (RNN))

Sinir ağı içerisindeki yapıların yönlendirilmiş şekilde birbirine bağlandığı bir ağ modeli sunmaktadır. Zamansal olarak çeşitli olayların incelenmesine olanak tanımaktadır. İlk olarak 1990 yılında Jeff Elman tarafından tasarlanmıştır. Bu çalışmada; konuşma akışı içerisindeki kelimelerin sözcük yapıları ile ilgili çalışmada fiil-isim ayrımı gerçekleştirilmiştir. Bunun yanında canlı-cansız, insan-hayvan, avcı-yırtıcı gibi sınıflama işlemleri de yapılmıştır.

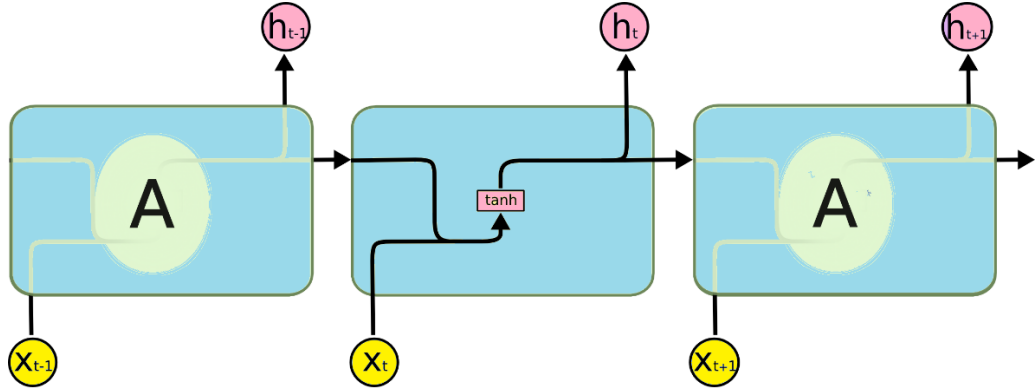


Şekil 2.25. Tekrarlayan Sinir Ağı Yapısı

Şekil 2.25.'de RNN yapısı basitçe gösterilmektedir. Zaman serisi olarak ifade edersek t anındaki girdi aynı zamanda $t-1$ anının bilgisini de içinde barındırmaktadır. Bu yapı bize bu ağ yapısında bir hafızası olduğunu da göstermektedir. Bu hafıza kabiliyeti sayesinde, sinir ağına girdi olarak; metin, konuşma, zaman serileri, vb. verilebilir. Dil çeviri işlemi, konuşma analizi, istatistiksel çıkarım ya da çeşitli finansal hesaplamalar, vb. gibi birçok farklı, ard arda ilerleyen konular için RNN kullanılabilir.

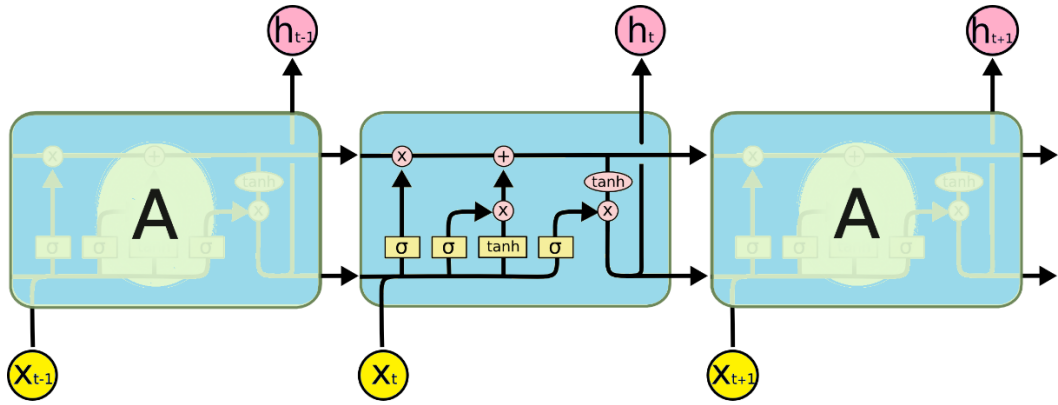
2.4.2.1.3. Uzun kısa vadeli hafıza ağları (Long short-term memory (LSTM))

Ard arda ilerleyen olaylar arasında bazı boşluklar olabilmektedir. Bu durumlarda Tekrarlayan sinir ağı yapısı yetersiz kalmaktadır. LSTM ağları bu sorunun önüne geçebilmek için 1997 yılında Schmidhuber tarafından tasarlanmıştır. [46]. Örneğin, bir sonraki kelimeyi tahmin etmek isteyen bir modelde, “su akar” cümlesindeki son kelimeyi bulmak RNN için basittir. Fakat karmaşık yapılar olduğunda mesela, “O Türkiyeden geldi, çok güzel Türkçe konuşuyor” ifadesindeki “Türkçe” kelimesini bulmak RNN için bir önceki iterasyon bir şey ifade etmediğinden zor olmaktadır. Ön bilgi yani Türkiye ifadesi sadece bir önceki cümlede değil daha da uzak da olabilirdi. RNN yönteminde bu sorunları gidermek için, Tanh aktivasyon yerine ReLU kullanılması, çeşitli ön işlemler gibi çalışmalar yapılmıştır. Fakat en etkili olarak sorunu LSTM çözmektedir.



Şekil 2.26. RNN ağ yapısı

Şekil 2.26.'da RNN ağ yapısında tekrarlayan tek bir katman olduğu gösterilmektedir.



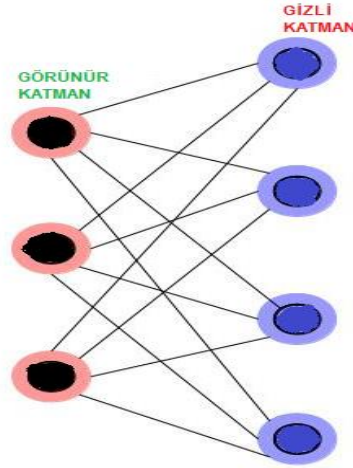
Şekil 2.27. LSTM ağ yapısı

Şekil 2.27.'de ise LSTM ağ yapısında tekrarlayan dört farklı katman olduğu görülmektedir. Bu katmanlar, üç sigmoid katmanı ve tanh katmanıdır. Sigmoid katmanı, 0-1 arasında değerler çıkarmaktadır. Her bir bileşenden geçmesi gereken bileşeni göstermektedir. 0 değeri hiçbirşeye izin vermemeyi ifade ederken 1 değeri ise herşeye izin vermeyi ifade etmektedir. Siner ağındaki bir hücrenin kontrol altında tutulması ve korunması için bu sigmoid kapıları kullanılmaktadır.

LSTM ağları, el yazısı tanıma [47], ses tanıma [48] gibi RNN'de verilen örnekler gibi birçok farklı alanda kullanılmaktadır.

2.4.2.1.4. Sınırlı boltzmann makineleri (Restricted boltzmann machine (RBM))

Girişe verilen veriseti üzerinde olasılıksal dağılımları öğrenebilmektedir [49]. Sınıflandırma [50], boyut küçültme [51], konu modelleme [52] ve danışıklı filtreleme [53] işlemleri yapabilmektedir.

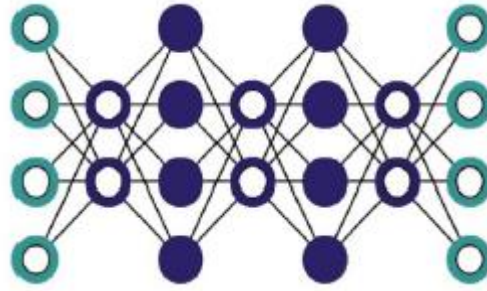


Şekil 2.28. Sınırlı Boltzmann Makineleri Ağ Yapısı

Şekil 2.28.'de gösterildiği gibi iki katmanlıdır. Düğümlerde hesaplamalar yapılmaktadır. Düğüm nöronu ifade etmektedir. Bir düğüm başka bir düğüm ile bağlanabilmektedir. Fakat aynı katmanda olanlar birbiriyle bağlanamazlar. Katmanlar arasında bir geçiş olmadığı için kısıtlı boltzman makineleri ismi verilmiştir. Girdi hesaplanır daha sonrasında bu girdi bir sonraki düğüme random şekilde ya iletilir ya da iletilmez [54]. Boltzman makinelerinde gözetimsiz öğrenim yapılmaktadır.

2.4.2.1.5. Derin inanç ağı (Deep belief network (DBN))

Yönlendirilmiş hem de doğrulanmamış kenarları olan grafiksel çok katmanlı bir sinir ağı modelidir. Katmanlar birbiri ile bağlantılıdır. Başka bir ifadeyle RBM yığını olarak da tanımlanabilmektedir. Şekil 2.29.'da da görüldüğü gibi RBM katmanları önceki ve sonraki katmanlarla bağlantı kurmaktadır. Son katman da Softmax katmanı ile sınıflandırma ile, denetimsiz bir öğrenme gerçekleştirilmektedir.

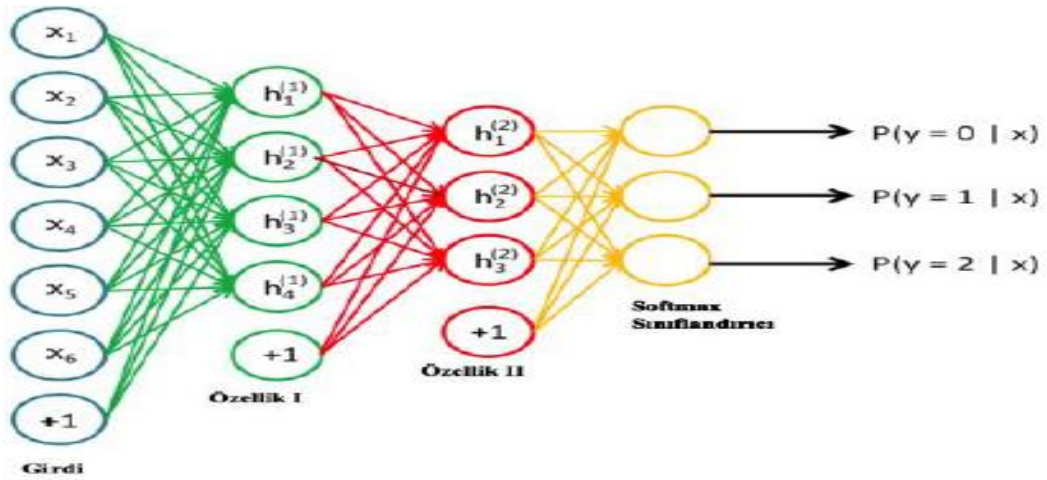


Şekil 2.29. Derin İnanç Ağları (DBN)

DBN, görüntü tanıma ve tanımlama gibi çeşitli çalışmalar yapılmaktadır. Ayrıca insan hareketi analizi, çeşitli hareket dizilerini hibrit bir şekilde kullanarak ve hareket yakalama sırasında kaybolan verilerin gösterimi gibi çalışmalar yapılmıştır [55]

2.4.2.1.6. Derin oto-kodlayıcılar (Deep/Stacked autoencoder (DAE))

Katmandaki çıktılar ardışık katmanın girişlerine bağlandığı çok katmanlı bir sinir ağıdır [56]. Veri kümesi için boyut indirgeme amacıyla bir şifrelemeli öğrenme sistemi sunmaktadır. Denetimsiz öğrenmede kullanılan özel bir yapay sinir ağıdır.



Şekil 2.30. Derin Oto-Kodlayıcı Ağ Yapısı

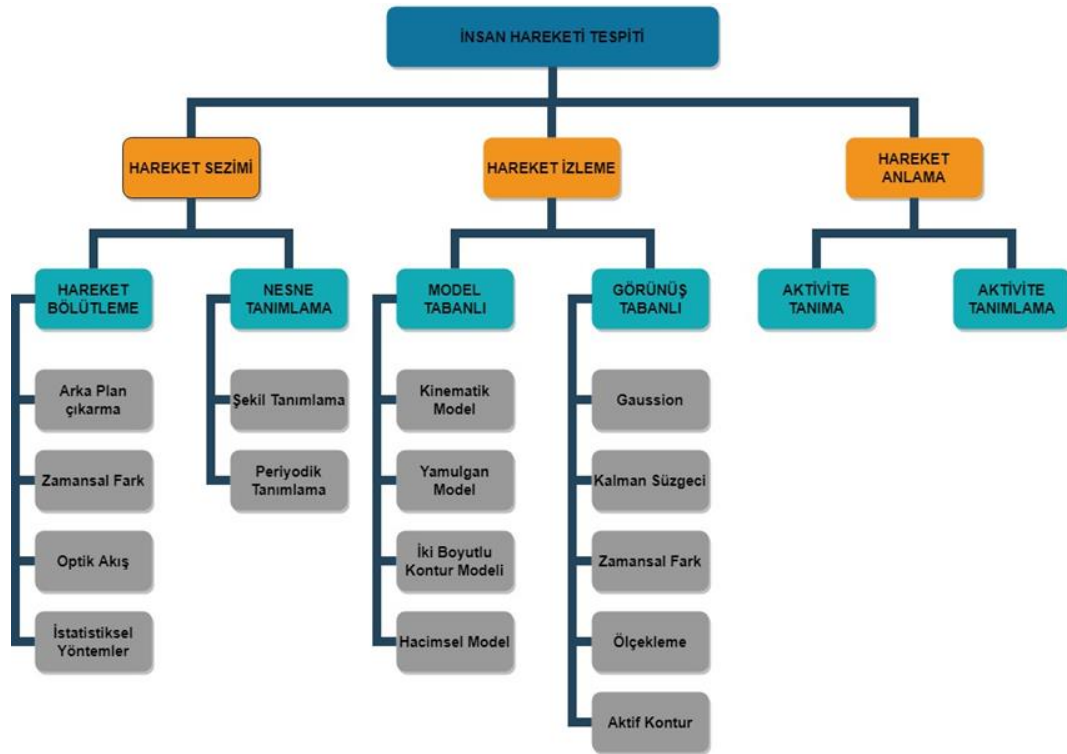
Her katman a gelen girdiler bir sonraki katmana geçişte şifrelenmektedir. Şekil 2.30.'da ağ yapısı gösterilmektedir. Özellik I olarak nitelendirilen yapı girdiden elde edilen şifreyi ifade etmektedir. Bu şifre diğer katmana geçerek özellik II şifresini

oluşturmaktadır. Son olarak softmaxden geçerek sınıflandırma işlemi gerçekleştirilmektedir.

DAE, veri kümesini sıkıştırarak ve hatayı en aza indirgeyerek en iyi öğrenme performansını amaçlamaktadır. İleri beslemeli bir sinir ağıdır. Sıkıştırma yaptığından veri yükünü azaltmakta ve öğrenme hızını ve performansını arttırmaktadır.

Spam maillerin sınıflandırılması [57], kumaş hatası tespiti [56], Konuşma spektrogram kodlaması üzerindeki bir çalışmada [58], Anormallik tespiti çalışmalarında [59] kullanılmaktadır.

2.5. İnsan Hareket Tespiti



Şekil 2.31. İnsan hareketi tespiti aşamaları

Makine öğrenmesi ve görüntü işleme konularında insan hareketinin tespit edilmesi oldukça önemli bir alan olarak kabul görmüş ve üzerinde halen daha birçok çalışma yapılmaya devam etmektedir. Genel anlamda hareket analizi sistem yapısı şekilde de gösterildiği gibi; hareket sezimi, izleme ve hareketi anlama olmak üzere 3 başlıkta

incelenmektedir. Hareket sezimi; video dizisinde insan var mı? sorusuna cevap vermektedir. Hareket tespit etmekteki önemli ilk adımı oluşturmaktadır. İzleme; hareketin nerede olduğu ve insanın takibini ifade etmektedir. Hareketi anlama; insanın yaptığı hareketi etiketleme yani yaptığı eylemi tespit etmeyi ifade etmektedir.

2.5.1. Hareket sezimi

2.5.1.1. Hareket bölütleme

Hareket bölütleme; hareket eden nesnenin ortamdaki yalıtılmasını ifade etmektedir. Bölütleme algoritması görüntü işleme çalışmalarındaki ortam kalibrasyonu sorunları göz önünde bulundurularak seçilmelidir. Bu problemleri çözmek amacıyla, 3 boyutlu kameralar, RGB-D sensörlerin kullanımı günümüzde yaygınlaşmaktadır. Bu derinlik kameraları, algoritma kombinasyonu hızı, doğru ve hızlı tanımlama oklüzyonlardan kaynaklı olan sorunları ortadan kaldırmaları açısından avantajlar sağlamaktadır [60].

Fakat bunların yanında bu sistemler genellikle laboratuvar ortamında test edilebilmişlerdir, pahalıdır ayrıca etkili mesafe sınırlamaları ve gürültü problemleri de bulunmaktadır. Bu açıdan kullanılabilir, pratik, ucuz ve uyarlanabilir bir çalışma için standart bir RGB kamerası kullanımı önemli olmaktadır [61].

Hareket bölütleme işlemi için, en temel ve basit çözüm arka plan çıkarma işlemidir. Bir öncesinde gelen çerçeveyi arka plan olarak kabul eden ve buna göre hareketli kısmı tespit eden yöntem ise zamansal fark yöntemi olarak tanımlanmaktadır. Arka planı çok değişken olan video yapıları için etkili bir yöntemdir. Ayrıca 2 boyutlu hareketlerin belirlenmesinde kullanımı oldukça yaygın olan optik akış yöntemi de bölütleme için kullanılmaktadır. Ard arda gelen çerçevelerin ışık, arka plan gibi çeşitli parametrelerin istatistiksel analizlerinden yararlanarak da bölütleme gerçekleştirilebilmektedir [62].

2.5.1.2. Nesne tanımlama

Çok sayıda nesnenin birbirinden ayrılması işleminin diğer adı da nesne tanımlama olarak ifade edilmektedir. Nesnelere şekillerine ya da hareketlerine göre tanımlanabilirler. Bu tanımlama işlemleri için görüntü işleme kapsamında birçok farklı teknik kullanılmaktadır. Periyodik olarak video dizisinde tekrar eden hareketlerin analizi de bu tanımlamayı sağlamaktadır. Örneğin, bir insan yürüme eylemi yaptığında her zaman elini öne arkaya periyodik olarak sallar.

2.5.2. Hareket izleme

2.5.2.1. Model tabanlı

Hareket bölütleme sonucu elde edilen görüntü çerçeveler boyunca izlenmelidir. Bu izleme işlemi için çeşitli modeller geliştirilmiştir. Bunlar; insan vücudunun serbestlik derecesini kullanarak izleme yapan kinematik model, optik akış bilgisinden yararlanan eklemli yamulgan model. İki boyutlu kontur model ise insan vücudunu parçalara ayırarak kitle merkezi ya da eksenlerin oranlarına göre izleme gerçekleştirmektedirler. İnsan iskelet yapısını ele alarak bu iskelet yapısının birbirine göre konumlarını kullanan çöp adam modeli de bulunmaktadır. Çöp adam modelinin daha gelişmiş bir versiyonu olan hacimsel modelde ise küre, silindir, konilerle vücut bölümleri temsil edilmektedir. Bu model görüş açısından daha bağımsız yapıdadır fakat hesaplaması karmaşık olduğundan işlem yükü fazladır.

2.5.2.2. Görünüş tabanlı

Görünüş tabanlı yaklaşım ile model tabanlı yaklaşım arasındaki fark; görünüş tabanlı yaklaşımda izlenecek nesnenin rengi ve doku özelliklerinden yararlanılmakta iken model tabanlı yaklaşımda ise hareketin ön bilgisinden yararlanılmasıdır. Bu bilgiler, Şekil 2.31.'de belirtildiği gibi sıralanabilmektedir.

2.5.3. Hareket anlama

2.5.3.1. Aktivite tanıma ve tanımlama

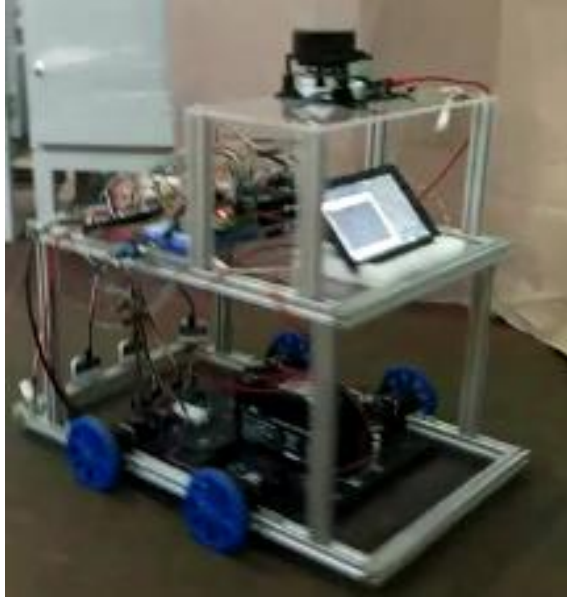
Hareketi anlama işlemi hareketin belirlenmesi ve o harekete uygun özelliğın harekete etiketlenmesi ile olmaktadır. Bu tanıma işlemi, önceden belirlenmiş dataya uygun olarak eğitim ile öğrenme işleminin gerçekleşmesi olarak düşünülür ise tanımlama işlemi ise elde edilen modele göre gerçek zamanlı sistemde onu etiketlemek olarak adlandırılabilir. Tanıma ve tanımlama işlemleri sadece yapay zekâ, derin öğrenme değil matematiksel çeşitli yöntemler kullanılarakda yapılabilmektedir. Örneğın, ard arda gelen olayların sıralanmasında kullanılan istatiksel bir yaklaşım modeli olan saklı markov modeli yürüyüş stili ve insan aktivitesi tanıma için kullanılmaktadır [63].

Tez çalışmasında da insan hareket tespit aşamaları Şekil 2.3.1.'de belirtildiği şekilde gerçekleştirilmiştir. Hareket seziminde, temel olarak kullanılan arka plan çıkarma optik akış yöntemleri uygulanmıştır. Hareket izleme aşaması için, yıldız iskelet modeli kullanılmıştır. Yıldız iskelet modeli davranış tespit etme işlemi için işlem yükünü azaltması, donanımsal gereksinimlerinin daha az olması ve tespit etme işleminin daha hızlı gerçekleştirilmesi gibi avantajlarıyla diğer yöntemlere göre tercih edilmiştir [60]. Hareketin anlaşılması için ise doğruluğın kıyaslanması ve gerçek ortama uyumlu stabil çalışabilen sistem önerisi sunabilmek için, matematiksel yöntem ve derin oto kodlayıcı yapay zeka yöntemi uygulanmıştır [58,59].

BÖLÜM 3. MATERYAL VE YÖNTEM

3.1. Materyal

Arařtırmada, Őekil 3.1.'de gsterilen Sakarya Uygulamalı Bilimler niversitesinde oluřturulmuř deneysel otomatik ynlendiricili ara kullanılmıřtır.



Őekil. 3.1. SUB Deneysel OYA

Cihaz, RPLIDAR 360° lazer tarayıcılı sensre ek olarak etrafındaki engelleri ultrasonik sensrler ile de kontrol etmektedir. RİS sisteminin kullanıldıđı kamera ve lidar verileri Raspberry pi 3b + dan iřlenen veriler STM32'ye iletilerek aracın kontrol sađlanmaktadır. 2 tekeri birbirine seri bađlı olacak Őekilde 4 tekerlidir. Besleme kaynađı iin 12 volt ak kullanılmaktadır.

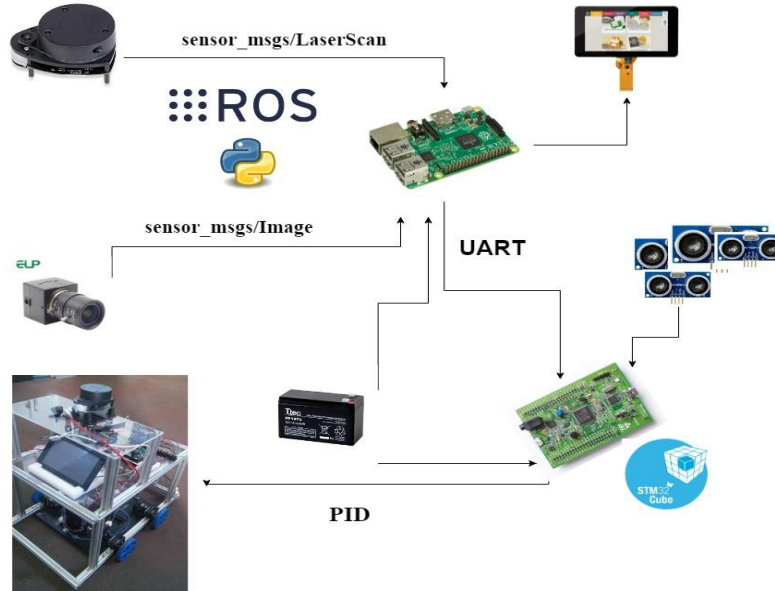
3.2. Yöntem

3.2.1. Kullanılan araç-gereçler



Şekil 3.2. Simülasyon çalışmaları sistem diyagramı

RPLIDAR 360° lazer tarayıcı sensörü ile harita alma işleminin ardından simülasyon çalışmalarının tamamı ASUS i7 işlemcili CPU laptop ile RİS içerisinde gerçekleştirilmiştir. Şekil 3.2.'de sistem diyagramı gösterilmektedir.



Şekil 3.3. Gerçek ortam sistem diyagramı

Şekil 3.3.'de oluşturulan sistem diyagramında gösterilen malzemeler gerçek ortamda yapılan çalışma için kullanılmıştır. Raspberry pi 3B+ mikro bilgisayarı, sensörlerden RİS ortamında veri almaktadır. Sensörlerden aldığı verileri işleyip anlamlı değerlere dönüştürmekte ve UART protokolü ile STM32F4 mikrodenetleyiciye göndermektedir. Sistemde 3 çeşit sensör bulunmaktadır. Bunlar; kamera, RPLIDAR 360° lazer tarayıcı ve ultrasonik sensörlerdir. Kamera ve lidar, RİS ortamında sensor_msgs/Image ve sensor_msgs/LaserScan yayınları ile verilerini raspberrypiye iletmektedirler. Ultrasonic sensör ise verilerini doğrudan STM32F4'e iletmektedir. Raspberrypi'ye bağlı olan ekran ile RİS içerisindeki yayınlar takip edilebilmekte ve davranış tespit etme çıktıları alınabilmektedir.

3.2.2. Kullanılan programlama dilleri

Araştırmada robot işletim sistemi içinde bulunan programlama dilleri de olan python ve C++ dilleri kullanılmaktadır. Simülasyon çalışmaları RİS (gazebo, rviz) üzerinden yapılmaktadır. Davranış tespiti uygulamalarının tamamı python dili kullanılarak gerçekleştirilmektedir.

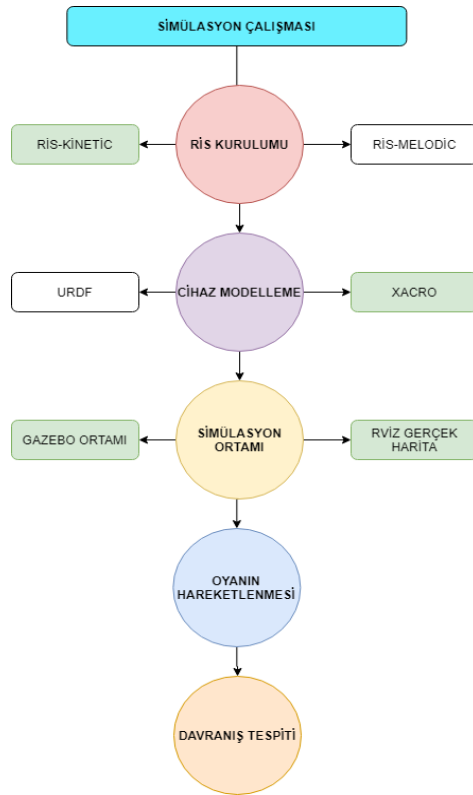
3.2.3. Simülasyon çalışmaları

Çalışmaların tamamı öncelikle simülasyon ortamında test edilmiştir. Simülasyon işlemleri robot işletim sisteminde gerçekleştirilmiştir. Simülasyon işlemlerinin yapılmasının amacı gerçek ortamda ortaya çıkacak problemleri en aza indirmektir. Öncelikle Linux ortamına RİS kurulmuştur. Ardından modellenen robot ile hedeflenen çalışma simüle edilen ortamda test edilmiştir. Aşağıda Şekil 3.4.'de hastane koridorlarını temsil eden insanların bulunduğu simülasyon ortamı gösterilmektedir.



Şekil 3.4. Hastane koridorlarını temsil eden simülasyon ortamı RViz haritası ve Gazebo çizimi

Simülasyon işleminde RİS-Gazebo ve RViz ortamları birlikte kontrol altında tutulmaktadır. Aşağıda Şekil 3.5.'de verilen şemada görüldüğü gibi aşamalarla simülasyon çalışmaları sırasıyla tamamlanmıştır.



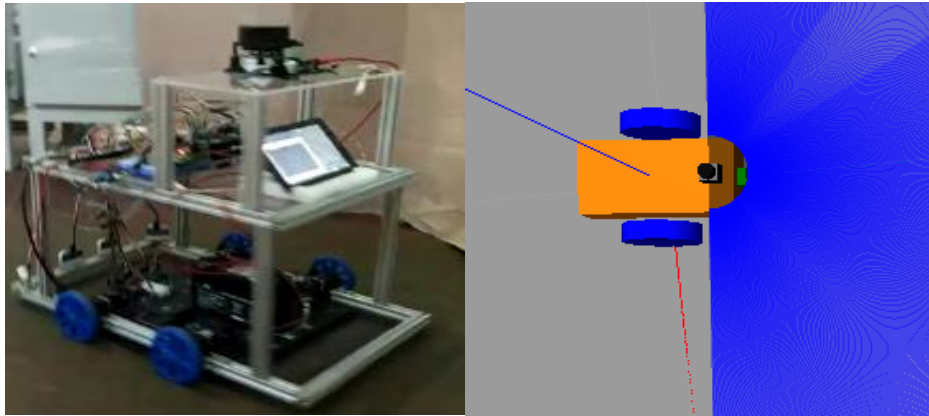
Şekil 3.5. Simülasyon çalışmaları akış diyagramı

3.2.3.1. RİS kurulum işlemleri

RİS Linux ortamında kullanım açısından daha kolay ve kullanışlıdır. Ayrıca örneğin, ubuntu 16 için RİS-kinetic kullanılırken ubuntu 18 için RİS-melodic kullanılmaktadır. Çalışmalar simülasyon aşamasında ubuntu 16 yüklü laptop ile gerçekleştirilmiştir. Ve ubuntu kinetic kurulumu gerçekleştirilmiştir. Kurulum için RİS in kendi sitesinden aşama aşama talimatlar takip edilmiştir [64]. Kurulum işlemi sonrasında Gazebo ve rviz otomatikmen yüklenmektedir. Bu aşamalar tamamlandıktan sonra kullanılacak robot modellemesi gerçekleştirilmiştir.

3.2.3.2. Cihazın modellenmesi

RİS içerisinde gazebo ve rviz ortamlarında kullanılacak olan robot kişinin gerçekte oluşturduğu robot şeklinde çizilebilmekte ya da farklı aynı benzer yapıda olan bir robotla da test edilebilmektedir. Cihazın modellenmesi işlemi urdf ya da xacro uzantılı şekilde yapılabilmektedir. Xacro, bazı işlemleri bir fonksiyon gibi kullanabilmeyi sağlamaktadır. Bu da yeni şekillerin daha rahat oluşturulmasını sağlamaktadır. Bu çalışmada, aşağıda Şekil 3.6.'da simülasyon testi için kullanılan robot modeli gösterilmektedir. Model Xacro uzantılı yapıda oluşturulmuştur.



Şekil 3.6. Gerçek Cihaz ve Simülasyonda kullanılan model

Gerçek cihazda olduğu gibi kamera ve RPLIDAR sensörleri bulundurduğundan simülasyon çalışmaları için uygundur. Ayrıca teker sayısındaki farklılık simülasyonu

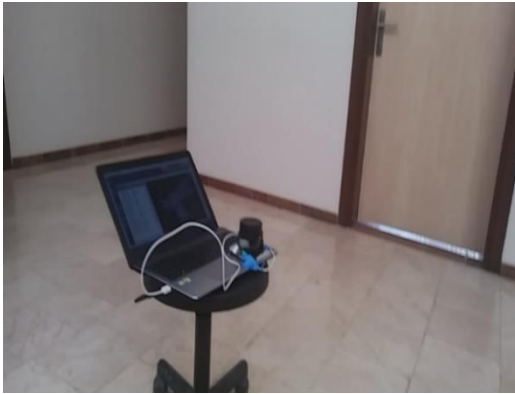
etkilememektedir. Çünkü tekerlerin kontrolü sağlanırken sağ ve soldaki tekerler birbirine bağlı olduğundan aynı hareketleri yapmaktadırlar.

3.2.3.3. Simülasyon ortamının oluşturulması

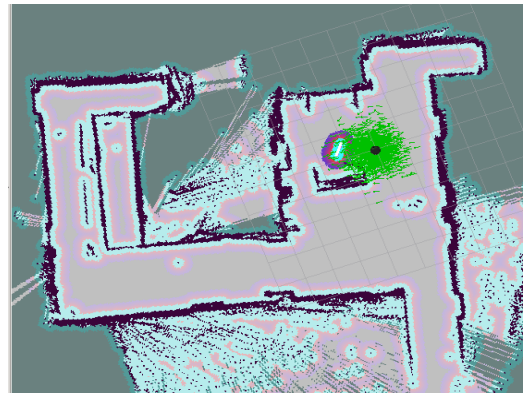
Tez çalışması için belirlenen test alanında öncelikle RViz ortamında harita alma işlemi gerçekleştirilmiştir. Haritanın çıkarılması işlemi RPLİDAR 360° lazer tarayıcı ve laptop kullanılarak yapılmıştır.

3.2.3.4. RPLİDAR ile haritalama işlemi

Farklı birçok konu için eklentileri kullanarak alınan bilgileri görselleştirmenize izin veren bir RİS grafik ara yüzü olan RVIZ ortamında RPLIDAR 360° lazer tarayıcı ile SLAM (Eş Zamanlı Konum Belirleme ve Haritalama) algoritması kullanılarak harita alma işlemi Şekil 3.7.'de gösterildiği gibi gerçekleştirilmiştir. SLAM algoritmaları, kendi kendini kontrol edip karar verme ve verdiği kararları uygulama yeteneğine sahip olan bir robotun hiç bilinmeyen bir ortama bırakıldığında eş zamanlı olarak kendi lokasyonunu ve ortamı tanması ve bu doğrultuda işlem yapmasını konu almaktadır [65].



Şekil 3.7. RPLIDAR 360° ile haritalama

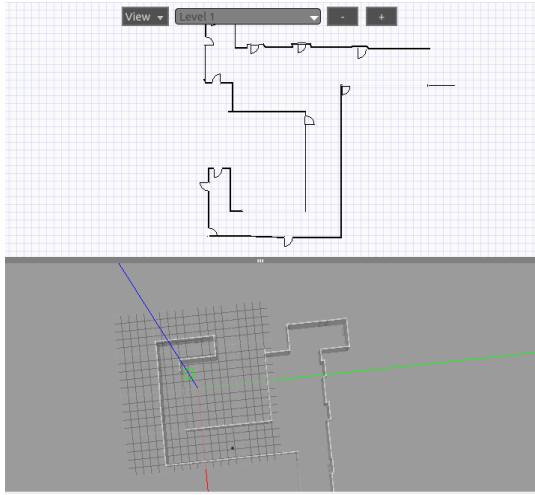


Şekil 3.8. Çıkarılan haritanın RViz'de gösterimi

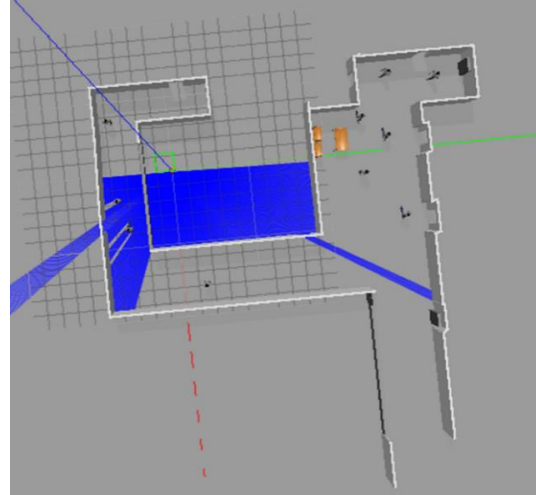
Harita alındıktan sonra kaydedilerek RViz'e eklenmektedir. RViz de aynı harita üzerinde cihazı hareket ettirebilmektedir. Şekil 3.8.'de cihazın harita üzerinde hareketi gösterilmiştir.

3.2.3.5. Gazebo simülasyon ortamının oluşturulması

SLAM algoritması kullanılarak oluşturulup kayıt altına alınan harita referans alınarak ve Şekil 3.9.'da gösterildiği gibi eş boyutlarda eş zamanlı çalışmaların test edilebileceği bir RİS-Gazebo ortamı oluşturulmuştur. Ortama Şekil 3.10.'da da görüldüğü gibi insanlar ve nesnelere yerleştirilerek gerçek ortamın daha iyi simüle edilebilmesi sağlanmıştır.



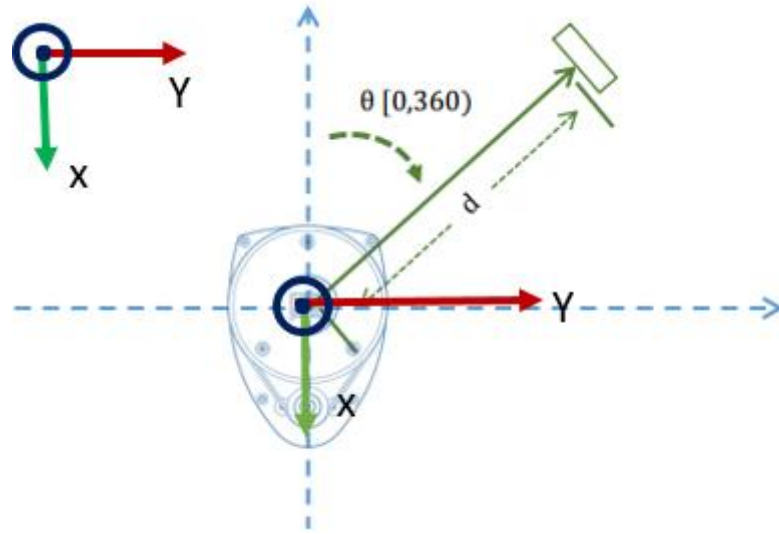
Şekil 3.9. Gazebo ortamında alınan haritanın çizimi



Şekil 3.10. Test simülasyon ortamının son hali

3.2.3.6. OYA'nın hareketi

Gazebo ortamının hazırlanmasının ve OYA modelinin oluşturulmasının ardından çalışma için gerekli olan, engelden kaçarak ilerleme algoritması oluşturulmuştur. Algoritmaya göre OYA, simülasyon ortamında rplidar ile önüne çıkan engeli fark ederek ona çarpmadan yönünü değiştirmektedir.



Şekil 3.11. RPLİDAR 360 Lazer Tarayıcı

OYA, lazer tarayıcıdan karşısına çıkan engelin nerede ve ne kadar mesafede olduğunu öğrenebilmektedir. Şekil 3.11.'de gösterildiği gibi cihaza yerleştirilen konumuna göre örneğin, 315° - 360° ve 0° - 45° arası ön kısım olarak belirlenmiştir. Bu bölgeye yarım metreden daha yakında bir engel geldiğinde cihaz durmaktadır. 45° - 90° arası sağ olarak belirlenirken, 270° - 315° aralığı ise sol olarak belirlenmektedir. Cihaz sağında bir engel gördüğünde sola, solunda bir engel gördüğünde ise sağa doğru yönelmektedir. Her iki yanında engel olduğu durumlarda aradaki mesafesi en az olana göre hareket etmektedir.

3.2.3.7. Gerçek zamanlı davranış tespit etme

Araştırmada davranış tespit etme işlemi OpenPose algoritması ile elde edilen insan iskelet sistemindeki verilere dayanarak yapılmıştır. Eklem koordinatlarının birbirine göre konumunu referans alan matematiksel yöntem ve gönüllü deneklerden elde edilen veri setinin derin sinir ağları ile eğitilmesiyle oluşan yapay zekâ yöntemiyle davranış tespiti iki farklı şekilde gerçekleştirilmiştir. Bu yöntemlerin simülasyon ortamında çalışmaları RİS-Gazebo ortamında gerçek zamanlı olarak test edilmiştir.

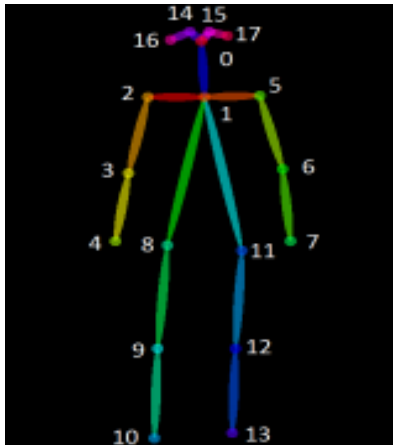
3.2.3.7.1. OpenPose algoritması



Şekil 3.12. Gines Hidalgo ve Hanbyul Joo tarafından Openpose Algoritmasının Gösterimi

OpenPose algoritması, üç boyutlu kinect kameralarla da yapılabilen insan iskeleti çıkarma işlemini, herhangi bir kamera ile yapabilmektedir. Kısmi İlgi Alanları olarak adlandırılan parametrik olmayan bir gösterim kullanarak oluşturulmuştur. Görüntüde kişi sayısına bağlı olmadan yüksek doğruluk ve gerçek zamanlı performans sağlanmaktadır [20]. CMU Panoptic Studio veri kümesi kullanılarak oluşturulmuştur. Şekil 3.12.'de araştırmacılardan iki tanesi kendileri üzerinde OpenPose algoritmasının çalışmasını göstermişlerdir.

3.2.3.7.2. Matematiksel yöntem



Şekil 3.13. OpenPose iskelet eklem bölgeleri

$$\begin{aligned} X_{frame1} &= (\text{joint}(1) + \text{joint}(8) + \text{joint}(11)) / 3 \\ X_{frame2} &= (\text{joint}(1) + \text{joint}(8) + \text{joint}(11)) / 3 \\ Y_{frame1} &= (\text{joint}(1) + \text{joint}(8) + \text{joint}(11)) / 3 \\ Y_{frame2} &= (\text{joint}(1) + \text{joint}(8) + \text{joint}(11)) / 3 \end{aligned}$$

$$\begin{aligned} A_{frame1} &= (\text{joint}(8) + \text{joint}(11)) / 2 - \text{joint}(1) \\ A_{frame2} &= (\text{joint}(8) + \text{joint}(11)) / 2 - \text{joint}(1) \\ H1 &= A_{frame2} / A_{frame1} \\ B_{frame1} &= ((\text{joint}(9) + \text{joint}(12)) - (\text{joint}(8) + \text{joint}(11))) / 2 \\ B_{frame2} &= ((\text{joint}(9) + \text{joint}(12)) - (\text{joint}(8) + \text{joint}(11))) / 2 \\ H2 &= B_{frame2} / B_{frame1} \end{aligned}$$

$$\begin{aligned} C_x &= X_{frame2} - X_{frame1} \\ C_y &= Y_{frame2} - Y_{frame1} \end{aligned}$$

$$C_x < 30 \text{ and } C_y < 20:$$

$$ty1 = \text{joint}(1)$$

$$ty8 = (\text{joint}(8) + \text{joint}(11)) / 2 \text{ # y kordinat}$$

$$ty9 = ((\text{joint}(9) + \text{joint}(12))) / 2$$

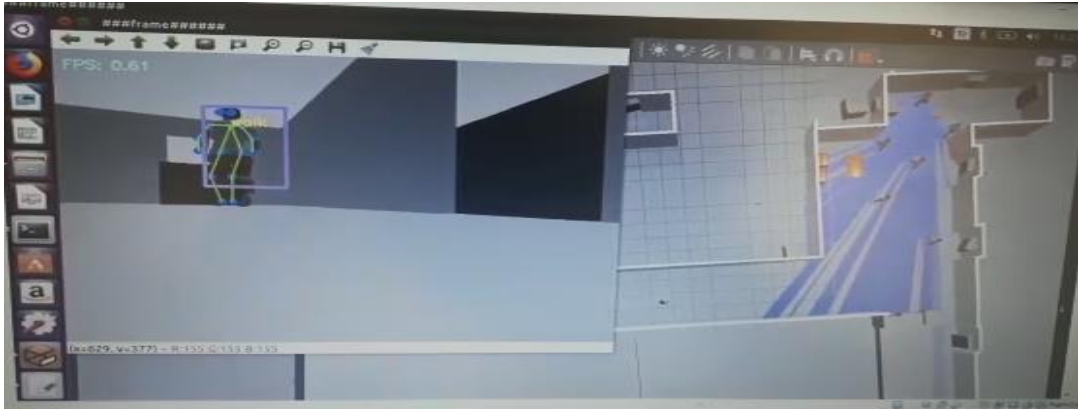
$$t = (ty8 - ty1) / (ty9 - ty8)$$

$$C_x < 30 \text{ and } C_y \geq 30:$$

İki frame arası farklar alınır. Aynı hesaplamalar yapılır. ty1 ty8 ty11 değerleri için

Şekil 3.14. Matematiksel Yöntem Referans İşlemleri

Matematiksel metot; Şekil 3.13.'de gösterilen eklem bölgelerinin x ve y ekseninde birbirlerine olan mesafe miktarlarına göre ve video içerisindeki önceki frame ve o anki frame arasındaki farklılıklar temel alınarak gerçekleştirilmiştir. Yöntem bazı basit eylemleri destekler, çünkü eylem şablonu esas olarak bağlantı noktasının konumuna göre bilgileri değiştirir ve ardından eylem şablonunu yapmak için bazı parametreler manuel olarak tasarlanır. Örneğin; yürürken, kişinin birleşme noktası bir bütün olarak bir yönde hareket eder ve üst gövdenin alt gövdeye oranı çok fazla değişmez. Sabit durma eylemi için; bağlantı noktaları sürekli bir süre için nispeten kararlıdır ve yatay ve dikey yönde büyük bir yer değiştirme söz konusu değildir [20].



Şekil 3.15. Gazebo ortamında matematiksel modelin test edilmesi işlemi

Algoritmanın oluşturulması ve Gazebo ortamında çalıştırılması ile kişilerin davranışları Şekil 3.15.'de gösterildiği gibi gerçek zamanlı olarak takip edilebilmiştir. Davranış tespiti konusunda oluşturulan tüm algoritmalar python dili ile oluşturulmuştur. Bu dosyalar RİS içerisinde bulunan pakette script içerisine kaydedilerek işlem yapılmıştır.

3.2.3.7.3. Yapay zekâ yöntemi

Yapay zekâ yöntemi için ise; öncelikle veri toplamak üzere 20 erkek 20 kadından oluşan 40 kişilik bir gönüllü grup ile 7 davranışın yapılacağı 5'er saniyelik videolar kaydedilmiştir. Bu videolarda davranışın tam yapılma anı olan 25 çerçeveye ait eklem bilgilerinin koordinatları, sonuna temsil ettiği hareketler eklenerek kaydedilmiştir. Yani 1 gönüllü denekten yaptığı 7 hareket için 175 çerçeve eklem

bilgisi alınmıştır. Toplam da 7000 verinin bulunduğu geniş bir veri seti elde edilmiştir.



Şekil 3.16. Verisetinde bulunan düşme, koşma, yürüme, oturma, kavga, eylem, bekleme eylemleri

Verilerde elde edilen davranışlar 0-6 arasında sınıflandırılarak veriseti oluşturulmuştur.

0 => Koşma

1 => Yürüme, şeklinde sınıflandırılmıştır. Verilerden elde edilen koordinatlar, koordinat sonlarına eklenek 1 veri parçası oluşturulmuştur. Aşağıdaki tabloda 3 adet veri örnek olarak gösterilmiştir.

0.48,0.24,0.5,0.31,0.45,0.31,0.42,0.44,0.45,0.41,0.55,0.31,0.57,0.41,0.55,0.39,0.46,0.57,0.45,0.74,0.44,0.87,0.52,0.57,0.52,0.74,0.54,0.87,0.48,0.22,0.49,0.22,0.46,0.22,0.51,0.2,2
0.39,0.22,0.42,0.28,0.37,0.28,0.38,0.41,0.37,0.5,0.45,0.26,0.49,0.37,0.46,0.44,0.4,0.48,0.4,0.65,0.42,0.74,0.45,0.48,0.45,0.65,0.45,0.76,0.39,0.2,0.4,0.2,0.38,0.22,0.42,0.2,6
0.39,0.24,0.42,0.28,0.37,0.31,0.38,0.41,0.37,0.52,0.45,0.28,0.49,0.37,0.46,0.46,0.4,0.48,0.4,0.65,0.42,0.74,0.45,0.48,0.45,0.65,0.45,0.78,0.38,0.22,0.4,0.22,0.38,0.24,0.42,0.22,6

Tablo 3.1. Verilerin Yapay Sinir Ağına Geçmeden Önce Düzenlenmiş Hali

Veri setinin tamamının hazırlanmasının ardından oluşturulan datalar derin sinir ağları ile eğitilmiştir. Eğitimde kabaca girdi verisinin sıkıştırılmış gösteriminden en iyi özelliklerin öğrenilmesini hedefleyen bir ileri beslemeli sinir ağı olan, her bir

katmandaki çıktılar ardışık katmanın girişlerine bağlanması ile elde edilen derin veya yığılıl oto-kodlayıcılar kullanılmıştır. Derin oto-kodlayıcılarla; potansiyel olarak benzer bir performans gösteren sığ ağa göre daha az birim içeren karmaşık verileri modelleyebilmekte, daha az veri kullanarak iyi oranlarda öğrenme elde edilebilmektedir [42]. Eğitilen model Şekil 3.17.'de gösterilmiştir.

```

6756/6756 [=====] - 1s 100us/step - loss: 0.0135 - accu
racy: 0.9950 - val_loss: 0.0295 - val_accuracy: 0.9855
Model: "sequential_1"

```

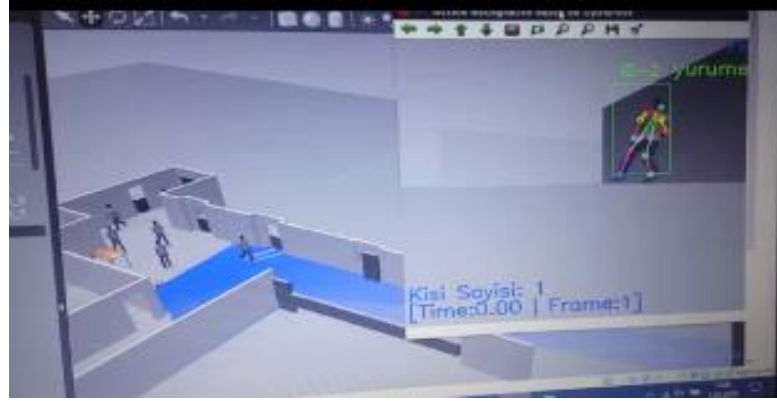
Layer (type)	Output Shape	Param #
dense_1 (Dense)	(None, 256)	9472
batch_normalization_1 (Batch Normalization)	(None, 256)	1024
dense_2 (Dense)	(None, 64)	16448
batch_normalization_2 (Batch Normalization)	(None, 64)	256
dense_3 (Dense)	(None, 32)	2080
batch_normalization_3 (Batch Normalization)	(None, 32)	128
batch_normalization_4 (Batch Normalization)	(None, 32)	128
dense_4 (Dense)	(None, 7)	231

Şekil 3.17. Derin Sinir Ağı Eğitim Modeli

Model, üç ağırlık katmanı ve son sınıflandırma katmanı olmak üzere dört katman ile oluşmuştur. Derin otokodlayıcı modelleri verisetinin çok ve karmaşık olmadığı durumlarda genellikle üç katmandan oluşmaktadır. Nöron sayıları ise her çalışma için sabit değildir. Çalışmada en iyi eğitimin olduğu değerler deneme yanılma yoluyla belirlenmektedir [57]. Ağırlık (Dense) katmanı; matris vektör çarpımını temsil eder. Bu katmanda verimli bir hesaplama gerçekleştirebilmek için ReLU fonksiyonu kullanılmıştır. Daha öncesinde anlatıldığı gibi, ReLU fonksiyonunda pozitif değerler korunurken negatif değerler sıfır olarak kabul edilmektedir. Bu durumun avantajı da ağın daha hızlı çalışmasını sağlamıştır. Ağırlık katmanları arasında batch normalizasyon adı verilen toplu normalleşme işlemi yapılmıştır. Toplu normalleştirme yapay sinir ağlarının hızını, performansını ve dengesini arttırmak için kullanılır [33]. Son katmanda ise 7 farklı davranış için eğitim gerçekleştirildiğinden, ikiden fazla sınıflama işlemi gereken durumlarda kullanılan, özellikle derin öğrenme modellerinin çıkış katmanında genellikle tercih edilmekte olan girdinin belirli sınıfa ait olma olasılığını olasılıklar hesaplarla 0-1 aralığındaki değerlere göre hesaplayabilen Entropi (Softmax) katmanı uygulanmıştır.

Modelin hem eğitim hem de doğrulama (validasyon) veri setleri (etiketli test) üzerinde karşılaştırılabilir bir performansa sahip olduğu görülmektedir. Karışıklık matrisinde değerler büyük oranda tutarlı olarak belirlenmiştir.

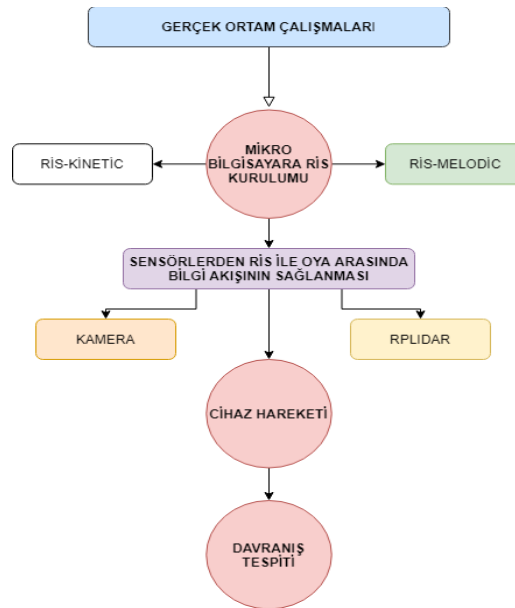
Modelin tamamlanmasının ardından gerçek zamanda çalışan algoritma oluşturulmuştur.



Şekil 3.18. Derin Sinir Ağı yöntemi ile oluşturulan algoritmanın Gazebo ortamında test edilmesi

Algoritma RİS-Gazebo simülasyon ortamında Şekil 3.18.'de gösterildiği gibi test edilmiştir

3.2.4. Gerçek ortamda yapılan çalışmalar



Şekil 3.19. Gerçek ortam çalışmaları akış diyagramı

Şekil 3.19.'da belirtildiği akış şeklinde işlemler gerçekleştirilmiştir. Kurulumlar öncelikle yapılmıştır. Gerçek ortam çalışmaları için Raspberry pi SD kartına ubuntu-mate 18 yüklendiğinden RİS-Melodic kullanıldı. Kurulum işleminin ardından simülasyon ortamında yapılan çalışmalar ile önceden ön hazırlığı yapılan işlemler gerçek sensörler ve RİS sistemi içerisinde çalıştırılmıştır. Laboratuvar ortamında oluşturulmuş OYA'ya RİS aracılığıyla, sensörlerden alınan yayınlar işlenerek gönderilmiş ve cihazın kontrolü bu şekilde sağlanmıştır.

3.2.4.1. Sensör düğümlerinden bilgi akışı sağlanması

Öncelikle kamera ve lazer tarayıcı sensörün RİS sisteminde nasıl açılacağı konusu için çalışılmıştır. RİS içerisinde bir sensörün çalıştırılması için öncelikle o sensöre ait RİS paketinin sistemde bulunması gerekmektedir. Kamera sensörü için eğer usb-kamera kullanılıyorsa farklı bir düğüm, raspberry pi kamerası çalıştırılıyorsa pi düğümünün aktifleştirilmesi gerekmektedir. Sensörler RİS içerisinde çalıştığında aynı zamanda verilerinin de yayınlamaktadırlar. Bu sensör yayınlarının yayınlanması ve alınması işlemleri subscriber- publisher adı verilen yapılar ile sağlanmaktadır.

OYA cihazına sensörleri doğrudan entegre edebilmek için, öncelikle RİS sistemi raspberry içerisinde çalıştırılmıştır. SD karta ubuntu-mate yüklenmiş ve RİS-Melodic kurulumu yapılmıştır. RİS'in diğer bileşenlerinin düzgün çalıştığı saptandıktan sonra sensörler raspberry'e bağlanarak veri akışları gerçekleştirilmiştir.

3.2.4.1.1. Kamera

Çalışma usb kamera kullanılarak yapılmıştır. Bu yüzden RİS için usb- camera düğümü etkinleştirilmiştir.

Usb kameranın sorunsuz kablo bağlantısı oluşturduğunu kontrol için;

⇒ `ls /dev | grep video*`

komutu kullanılmaktadır.

⇒ `video0`

çıktısı alındığında bağlantının normal olduğu saptanır.

usb-camera düğümü için ise;

⇒ sudo apt install ros-melodic-usb-cam

⇒ sudo apt install ros-melodic-web-video-server

komutları aktif edilmelidir. Kamera düğümünün çalışmasının testi ise;

⇒ roslaunch usb_cam usb_cam-test.launch

komutu ile yapılmıştır. Usb kamera düğümü usb_cam/image_raw şeklinde yayın yapmaktadır.

3.2.4.1.2. RPLİDAR 360 lazer tarayıcı

Simülasyon aşamasındaki kurulumları ile aynı kurulumlar, raspberry içerisinde de yapılmıştır. Fakat düğüm çalıştırıldığında yayınladığı veriler çok fazladır. Engelden kaçma algoritması için bize gerekli olan veri sensörün mesafe verisi olduğundan öncelikle bu veri subscribe edilmiştir. Rplidar düğümü içinde Laserscan şeklinde yayın yapılmaktadır.

3.2.4.2. Cihazın hareketi

Kullanılan OYA, ultrasonic sensörler kullanılarak PID kontrolü ile hareket etmektedir. RİS sistemi içerisinde verileri alınan kamera ve RPLIDAR ile de güvenliği daha arttırılmıştır. RPLIDAR hangi yönde ne mesafede engel olduğunu saptamaktadır. Öncelikle alınan data 360 ° ye göre skalalanmıştır. Ardından simülasyon ortamında olduğu gibi açılar ayarlanarak. Engelin önde, sağda ya da solda olmasına göre hareket sağlanmıştır. Raspberry, RİS vasıtasıyla RPLIDAR'dan aldığı bilgileri anlamlı değerlere dönüştürerek STMF4 ile kontrol edilen tekerlere iletmıştır ve güvenli hareket oluşturulmuştur.

3.2.4.3. Gerçek zamanlı davranış tespit etme

Davranış tespit etme algoritması simülasyon ortamı ile tamamen aynıdır. Farklılık olarak donanımın çok güçlü olmamasından kaynaklanan yavaşlama problemleri

yaşanmıştır. Bunun önüne geçmek için raspberry pi kamerası ve Jetson nano ile denemeler yapılmış daha iyi sonuçlar elde edilmiştir.

Tablo 3.2. Jetson nano ile raspberry pi B + fps değerleri

	JETSON NANO	RASPBERRY Pİ 3 B+	ASUS LAPTOP	CPU
FPS	1- 1.5 fps	0.01 – 0.5 fps	0.4- 0.7 fps	

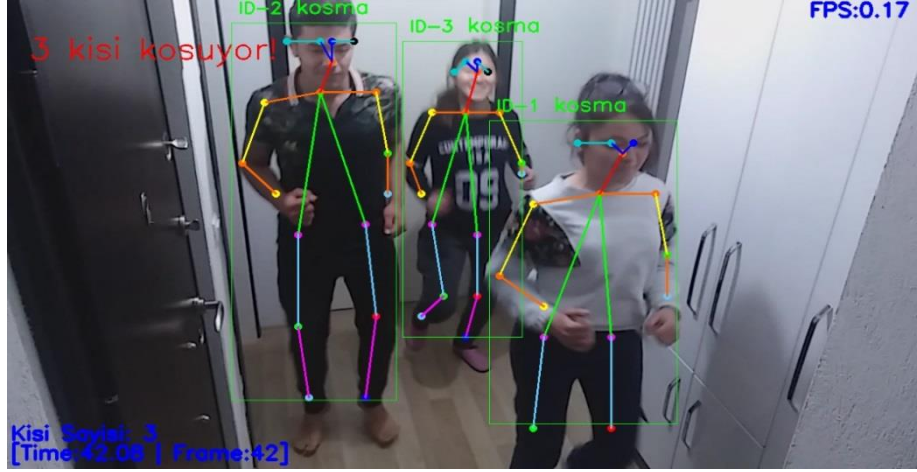
Donanımsal çalışma hızı, deneysel olarak çalışmaların test edilebilir yapıda olması açısından yeterli olsa da Tablo 3.2.'de de görüldüğü gibi gerçek bir ortam için yeterli değildir. Örneğin; 4352 çekirdekli RTX 2080Ti donanımına sahip bir endüstriyel bilgisayar, python yazılım koduyla yazılmış gerçek zamanlı bir çalışma için yaklaşık 111 fps civarında başarılı sonuçlar verebilmektedir. İşlem gücü ve GPU performansı yeterli seviyede olan herhangi bir endüstriyel bilgisayar ile tez çalışması algoritmaları sorunsuz şekilde çalışarak gerçek bir hastane ortamında kullanılabilir durumdadır.

3.2.5. Merkeze anormal durumların bilgisinin gönderilmesi

Araştırmada, belirlenen anormal durumlar;

- Çok kişinin birden koşması,
- Bir kişinin düşmesi
- Çok kişinin birden eylem hareketi yapması
- kavga eden kişiler gibi durumlar için güvenlik birimlerine çıktı üretilmesi

amaçlanmıştır. Bu üretilen çıktılar micro bilgisayarlara uzakdan VNC ile bağlı halde olduğunda görüntülenerek saptanabilmektedir. Ayrıca bu çalışma farklı ortamlarda kullanılacak olursa, çeşitli sinyallerle de bilgi akışı sağlayarak haber ulaştırılabilmektedir.



Şekil 3.20. Çok kişinin birden koşması durumunun gösterimi

Şekil 3.20.'de 3 kişinin birden koşmasının gerçek ortamda tespiti gösterilmektedir. Hastane ortamında genellikle insanlar toplu halde koştuklarında bir hastayı acile yetiştirme vb. gibi önemli acil durumları çağrıştırdığından bu durum anormaldir ve acil servis ekibinin her ihtimale karşı haberdar edilmesi gereken bir durumdur.



Şekil 3.21. Birinin düşmesi durumunun simülasyonda gösterimi

Hastane ortamında düşen herhangi birinin sebebi ise kalp krizi, beyin kanaması, bayılma vb. gibi durumlarda ortaya çıkabileceğinden düşen bir kişi bile anormal durum sınıfına dahil edilmiştir. Şekil 3.21.'de simülasyon ortamında OYA'nın düşen birini tespit etmesi sonucu kendini beklemeye alması gözlenmektedir.



Şekil 3.22. Kavga ortamının gerçek ortamda gösterilmesi

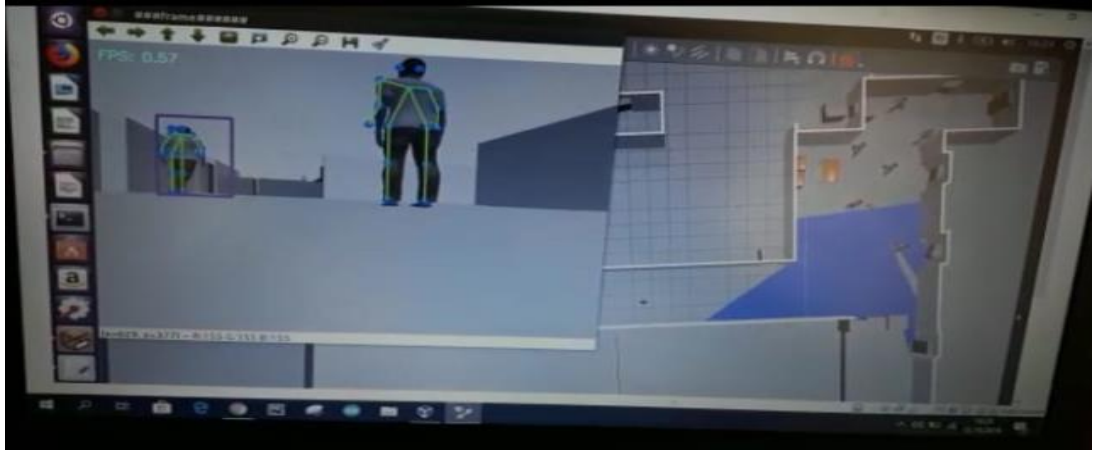
Kavga eden ve eylem yapan kişiler, her toplu bulunulan ortamlar için tedirginlik oluşturan durumlardır. Bu yüzden bunlarda anormal davranış sınıfına dahillerdir. Şekil 3.22.'de gerçek ortamda kavga eden kişilerin tespit edilmesi gösterilmektedir.

BÖLÜM 4. ARAŞTIRMA BULGULARI

4.1. Davranış Tespiti Başarı Oranları

4.1.1. Matematiksel yöntem

Matematiksel yöntemde, bağlantı noktalarının konumuna göre eylemler bazı kordinatlar ve mesafeler için oluşturulmuştur. Eylemler şablonlarla eşleştirilirken bunlar kod içerisinde manuel olarak gerçekleştirildiklerinden matematiksel yöntem benzer hareketleri daha çok birbirine karıştırmaktadır ve her hareketi tanımlayamamaktadır. Şekil 4.1.'de tanımlayamadığı bir an için örnek gösterilmektedir. Koşma- yürüme, kavga – eylem hareketleri kordinat olarak benzer yapıda olduklarından seçiminde güçlük oluşmaktadır.

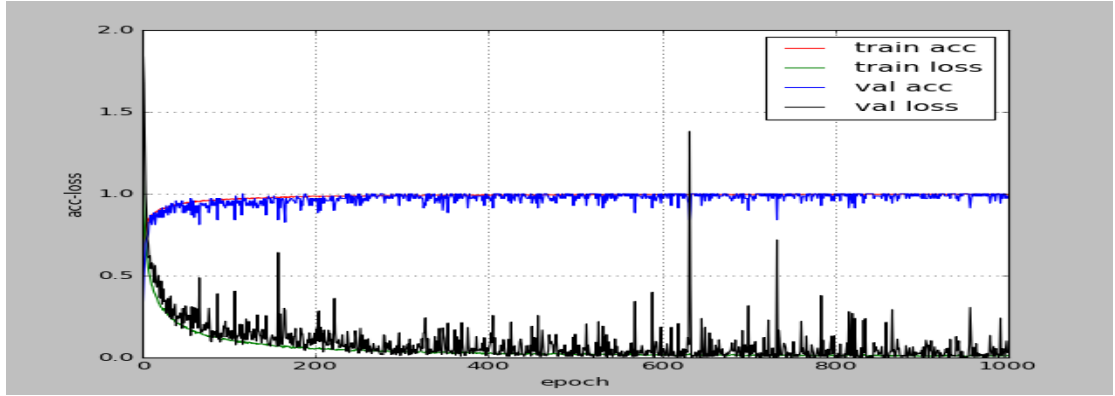


Şekil 4.1. Simülasyon ortamında matematiksel işlemin testi

4.1.2. Yapay zeka yöntemi

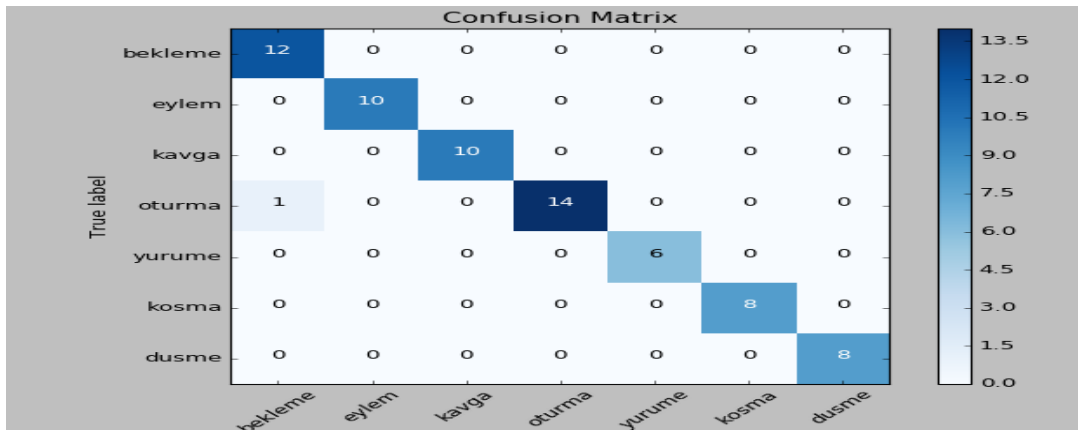
Derin oto kodlayıcılar ile yıldız iskelet model verileri kullanılarak oluşturulmuş modelin eğitimi sonucunda elde edilen başarı oranı %98 oranında

gerçekleştirilmiştir. Eğitim birçok kez tekrar edilmiş elde edilen en iyi sonuç kullanılmıştır. Eğitim aşaması sonrasında oluşan grafikler Şekil 4.2. ve 4.3.'de gösterilmiştir. Modelin hem eğitim hem de doğrulama (validasyon) veri setleri (etiketli test) üzerinde karşılaştırılabilir bir performansa sahip olduğu görülmektedir. Karışıklık matrisinde değerler büyük oranda tutarlı olarak belirlenmiştir.



Şekil 4.2. Eğitim modeli doğruluk ve kayıp grafikleri

Kayıp grafiği 0 değerine yakın ve validasyon kaybı ile paralel yapıdadır. Doğruluk grafiği de aynı şekilde validasyon grafiği ile paralel ve 1 değerine ulaşmıştır. Bütün bunlara bakıldığında ise doğruluk ve kayıp grafikleri başarılı bir eğitimin gerçekleştirildiği sonucunu göstermektedir.



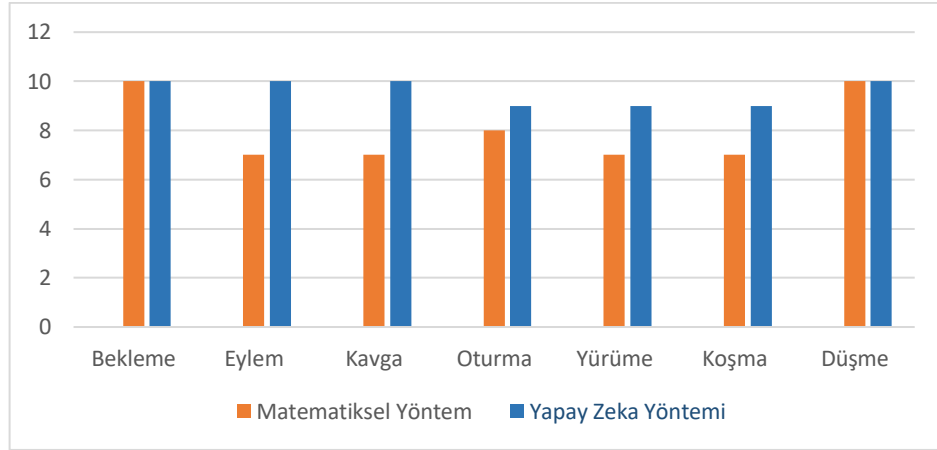
Şekil 4.3. Karışıklık Matrisi (Confusion Matrix)

Karışıklık matrisinde ise bu eğitim sonucunda oluşturulan eylemlerden bekleme, eylem, kavga, yürüme, koşma, düşme için verilen gerçek değerleri gerçek sonuçlarla

şşırmadan verirken sadece oturma verisi için verilen 15 verinin sadece 1 tanesini bekleme verisi ile karıştırdığı görülmektedir.

Davranış tespit işleminde matematiksel ve yapay zeka yöntemleri için ayrı ayrı test işlemi yapılmıştır. Yapılan çalışmalarda matematiksel yöntem başarısı yapay zeka yöntemlerine göre daha zayıf kalmıştır.

Tablo 4.1. Matematiksel yöntem ve yapay zeka yönteminin davranışları tespit etme oranları grafiği

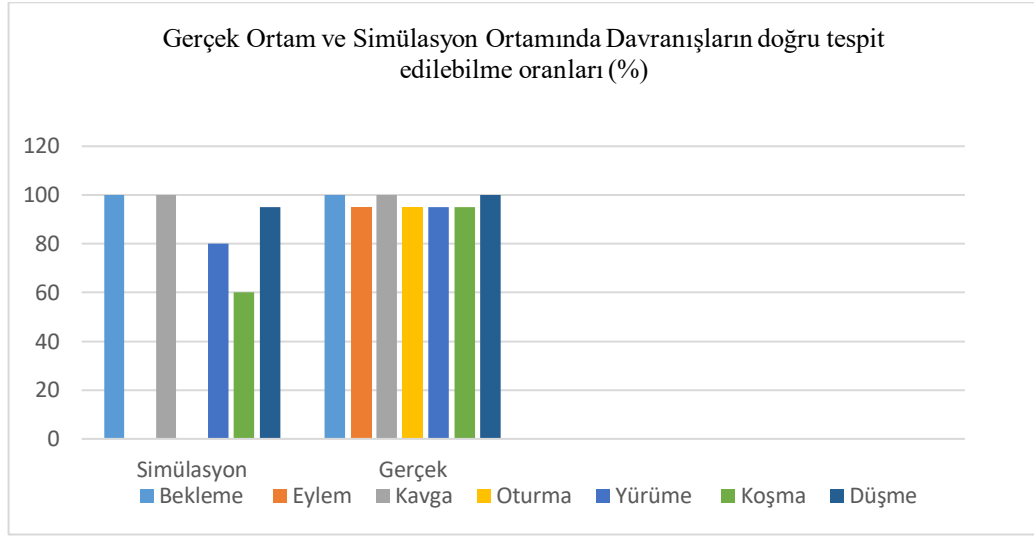


Tablo 4.1.'de matematiksel yöntem ve yapay zeka yönteminin davranış tespitlerine göre kıyaslaması gösterilmektedir. Tablodan da anlaşılacağı üzere davranışların tespit edilmesi konusunda matematiksel yönteme kıyasla yapay zeka yöntemi oldukça güçlü görünmektedir.

4.2. Simülasyon Ortamı ve Gerçek Ortam Kıyaslaması

Tablo 4.2.'de davranışların gerçek ortam ve simülasyon ortamında yüzdesel olarak doğru tespit edilme oranları gösterilmektedir.

Tablo 4.2. Gerçek ortam ve Simülasyon ortamının davranış tespiti için kıyaslanması



Bekleme ve kavga davranışları simülasyon ortamında hatasız şekilde tespit yapmakta iken eylem ve oturma davranışlarının tespiti mümkün olmamaktadır. Yürüme %80, koşma %60 ve düşme %90 oranlarında doğruluk göstermektedir. Gerçek ortamda yapılan deneylerde ise; bekleme, kavga, düşme eylemleri hatasız bir şekilde tespit edilebilmektedir. Diğer davranışlar ise %90 doğruluk ile tespit edilmektedir.

Araştırmada, simülasyon ortamı aracın hareketi için yazılan kodların testi için daha kullanışlı ve gerçeğe yakın sonuçlar vermektedir. Bu açıdan RPLIDAR verileri için simülasyon ortamı için kullanılan değerler, algoritmalar gerçek ortam ile oldukça benzerdir. Fakat davranış tespiti çalışmasında kullanılan insan figürleri hareketli yapıda olmadığından her hareketin rahatlıkla test edilmesi sağlanamamaktadır. Bu yüzden elde edilen sonuçlar gerçek ortamdakine kıyasla daha kısıtlı kalmaktadır.

BÖLÜM 5. TARTIŞMA VE SONUÇ

Endüstri faaliyetlerindeki gelişmelerle birlikte kullanımı yaygınlaşan otomatik yönlendirmeli araçlar (OYA); hastane içi lojistik akışların otomasyonun sağlanması, kaynakların en hızlı ve etkin şekilde kullanılmasına katkı sağlaması, hasta bakımının verimliliğini ve kalitesini arttırması gibi nedenlere bağlı olarak hastane ortamlarında da kullanılmaktadırlar. Taşınan malzemelerin bütünlüğünü korumak, yüksek seviyede hijyen sağlamak ve hastaların sağlığını garanti altına almak için sağlık sektöründe introlojistik ulaşım kalitesi diğer sektörlerden çok daha önemlidir. Acil bekleme koridorları birçok hasta ve hasta yakınının bir arada bulunduğu ortamlardır. Ön müdahale işleminin ardından hastaların durumlarına göre çok acil olandan daha az acil olana doğru; kırmızı, sarı veya yeşil alanlara yönlendirilmektedirler. Ön müdahale biriminde yanlış teşhis konulabilmesi ve hasta kayıt sistemlerinin yavaş ilerlemesi gibi sebeplerle kırmızı alanda olması gereken bir hasta yeşil alanda çok uzun süre sıra beklemek durumunda kalabilmektedir. Yapılan bazı araştırmalara göre yatışı planlanan hastalar için acilde bekleme süresi normal şartlarda olması gerekenin çok üzerinde olduğu gözlenmiştir [66]. Bu sebeplerle bekleme salonlarında bayılma, panik hali gibi durumlar olabilmektedir. Hasta yakınlarında üzüntü ve strese bağlı olarak gerginlik, başka kişiler ya da sağlık personeli ile bir tartışma ortamı oluşabilmektedir. Bu gibi problemler göz önünde bulundurularak hastane ortamı gibi toplu halde insanların bulunduğu ortamlarda görsel gözetimli güvenlik sistemleri bulunmaktadır. Görsel gözetleme sistemleri; çok miktarda bilgiyi depolayabilir, eğer akıllı sistemler mevcut ise bu bilgileri ayıklayabilmekte, içerisinden çeşitli çıkarımlar da yapabilmektedir [67]. Bu tez çalışmasında da görsel gözetleme sistemleri bir OYA'nın içerisinde birleştirilerek hibrit bir işlem gerçekleştirilmiştir. OYA'lar hastane ortamında malzeme taşıma işleminin yanında bir gözetleyici görevinde kullanılmıştır. Ulaştığı yerleri görüntüleyerek insan iskelet sistemi verilerine göre davranış analizi yapmaktadır. Bilgiyi sadece görüntü olarak değil davranışlar olarak

toplamaktadır. Görsel gözetleme sistemleri çoğunlukla videoyu depolamak, dağıtmak gibi işlemler için kullanılmaktadırlar. Görselden elde edilen problemi tespit etmek ya da ulaşılmak istenen veriyi elde etmek gibi asıl bilgiyi oluşturan şeyleri ise insan operatörlerin gözetimine bırakmaktadır. Videoların manuel olarak analiz edilmesi dikkatli ve titiz bir çalışma gerektirmektedir. Bu durum otonom olarak sonuç üreten bir sisteme göre daha maliyetli ve yavaş sonuç üretilmesine sebep olmaktadır. Ayrıca bu sistemlerde bulunan görsel gözetleme araçları da sabit tek bir bölgeyi izleme olanağı sağlamaktadır. Bütün bunlar göz önüne alındığında, hali hazırda bulunan gözetleme sistemleri ile durum bilgisinin hızlı ve kolay şekilde sağlanması daha zorlukla olmaktadır. Tez çalışmasında, otomatik yönlendirmeli araçlarda görsel gözetim sistemi ile elde edilen görüntülerdeki kişilerin davranışlarının tespit edilmesi gerçek zamanlı olarak gerçekleştirilmiştir. Robot işletim sistemi (RİS) üzerinde test edilerek deneysel OYA ile gerçek ortamda çalıştırılmıştır. RİS, içerisinde birçok düğümü eş zamanlı çalıştırarak işlem yapabilme, düğümlerde oluşan bir problemde diğerine zarar vermeden çalıştırma, farklı yazılım dilindeki kodlar için kullanılabilme gibi OYA'lar için daha birçok avantajı içerisinde bulundurmaktadır. Çalışmada RPLIDAR'dan alınan verilere göre karşısına çıkan engellerden korunarak yol alırken farklı bir düğüm ile görüntü işleme çalışması gerçekleştirilmiştir. RİS sayesinde düğümler arasında geçiş sağlanabildiğinden belirlenen bir kargaşa durumunda, ortamı kayıt altında tutması için OYA durdurulabilmektedir.

Temel davranışların tespit edilmesi için openpose algoritmasından elde edilen eklem koordinat verilerinin kullanıldığı yöntemler ile davranış tespiti derin-yığınli otokodlayıcılar ile gerçekleştirilmiştir. Derin otokodlayıcılar, girişteki veriyi çıkış verisine doğrusal olmadan eşleştirerek verisetinin boyutunu düşürmektedir. Bu özelliği ile Temel bileşenler analizine (Principal component analysis (PCA)) benzerdir. Temel bileşen analizi, verileri başka bir alana yansıtarak doğrusal bir dönüşüm ile veri kümesini daha az sayıya sınırlayarak boyutun küçülmesini sağlamaktadır. Çalışmada bulunan veriler birbirine doğrusal olmayan ve temel vücut kilit noktalarını belirttiğinden yığınli otokodlayıcılar verinin çok daha hızlı bir şekilde verimli olarak eğitilmesini sağlamaktadır.

Sonuç olarak, tez çalışması hastane ortamında malzeme taşıma işlemi yapan OYA'lar için eş zamanlı yeni bir görev tanımlama işlemi gerçekleştirmiştir. Görsel gözetleme işlemi ve RİS sistemi içerisinde kullanarak hibrit bir OYA cihazı geliştirilmiştir. Çalışma OYA cihazı için genel anlamda yapılan yol planlama işlemi yanında, görsel gözetleme işlemleri için çalışan operatörlerin işyükünün azalmasına da katkı sağlamaktadır.

KAYNAKLAR

- [1] Ullrich, G., Automated guided vehicle systems, Springer-Verlag, Berlin Heidelberg, 978–981, 2015.
- [2] Schulze, L., Wullner, A., The Approach of Automated Guided Vehicle Systems, 2006 IEEE International Conference on Service Operations and Logistics, and Informatics, Shanghai, China, 522-527, 2006.
- [3] Amaya CA, Beaulieu M., Landry S., Rebolledo C., Velasco N., Potenciando la logística hospitalaria: tres casos, tres trayectorias, 14 (4): 85-98., 2010.
- [4] Aydın, A., Otomatik yönlendirmeli araçlarda yörünge kontrolü, Fırat Üniversitesi, Elektronik ve bilgisayar eğitimi anabilim dalı, Yüksek lisans tezi, 2012.
- [5] Bulut, E., Endüstri 4.0 ve inovasyon göstergeleri kapsamında Türkiye analizi, (ASSAM- UHAD) ASSAM International Refereed Journal, Baskı: 7, 2017.
- [6] Llopis-Albert, C., Rubio, F., Valero, F., Fuzzy-set qualitative comparative analysis applied to the design of a network flow of automated guided vehicles for improving business productivity, Journal of Business Research, Elsevier, vol. 101(C), 737-742, 2019.
- [7] Hwang, H., Kim, S.H., Development of dispatching rules for automated guided vehicle systems. J. Manuf. Syst. 137–143, 1995.
- [8] Menyhárt, J., Szabolcsi, R., Artificial intelligence applied for technical status diagnostics of the batteries of automated guided vehicles, 2019 International Conference on Military Technologies (ICMT), Çek Cumhuriyeti, 1-8, 2019.
- [9] Ma, X., Bian, Y., Gao, F., An improved shuffled frog leaping algorithm for multiload AGV dispatching in automated container terminals, College of Mechanical Engineering, Tongji Üniversitesi, Çin, 2020.
- [10] Yılmaz, Z., Bayındır, L., Simulation of Lidar-Based Robot Detection Task using ROS and Gazebo, European Journal of Science and Technology Special Issue,(17), 513-529, 2019.

- [11] Schulze, L., Behling, S., Buhrs, S., People following automated guided vehicles-research and application, Proceedings of the International MultiConference of Engineers and Computer Scientists, Hong Kong, 18- 20, 2009.
- [12] Zhang, J., Peng, L., Ju, Z., Liu, H., Feng, W., Human-AGV Interaction: Real-Time Gesture Detection Using Deep Learning, International Conference on Intelligent Robotics and Applications, Çin, 231-242, 2019.
- [13] Yan, J., Yan, S., Zhao, L., Wang, Z., Liang, Y., Research on Human-Machine Task Collaboration Based on Action Recognition, 2019 IEEE International Conference on Smart Manufacturing, Industrial & Logistics Engineering (SMILE), Hangzhou, China, 117-121, 2019.
- [14] Mulcahy, D. E., Materials Handling Handbook, McGraw-Hill, New York, 1999.
- [15] Korçak, E. B., Otomatik Malzeme Taşıma Sistemi Tasarımı İçin Hibrit Yaklaşım, Gazi Üniversitesi, Fen Bilimleri Enstitüsü, Endüstri Mühendisliği Anabilim Dalı, Yüksek Lisans Tezi, 2010.
- [16] Mercan, E., Otomatik Yönlendirmeli Araç Tasarımı ve Gerçek Zamanlı Kontrolü, Atatürk Üniversitesi, Elektrik ve Elektronik Mühendisliği Anabilim Dalı, Yüksek Lisans Tezi, 2010.
- [17] Forger, G., Non-wire Guidance Systems Take Control of AGVs, Modern Materials Handling, 1998.
- [18] Digani, V., Secchi, C., Cotena, G., Sabattini, L., Technological Roadmap to Boost the Introduction of AGVs in Industrial Applications, Department of Engineering Sciences and Methods (DISMI), University of Modena and Reggio Emilia via G. Amendola 2, 42122 Reggio Emilia – İtalya, 2013.
- [19] Quigley, M., Gerkey, B. and Smart, W.D., Programming Robots with ROS: A Practical Introduction to the Robot Operating System, O'Reilly Media, United States of America, 1. Edition, 2016.
- [20] Çelikbaş, Ş., Kamanlı, A.F., Özdemir, A.E., Robot İşletim Sisteminde İnsan Davranışının Poz Tahmini Algoritmalarıyla Tespit Edilmesi, Spec International Conference On Engineering & Natural Sciences-4, Ankara, 138-144, 2019.
- [21] Martinez, A., Fernandez, E., Learning ROS for Robotics Programming: A practical, instructive, and comprehensive guide to introduce yourself to ROS, the top-notch, leading robotics framework, Packt Publishing, Mumbai, 1. Edition, 2013.

- [22] Kövecses, J., Piedboeuf, J., Lange, C., Dynamics Modeling and Simulation of Constrained Robotic Systems, IEEE/ASME, Kingston, Canada, 2003.
- [23] Takaya, K., Asaiy, T., Kroumovz, V., Smarandachex, F., Simulation Environment for Mobile Robots Testing Using ROS and Gazebo, 20th International Conference on System Theory, Control and Computing (ICSTCC), Romanya, 2016.
- [24] Ervan, O., Improved Performance For Slam Techniques Using Trap Configured 2d Lrfs, İstanbul Teknik Üniversitesi, Teknoloji Fakültesi, Kontrol ve Otomasyon Mühendisliği, Yüksek Lisans Tezi, 2016.
- [25] http://docs.ros.org/indigo/api/rviz/html/user_guide/, Erişim Tarihi: 08.01.2020.
- [26] Krishnamurthy, N.N., Batra, R. and Karwan, M.H., Developing conflict-free routes for automated guided vehicles, Operations Research, Vol. 41 No. 6, 1077-90,1993.
- [27] Bozer, Y.A., Srinivasan, M.M., Tandem configurations for automated guided vehicle systems and the analysis of single vehicle loops, IIE Transactions, Vol. 23 No. 1, 72-82, 1991.
- [28] McCULLOCH, W.S., PITTS, W.A., A Logical Calculus of the Ideas Immanent in Nervous Activity, Bulletin of Mathematics and Biophysics, 115-133, 1943.
- [29] Saridoğan, M., Öz Örgütlemeli (Kohonen) Haritalar İle Esnek Üretimde Yer Alan Otomatik Güdümlü Araçların (OGA) Rotalarının Belirlenmesi, İstanbul Üniversitesi, Fen Bilimleri Enstitüsü, Endüstri Mühendisliği Anabilim Dalı, Endüstri Mühendisliği, Yüksek Lisans Tezi, 2009.
- [30] Sivanandam, S. N., Deepa, S. N., Introduction to Neural Networks Using Matlab 6.0, Tata McGraw-Hill Education, 2006.
- [31] Karlik, B., Olgac, A.V., Performance Analysis of Various Activation Functions in Generalized MLP Architectures of Neural Networks, 4. Baskı, Journal of Artificial Intelligence And Expert Systems (IJAE), İsviçre, 111-122, 2010.
- [32] Goodfellow, I., Bengio, Y., Courville, A., Deep Learning, MIT Press Yayın Evi, 175-195, 2017.
- [33] Glorot, X., Bordes, A., Bengio, Y., Deep Sparse Rectifier Neural Networks, Appearing in Proceedings of the 14th International Conference on Artificial Intelligence and Statistics (AISTATS), Fort Lauderdale, FL, USA., 315-323, 2011.

- [34] Maas, A.L., Hannun, A.Y., Ng, A.Y., Rectifier Nonlinearities Improve Neural Network Acoustic, Proceedings of the 30 th International Conference on Machine Learning, Atlanta, Georgia, USA, 2013.
- [35] He, K., Zhang, X., Ren S., Sun, J., Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification, Microsoft Research, 2015.
- [36] Athiyaman, A., Robertson, R.W., Time Series Forecasting Techniques: Short-Term Planning in Tourism, International Journal of Contemporary Hospitality Management, 4 (4): 8-11, 1992.
- [37] Ataseven, B., Yapay Sinir Ağları İle Öngörü Modellemesi, Öneri Dergisi, 10(39): 101-115, 2013.
- [38] Chapelle, O., Scholkopf, B., & Zien, A., Semi-supervised learning, IEEE Transactions on Neural Networks, 20(3), 542-542,2009.
- [39] Bengio, Y., Courville, A., Vincent, P., Representation Learning: A Review and New Perspectives, IEEE Trans. Pattern Anal. Mach. Intell., 35. Baskı, 1798–1828, 2013.
- [40] Song, H. A., Lee, S.-Y., Hierarchical Representation Using NMF, In International Conference on Neural Information Processing., 466–473,2013.
- [41] <https://devblogs.nvidia.com/parallelforall/deep-learning-nutshell-history-training/> ., Erişim Tarihi:15.02.2020.
- [42] Şeker, A., Dirib, B., Balık, H.H., Derin Öğrenme Yöntemleri ve Uygulamaları Hakkında Bir İnceleme, Gazi Mühendislik Bilimleri Dergisi, 3(3): 47-64, 2017.
- [43] Aizenberg, I. N., Aizenberg, N.N., Vandewalle, J., Multiple-Valued Threshold Logic and Multi-Valued Neurons, Multi-Valued and Universal Binary Neurons, Boston, MA: Springer US, 25–80, 2000.
- [44] Hinton, G. E., Learning multiple layers of representation, Trends Cogn. Sci., 11. Baskı, 428–434, 2007.
- [45] Girshick, R., Donahue, J., Darrell, T., Malik, J., Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation, IEEE Conference on Computer Vision and Pattern Recognition, 580–587, 2014.
- [46] Wang, Y., A new concept using LSTM Neural Networks for dynamic system identification, American Control Conference (ACC), 2017.

- [47] Doetsch, P., Kozielski, M., Ney, H., Fast and Robust Training of Recurrent Neural Networks for Offline Handwriting Recognition, 14th International Conference on Frontiers in Handwriting Recognition, Heraklion, 2014.
- [48] Graves, A., Mohamed, A.-r., Hinton, G., Speech recognition with deep recurrent neural networks, IEEE International Conference on Acoustics, Speech and Signal Processing, Vancouver, Canada, 2013.
- [49] Coates, A., Ng, A., Lee, H., An Analysis of Single-Layer Networks in Unsupervised Feature Learning, PMLR, 215–223, 2011.
- [50] Larochelle, H., Bengio, Y., Classification using discriminative restricted Boltzmann machines, Proceedings of the 25th international conference on Machine learning- ICML '08, 536–543, 2008.
- [51] Hinton, G. E., Salakhutdinov, R. R., Reducing the Dimensionality of Data with Neural Networks, 313. Baskı, 504–507, 2006.
- [52] Hinton, G. E., Salakhutdinov, R. R., Replicated Softmax: an Undirected Topic Model, Advances in Neural Information Processing Systems 22, 1607–1614, 2009.
- [53] Salakhutdinov, R., Mnih, A., Hinton, G., Restricted Boltzmann machines for collaborative filtering, Proceedings of the 24th international conference on Machine learning- ICML '07, 791–798, 2007.
- [54] G. E., Hinton, A practical guide to training restricted Boltzmann machines, In Neural networks: Tricks of the trade, Berlin, Heidelberg, 599-619,2012.
- [55] Taylor, G. W., Hinton, G. E., Roweis, S. T., Modeling Human Motion Using Binary Latent Variables, In Advances in Neural Information Processing Systems 19, 1345–1352, 2006.
- [56] Seker, A., Gürkan, A., Stacked Autoencoder Method for Fabric Defect Detection, Cumhuriyet Üniversitesi Fen Fakültesi Fen Bilimleri Dergisi (CFD), 38. Baskı, 342-354, 2017.
- [57] Kaynar, O., Görmez, Y., Işık, Y.E., Oto kodlayıcı tabanlı derin öğrenme makinaları ile spam tespiti, 3. uluslararası yönetim bilişim sistemleri konferansı, 44-54, 2016
- [58] Deng, L., Seltzer, M., Yu, D., Acero, A., Mohamed, A., Hinton, G., Binary Coding of Speech Spectrograms Using a Deep Auto-encoder, In Interspeech, 1692–1695, 2010.
- [59] Lyudchik, O., Outlier detection using autoencoders, CERN Document Server, 2016.

- [60] Li, M., Leung, H., Graph-based approach for 3D human skeletal action Recognition, 87. Baskı, Elsevier, 195–202, 2017.
- [61] Barbič, J., Safonova, A., Pan, J.-Y., Faloutsos, C., Hodgins, J.K., Pollard, N.S., Segmenting motion capture data into distinct behaviors, Canadian Human-Computer Communications Society, 185–194, 2004.
- [62] Haritaoglu, I., Harwood, D., Davis, L. S., Real-Time Surveillance Of People And Their Activities, IEEE Trans. on PAMI, 22. Baskı, 809 – 830, 2000.
- [63] Adar, N., Kandemir, C., Video Dizilerinde Hareket Tanıma. Eskişehir Osmangazi Üniversitesi Mühendislik ve Mimarlık Fakültesi Dergisi, 24(2): 116-101, 2011.
- [64] <http://wiki.ros.org/kinetic/Installation/Ubuntu>, Erişim Tarihi: 6.02.2020
- [65] Bailey, T., Durrant-Whyte, H., Eşzamanlı lokalizasyon ve haritalama (SLAM): bölüm II. Simultaneous Localization and Mapping (SLAM): Part II, IEEE Robotics & Automation Magazine, 13 (3): sayfa 108-117, 2006.
- [66] Aydın, T., Aydın A.Ş., Köksal Ö., Özdemir F., Kulaç S., Bulut M., Uludağ Üniversitesi Tıp Fakültesi Hastanesi Acil Servisine Başvuran Hastaların Özelliklerinin ve Acil Servis Çalışmalarının Değerlendirilmesi, Akademik Acil Tıp Dergisi, 163-168, 2010.
- [67] Kekeç, T., Developing Object Detection, Tracking and Image Mosaicing Algorithms for Visual Surveillance, Sabancı Üniversitesi, Yüksek Lisans Tezi, 2013.

ÖZGEÇMİŞ

Şerife Çelikbaş, 06.09.1994'de Denizli'de doğdu. İlk, orta ve lise eğitimini Denizli'de tamamladı. 2012 yılında Denizli Lisesi Anadolu Statüsü'nden mezun oldu. 2012 yılında başladığı Namık Kemal Üniversitesi Biyomedikal Mühendisliği Bölümü'nü 2017 yılında bitirdi. 2017 yılında Sakarya Üniversitesi Biyomedikal Mühendisliği Bölümü'nde yüksek lisans eğitimine başladı. Halen Sakarya Üniversitesi Biyomedikal Mühendisliği Bölümü'nde yüksek lisans eğitimini sürdürmektedir.