T.C.
YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

ÇOK BÖLGELİ UÇUŞ SÜRESİ SENSÖRLERİ KULLANILAN OTONOM CİHAZLAR İÇİN TINYML PLATFORMU İLE NESNE ALGILAMA

Sergen Erkan ARSLAN

YÜKSEK LİSANS TEZİ
Elektronik ve Haberleşme Mühendisliği Anabilim Dalı
Haberleşme Programı

Danışman
Doç. Dr. Bülent BOLAT

Mayıs, 2024

T.C.
YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

ÇOK BÖLGELİ UÇUŞ SÜRESİ SENSÖRLERİ KULLANILAN OTONOM CİHAZLAR İÇİN TINYML PLATFORMU İLE NESNE ALGILAMA

Sergen Erkan ARSLAN tarafından hazırlanan tez çalışması 14.05.2024 tarihinde aşağıdaki jüri tarafından Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü Elektronik ve Haberleşme Mühendisliği Anabilim Dalı Haberleşme Programı **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Doç. Dr. Bülent BOLAT
Yıldız Teknik Üniversitesi
Danışman

Jüri Üyeleri

Doç. Dr. Bülent BOLAT, Danışman
Yıldız Teknik Üniversitesi _____

Dr. Öğr. Üyesi Murat TAŞKIRAN, Üye
Yıldız Teknik Üniversitesi _____

Dr. Öğr. Üyesi Necdet Tuğrul ARTUĞ, Üye
İstanbul Üniversitesi Cerrahpaşa _____

Danışmanım Doç. Dr. Bülent BOLAT sorumluluğunda tarafımca hazırlanan "Çok Bölgeli Uçuş Süresi Sensörleri Kullanılan Otonom Cihazlar için TinyML Platformu ile Nesne Algılama " başlıklı çalışmada veri toplama ve veri kullanımında gerekli yasal izinleri aldığımı, diğer kaynaklardan aldığım bilgileri ana metin ve referanslarda eksiksiz gösterdiğimi, araştırma verilerine ve sonuçlarına ilişkin çarpıtma ve/veya sahtecilik yapmadığımı, çalışmam süresince bilimsel araştırma ve etik ilkelerine uygun davrandığımı beyan ederim. Beyanımın aksinin ispatı halinde her türlü yasal sonucu kabul ederim.

Sergen Erkan ARSLAN

İmza

Bu tez öncelikle, eğitimin önemini her fırsatta anlatmaya çalışan dedem
Ferittin ARSLAN 'a, desteklerini esirgemeyen
kıymetli eşim ve aileme ithaf edilmiştir.

TEŞEKKÜR

Yüksek lisans eğitimim süresince bilgi ve tecrübelerini aktaran tüm hocalarıma, birlikte çalıştığım ARÇELİK A.Ş. Merkez AR-GE Sensör Yöneticiliği çalışanlarına ve son olarak bu günlere gelmemde en büyük pay sahibi olan, desteklerini her daim eksik etmeyen aileme ve kıymetli eşime sonsuz teşekkürlerimi sunuyorum.

Sergen Erkan ARSLAN

İÇİNDEKİLER

SİMGE LİSTESİ vii

KISALTMA LİSTESİ viii

ŞEKİL LİSTESİ x

TABLO LİSTESİ xii

ÖZET xiii

ABSTRACT xv

1 GİRİŞ 1

1.1 Tezin Amacı 1

1.2 Literatür Araştırması 1

1.3 Hipotez 3

2 YÖNTEM 4

2.1 TINY ML 4

2.1.1 Tiny ML Avantajları ve Zorlukları 5

2.1.2 Uygulama Alanları 7

2.1.3 Tiny-ML Geliştirmeye Yönelik Araçlar ve Teknikler 8

2.2 UÇUŞ SÜRESİ SENSÖRÜ (ToF) 19

2.2.1 Çalışma Prensibi 19

2.2.2 Kullanım Alanları 23

2.2.3 ToF Entegre Sensör Üreticileri 24

2.3 VL53L7CX ToF Sensör ve Veri Seti 28

3 UYGULAMA 32

3.1 Kullanılan Donanım 32

3.2 Kullanılan Yazılımlar 33

3.3 Veri Setinin Toplanması ve Etiketlenmesi 35

3.4 Kullanılan Tiny-ML Platformu ve Önerilen CNN Modeli 36

4 UYGULAMA SONUÇLARI **41**
4.1 Sinyal Seviyesi Verisi Model Sonucu 42
4.2 Yansıma Verisi Model Sonucu 43
4.3 Mesafe Değişimi Verisi Model Sonucu 43
4.4 Mesafe Verisi Model Sonucu 44
4.5 Tüm Sensör Verilerinin Özet Sonuçları 45

5 SONUÇ **47**

KAYNAKÇA **48**

TEZDEN ÜRETİLMİŞ YAYINLAR **53**

SİMGE LİSTESİ

cm	CentiMetre
GHz	GigaHertz
Hz	Hertz
KB	KiloByte
Kcsp	Kilocounts per second
KHz	KiloHertz
MB	MegaByte
MHz	MegaHertz
μsn	MicroSecond
mA	MiliAmper
ms	MiliSecond
mm	MiliMetre
mW	MiliWatt
nm	NanoMetre
ps	PicoSecond
V	Volt

KISALTMA LİSTESİ

2D	Two Dimensional
3D	Three Dimensional
ADAS	Advanced Driver Assistance System
AI	Artificial Intelligence
AIfES	Artificial Intelligence for Embedded Systems
AMCW	Amplitude Modulation Continuous Wave
ARM	Advanced RISC Machine
BLE	Bluetooth Low Energy
CMSIS	Common Microcontroller Software Interface Standard
CMSIS-NN	Common Microcontroller Software Interface Standard Neural Network
CNN	Convolutional Neural Network
CSV	Comma Separated Values
CW	Continuous Wave
dToF	Direct Time of Flight
Edge-AI	Edge Artificial Intelligence
ELL	Embedded Learning Library
EWARM	Embedded Workbench for Arm
FoV	Field of View
I2C	Inter-Integrated Circuit
IMU	Inertial Measurement Unit
IoT	Internet of Things
IO	Input-Output
IR	Infra Red

iToF	Indirect Time of Flight
ML	Machine Learning
NN	Neural Network
NNoM	Neural Network on Microcontroller
ONNX	Open Neural Network Exchange
PCB	Printed Circuit Board
ReLU	Rectified Linear Unit
RAM	Random Access Memory
RGB	Red Green Blue
RISC	Reduced Instruction Set Computer
RISC-V	Reduced Instruction Set Computer - Five
ROM	Read Only Memory
RTOS	Real Time Operating System
SCL	Serial Clock Line
SD	Secure Digital Memory Card
SDA	Serial Data
SMD	Surface-Mounted Device
SNR	Signal Noise Ratio
SPAD	Single Photon Avalanche Diode
ST	STMicroelectronics
TCN	Temporal Convolutional Networks
TFL	Tensor Flow Lite
TFLM	Tensor Flow Lite Micro
TFT	Thin Film Transistor
Tiny-ML	Tiny Machine Learning
ToF	Time of Flight
UART	Universal Asynchronous Receiver Transmitter
VCSEL	Vertical Cavity Surface Emitting Diode
WAVE	Waveform Audio

ŞEKİL LİSTESİ

Şekil 2.1 Makine öğrenimi ve gömülü sistem üzerindeki Tiny-ML konumu 4
Şekil 2.2 Gecikme ve doğruluk karşılaştırması 5
Şekil 2.3 4 Referans uygulama için gecikme ve enerji sonuçları 6
Şekil 2.4 Tiny-ML uygulama alanları 8
Şekil 2.5 ST firmasının performansa göre mikro denetleyici serileri 9
Şekil 2.6 OpenMV Cam H7 / STM32H747I / STM32F746G kitleri 9
Şekil 2.7 Arduino Nano 33 BLE Sense geliştirme kiti 10
Şekil 2.8 TFLM uygulama akışı 11
Şekil 2.9 STM32Cube AI akışı 11
Şekil 2.10 Edge Impulse veri görüntüleme 14
Şekil 2.11 SensiML akışı 14
Şekil 2.12 SensiML etiketleme arayüzü 15
Şekil 2.13 Reality AI veri toplanma 15
Şekil 2.14 Reality AI portal arayüzü 16
Şekil 2.15 NanoEdge AI Studio uygulama akışı ve arayüzü 16
Şekil 2.16 Budama örneği 17
Şekil 2.17 Niceleme örneği 18
Şekil 2.18 Simetrik ve simetrik olmayan niceleme 18
Şekil 2.19 Model ağırlıkları nicelemesi 18
Şekil 2.20 Uçuş Süresi (ToF) ölçüm prensibi 19
Şekil 2.21 Uçuş Süresi (ToF) sensör alıcı ve vericisi 20
Şekil 2.22 Tekli ve çoklu piksel gösterimi 20
Şekil 2.23 Senkron ölçüm ve programlanabilir görüş açısı (FoV) 21
Şekil 2.24 Dolaylı uçuş süresi (iToF) sinyali 21
Şekil 2.25 Doğrudan uçuş süresi (dToF) sinyali 22
Şekil 2.26 Dolaylı uçuş süresi histogram modu 23
Şekil 2.27 ST çok bölgeli ToF sensörleri 25
Şekil 2.28 L5/L8 ve L7 sensör görüş açıları 25
Şekil 2.29 Koruyucu malzeme (Cam kapak) ve sinyal gürültüsü 26
Şekil 2.30 VL53L9CA 26
Şekil 2.31 TMF8820/21/22 SPAD bölge konfigürasyonu ve modları 27
Şekil 2.32 VL53L7CX 28

Şekil 2.33 Örnek robot üzerinde ToF sensörün konumu 29
Şekil 2.34 8x8 Mesafe, sinyal seviyesi, mesafe değişimi ve yansıma verileri . 29
Şekil 2.35 ToF görüş açısı ve piksellerin konumu 30
Şekil 2.36 Aydınlık ve karanlık koşulda ortam aydınlık verisi 30
Şekil 2.37 Ölçüm durumu bilgisi ile mesafe filtre örneği 30
Şekil 2.38 Hareket halindeki robot üzerinden alınan sinyal seviyesi değeri . . 31
Şekil 3.1 Kullanılan donanımlar 32
Şekil 3.2 VL53L7CX sensör devre şematiği ve PCB tasarımı 33
Şekil 3.3 VL53L7CX sensör akış diyagramı 33
Şekil 3.4 Veri toplama ve anlık görüntüleme yazılımı 34
Şekil 3.5 Veri toplama için kullanılan objeler 35
Şekil 3.6 Veri toplama deney düzeneği 35
Şekil 3.7 Veri aktarımı 36
Şekil 3.8 Sınıflandırma modeli 37
Şekil 3.9 Ağ katmanları 37
Şekil 3.10 Giriş ve yeniden şekillendirme katmanları 38
Şekil 3.11 1. 2 Boyutlu evrişim katmanı 38
Şekil 3.12 2. 2 Boyutlu evrişim katmanı 39
Şekil 3.13 Düzleştirme ve tam bağlantı katmanı 39
Şekil 3.14 Tüm CNN modeli 40
Şekil 4.1 Model özeti 41
Şekil 4.2 Sinyal seviyesi verisi öğrenim grafiği 42
Şekil 4.3 Yansıma verisi öğrenim grafiği 43
Şekil 4.4 Mesafe değişimi verisi öğrenim grafiği 44
Şekil 4.5 Mesafe verisi öğrenim grafiği 45

TABLO LİSTESİ

Tablo 2.1 Tiny-ML modellerinin gecikme değerleri 5
Tablo 2.2 Tiny-ML modellerinin enerji değerleri 6
Tablo 2.3 iToF ve dToF karşılaştırması 22
Tablo 2.4 VL53L7CX sensör çıktıları 28
Tablo 2.5 Yöntem için kullanılan sensör veri kümesi 31
Tablo 3.1 Veri sınıfları 36
Tablo 4.1 Sinyal seviyesi modeli karışıklık matrisi 42
Tablo 4.2 Yansıma verisi modeli karışıklık matrisi 43
Tablo 4.3 Mesafe değişimi verisi modeli karışıklık matrisi 44
Tablo 4.4 Mesafe verisi modeli karışıklık matrisi 45
Tablo 4.5 Özet model sonuçları 46

ÖZET

Çok Bölgeli Uçuş Süresi Sensörleri Kullanılan Otonom Cihazlar için TinyML Platformu ile Nesne Algılama

Sergen Erkan ARSLAN

Elektronik ve Haberleşme Mühendisliği Anabilim Dalı
Haberleşme Programı
Yüksek Lisans Tezi

Danışman: Doç. Dr. Bülent BOLAT

Son yıllarda, üzerinde birçok sensör barındıran otonom robotların hayatımızdaki sayısı ve önemi her geçen gün artmaktadır. Büyüyen robot pazarı ile otonom robot araştırmaları her zamankinden daha çok ön plana çıkmaktadır. Çoğu otonom robot engelleri tespit etmek ve navigasyonu gerçekleştirmek için LIDAR, uçuş süresi (ToF) sensörü, ToF kamerası, derinlik kamerası, geniş açılı kamera, IR optik sensörler ve elektromekanik anahtarlar kullanmaktadır. Kamera tabanlı sensör sistemleri büyük miktarda işlem gücü ve hafıza gerektirmektedir. Dolayısıyla bu tür sistemlerin maliyetleri daha yüksektir. Ayrıca özel hayat gizliliği konusunda kullanıcı dostu değildir. Öte yandan daha basit sensör uygulamaları yüksek hata oranlarına ve daha kötü performansa sahiptir. Tek nokta tabanlı optik sensörler, otonomun gerekliliği olan küçük engellerin tespiti için uygun değildir. Bu tür sensörler için özellikle kablolar ve oyuncaklar gibi küçük nesnelerin tespit edilmesi oldukça zordur.

Yapılan bu çalışmada, belirtilen sorunların üstesinden gelebilmek için çok bölgeli uçuş süresi (ToF) sensörü kullanılarak ve engelin varlığı ile konumunu tespit edebilen makine öğrenmesi modeli önerilmiş olup mikro denetleyici üzerinde koşabilen sıkıştırılmış ve optimize edilmiş makine öğrenimi (Tiny-ML) kullanılmıştır. Günümüzde hafıza ve işlem hızı olarak gelişen 32-bit ARM tabanlı mikro denetleyici sayısı günden güne artmakta ve internete bağlanabilen akıllı cihazlarda kullanımı yaygınlaşmaktadır. ARM tabanlı STM32F746NG

200 MHz saat hızına sahip mikro denetleyici kullanılarak robotun hareketi sırasında engellerin hızlı ve gerçek zamanlı olarak algılanması gerçekleştirilmiştir. Oluşturulan makine öğrenmesi modellerinin başarımları karşılaştırılmış ve sonuç olarak en yüksek başarıma sahip makine öğrenmesi modeli belirlenmiştir.

Anahtar Kelimeler: ToF, Tiny ML, Lidar, Kamera, Mikro denetleyici

ABSTRACT

Tiny Machine Learning Model for Object Detection with Multi Zone Time of Flight Sensors

Sergen Erkan ARSLAN

Department of Electronics and Communication Engineering
Master of Science Thesis

Supervisor: Assoc. Prof. Bülent BOLAT

In recent years, the number and importance of autonomous robots with various sensors in our lives have been increasing day by day. With the growing robot market, research on autonomous robots is more prominent than ever. Most autonomous robots use LIDAR, Time-of-Flight (ToF) sensors, ToF cameras, depth cameras, wide-angle cameras, IR optical sensors, and electromechanical switches to detect obstacles and perform navigation. Camera-based sensor systems require a significant amount of processing power and memory, making the costs of such systems higher. Additionally, they are not user-friendly regarding personal privacy. On the other hand, simpler sensor applications have high error rates and less performance. Single-point optical sensors are not suitable for detecting small obstacles required for autonomy. Detecting cables and toys is particularly challenging for such sensors. To overcome these challenges, a machine learning model that can produce the position and distance output of obstacles using a multi-zone Time-of-Flight (ToF) sensor has been proposed.

In this study, compressed and optimized machine learning (Tiny-ML) running on a microcontroller has been utilized. The number of 32-bit ARM-based microcontrollers is increasing day by day, and they are used in smart devices that can connect to the internet, benefiting from their improved memory and processing speed. Using an ARM-based STM32F746NG microcontroller with a clock speed of 200 MHz, obstacle detection during the robot's movement has been achieved quickly and in real-time. The accuracies of the created machine learning models have been compared. As a result, the most suitable machine learning model and

sensor have been identified.

Keywords: Time of flight, Tiny machine learning, Lidar, Camera, Microcontroller

1
GİRİŞ

1.1 Tezin Amacı

Bu tezin amacı, otonom robotlar için çok bölgeli uçuş süresi (ToF) sensörü ve optimize edilmiş makine öğrenimi (Tiny-ML) kullanılarak mikro denetleyici üzerinde çalışabilen gerçek zamanlı olarak engel olabilecek nesnelerin varlığını ve bölgesini algılamaktır. Ayrıca kurulan deney sistemi ile çok bölgeli uçuş süresi sensörü üzerinden alınan birden fazla verinin makine öğrenmesi modelindeki başarım sonuçlarının karşılaştırılması amaçlanmıştır. Sonuç olarak bu çalışmada, uçuş süresi sensörü kullanılarak optimize edilmiş makine öğrenimi (Tiny-ML) modeliyle %90 doğruluğun üzerinde, maliyet ve enerji verimli bir çözüm üretilmesi amaçlanmıştır.

1.2 Literatür Araştırması

Günümüzde otonom hareket edebilen robotlar evlerde gün geçtikçe artmaktadır. Özellikle temizlik robotları yaygın bir şekilde kullanılmaktadır. Bu robotlar içerisinde birden fazla sensör bulundurmaktadır. Bu sensörler haritalama, duvar takibi, uçurum algılama, toz yoğunluğu ve engel olabilecek nesneler algılama gibi birden fazla amaç için kullanılmaktadır. Bu çalışma, ev tipi robotların hareketine engel olabilecek nesnelerin algılanması için yapılmıştır.

Robotlarda engel algılama için birçok çözüm yöntemi bulunmaktadır. En basit çözüm yöntemi birden fazla elektromekanik anahtarlara bağlı olan bir tampon tasarımıdır [1]. Bu tasarımda robotun engele çarpmasıyla engelin hangi bölgesinde olduğunu elektromekanik anahtarlar ile robot algılayabilmektedir. Algılama işleminden sonra robot durur ve ters yönde hareket ederek engelden kaçar. Bu tasarım için alınmış bir patent bulunmaktadır [2]. Tampon hareketini algılayan elektromekanik anahtarlar yerine manyetik sensör kullanılarak da algılama gerçekleştirilebilmektedir [3]. Engele çarparak fiziksel algılama gerçekleştiren bu çözümlerle birlikte gelişmiş sensör çözümleri de kullanılmaktadır.

Ancak, gelişmiş sensör çözümlerinin engeli algılayamaması gibi durumlarda fiziksel tampon hareketi algılanarak robotun hareketi kesilir ve güvenlik sağlanır. Bu nedenle tampon hareketi algılayan çözümler, gelişmiş sensör çözümlerinin hatalarını kapatmak için kullanılmaktadır.

Gelişmiş sensör çözümü olarak genellikle kamera kullanılmaktadır [4]. Önerilen bu yöntem, robot üzerinde kızılötesi (IR) vericisini kullanarak yatay bir kızılötesi ışığını zemine bir çizgi halinde yansıtmaktadır. Robot üzerinde bulunan geniş açılı bir kamera ile zemin veya engel tarafından geri yansıtılan kızılötesi çizgisini içeren bir görüntü yakalar. Yakalanan bu görüntüde, kızılötesi çizgisine ait piksellerin görüntü koordinatları kullanılarak engelin varlığı tespit edilir. Bir başka çözüm olarak derinlik kamerası da kullanılmaktadır [5]. Bu makalede atalet ölçü ünitesi (IMU) ve derinlik kamerası birlikte kullanılarak hem ortamın haritalandırılması sağlanmakta hem de engellerin algılanarak robotun hareket rotası çıkarılmaktadır. Derinlik hesabı için görüntüleme sistemi içerinde iki adet kamera bulunmaktadır. Tek kamera kullanan yapılarda ise ToF teknolojisi yer almaktadır [6]. Bu makale, otonom sürüş sisteminin bir parçası olarak entegre edilen ToF teknolojisini kullanarak, sürücünün kör nokta bölgesinde bir engel olması durumunda uyarı alarmı üreten bir algoritmayı aktarılmaktadır. Nesnelerin algılaması için yüksek hızda ToF kamera uygulaması da bulunmaktadır [7]. Bu uygulama, CNN modeli kullanılarak görüntü işleme içermektedir.

Robot üzerinde kullanılan kamera tabanlı çözümlere alternatif olarak çoklu bölge uçuş süresi (ToF) sensörü düşünülebilir [8]. Bu yayında robot üzerine yerleştirilen çok bölgeli ToF sensörleri ile 360 derecede engellerin tespiti için bir tasarım önerilmektedir. Bu tasarım, robotlar üzerinde haritalama için kullanılan LIDAR sensör yerine de kullanılabilmektedir [9]. Tek bölgeli ToF sensörler için de geliştirilmiş uçurum algılama çalışması bulunmaktadır [10]. Ek olarak birçok temas gerektirmeyen uygulamalar da yer almaktadır [11]. Uygulama örneklerinden bazıları; tuvalet sifonu, musluk, dispanser, akıllı anahtar, halka açık ekran uygulamaları, hareket tanıma, endüstriyel robotlar, üretim hatları, kullanıcı dostu arayüz uygulamaları, içerik yönetimi, kahve makinesi sıvı ölçümü, depolar, otomatlar, ATM, dolaplar vb. olarak sıralanabilir.

Literatürde çoklu bölge ToF sensörleri için özel olarak tasarlanmış Tiny-ML çalışmaları bulunmaktadır. Tiny-ML, mikro denetleyici üzerine uygulanabilen küçük makine öğrenmesi çalışmalarıdır. Bu çalışmaların çoğunluğu el hareketi tabanlı uygulamalardır [12]. Bu yayında, küçük TCN modelini kullanarak hareket tanıma gerçekleştirilmektedir. Bu çalışmaya benzer olarak ToF sensör üreticileri tarafından da sağlanan hareket algılama yazılımları sunulmaktadır [13]. Hareket

türünün tespiti için tasarlanmış bir Tiny-ML uygulaması bulunmaktadır [14]. Bu uygulama belirli el hareketlerini sınıflandırabilmektedir. Bu hareketler ise sağa, sola, yukarıya, aşağıya kaydırma ve dokunma hareketleridir. Bu konu ile ilgili daha gelişmiş bir uygulama da bulunmaktadır [15]. Bu uygulamada ise sınıflandırdığı el hareketleri evrensel olarak kabul görmüş aktif olarak kullanılan hareketlerdir. Örneğin; dur, mola, beğenme, beğenmeme, yumruk ve kalp gibi tek elle veya 2 el ile yapılan hareketler. El hareketleri dışında ToF sensörler ile robot üzerinde uygulanan çalışmalar da bulunmaktadır [16]. Bu uygulamada zeminin türü yumuşaklık seviyesine göre belirlenebilir. Robot, yumuşak veya sert zemin tespiti yapmaktadır. Robot, yumuşak zemin olarak halı, sert zemin olarak ise parke ve fayans gibi zeminlerde eğitilmiş bir modele sahiptir. Model çıktısına göre robot temizleme modunu değiştirebilmektedir.

Yapılan incelemelerde robot üzerinde kullanılan ToF sensörlerin geniş bir uygulama alanı bulunmaktadır. Ancak engel tespiti için ToF sensörü yerine kamera kullanımı yaygındır. Kamera içeren uygulamaların özel hayatın gizliliği, kamera ve görüntü işleme maliyetleri gibi dezavantajları bulunmaktadır.

Tüm bu incelemeler, özel hayatın gizliliğini koruyan ve maliyet olarak uygun olan çok bölgeli ToF sensörünün kullanılma gerekliliğini göstermektedir. Ek olarak Tiny-ML yapısı ile ToF sensörün birlikte kullanımı engel algılama problemine çözüm getirecek motivasyonu vermektedir.

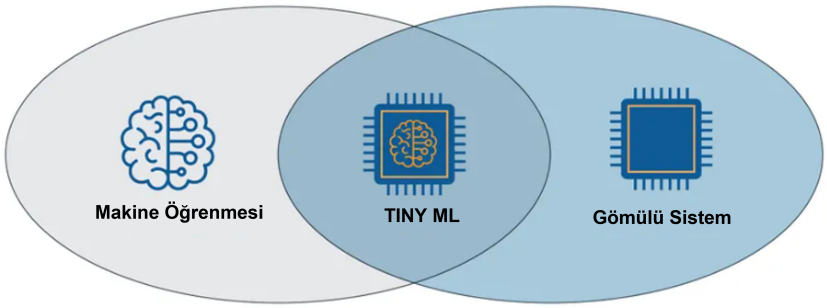
1.3 Hipotez

Evlerimizde ve ortak kullanım alanlarında farklı amaçlar ile kullanılan otonom robotların hareketini engelleyebilecek nesnelerin hızlıca algılanması istenilmektedir. Engel algılama işlemi, algılama mesafesi ve robotun hızına göre optimize edilmelidir. Böylelikle engelin algılanması durumunda, robotun hareket rotasını değiştirilebilmesi için süresi yeterli olacak. Engel algılama süresinin kısa olması robotun durmadan akıcı bir şekilde engellerden kaçabilmesini sağlayacaktır. Çok bölgeli uçuş süresi sensör ve mikro denetleyicilerde uygulanabilen makine öğrenmesi kullanılarak oluşturulan bu çalışmada, engelin algılanması ve oluşturulan modellerin başarım oranlarından bahsedilmektedir.

2
YÖNTEM

2.1 TINY ML

Tiny-ML, yapay zekanın yeni bir sınıfını temsil eden ve makine öğrenmesi modellerini mikro denetleyiciler (MCU) üzerinde çalıştırmak için kullanılan gömülü bir makine öğrenimi sistemleri bütünüdür (Şekil 2.1). Tiny-ML genel olarak donanımlar, algoritmalar, cihaz içi sensörlerin mikro denetleyici içerisinde veri analitiğini gerçekleştirebilen ve yazılımları da dahil olmak üzere hızla büyüyen bir makine öğrenimi teknolojileri ve uygulamaları alanı olarak da tanımlanabilir. Bu teknoloji, mW seviyesi ve altındaki sınırlı kaynaklara sahip cihazlarda yapay zeka kullanımının önünü açarak geniş bir uygulama potansiyeli sunmaktadır.



Şekil 2.1 Makine öğrenimi ve gömülü sistem üzerindeki Tiny-ML konumu

Günümüzde nesnelerin interneti (IoT) uygulamalarının yaygınlaşmasıyla birlikte yapay zeka uygulamaları giderek önem kazanmaktadır. Cihazlar üzerinden alınan verilerin bulut ortamına aktarılıp bulut üzerinde makine öğrenmesi modelinin çalıştırılması, kullanıcı tarafında güvenlik, gecikme ve maliyet gibi birçok dezavantajlı durum yaratmaktadır. Bu nedenle gömülü sistem üzerinde çalışabilen makine öğrenmesi uygulamaları ön plana çıkmaktadır. Bu uygulamaların geliştirilmesi için birçok geliştirme kartı ve makine öğrenmesi platformu bulunmaktadır.

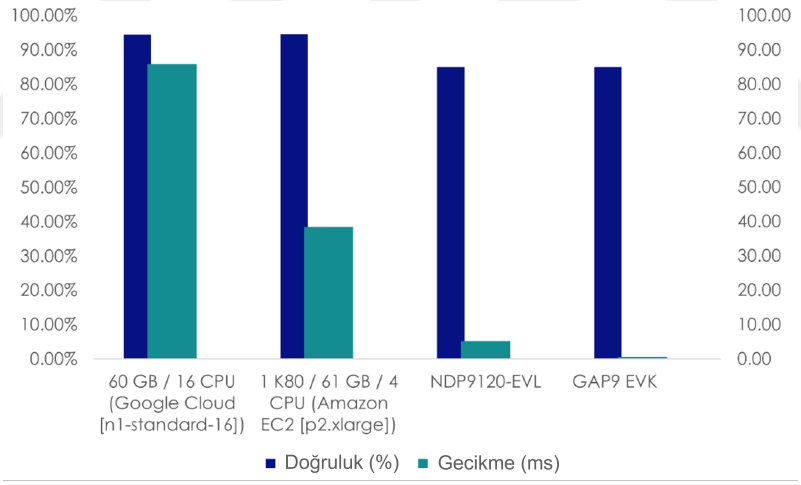
2.1.1 Tiny ML Avantajları ve Zorlukları

Tiny-ML, bulutta çalışan uygulamalara ve diğer kompleks gömülü sistemlere göre birçok avantaj içermektedir. Bu avantajların detayları aşağıdaki gibidir.

- Gecikme Süresi

  Tiny-ML sistemleri, model çıkarımı için veriyi herhangi bir sunucuya aktarmaz. Çünkü makine öğrenimi işlemleri sistemin kendisi tarafından yürütülür. Bu nedenle veri transferi ile zaman kaybedilmez. Bu da gecikmeyi mümkün olduğunca azaltır.

  Görüntü ve konuşma tanıma gibi gerçek zamanlı kararlar gerektiren uygulamalar için gecikme süresi kritik bir öneme sahiptir. Şekil 2.2, Google ve Amazon sunucuları ile Tiny-ML destekli iki kartın (NDP9120-EVL Kartı ve GAP9 EVK Kartı) ResNet görüntü sınıflandırma modelinin gecikme süresi ve doğruluğu arasındaki karşılaştırmayı göstermektedir [17]. Şekil 2.2 'de sunulan sonuçlar, Tiny-ML sistemlerinde doğruluk oranlarının çok değişmediği ve gecikmeyi önemli ölçüde azalttığını göstermektedir.



Şekil 2.2 Gecikme ve doğruluk karşılaştırması

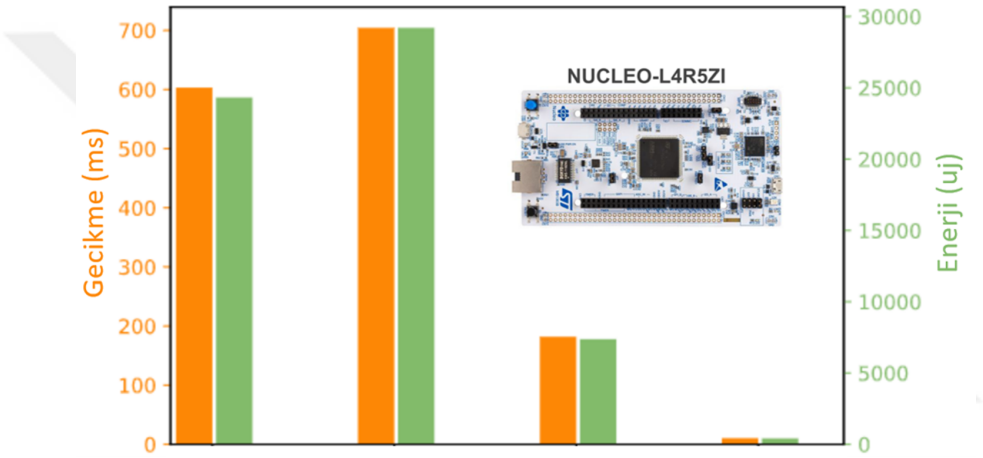
Ek olarak ST firmasına ait mikro denetleyiciler için bağımsız bir kuruluşun yaptığı farklı modellerdeki gecikme değerleri Tablo 2.1 'de verilmiştir [18].

Tablo 2.1 Tiny-ML modellerinin gecikme değerleri

Geliştirme Kartı	DSCNN	FC AutoEnc	MobNetV1	ResNetV1
NUC-H7A32ZI-Q	15,35 ms	1,82 ms	29,63 ms	51,95 ms
NUC-L4R5ZI	62,87 ms	6,86 ms	118,69 ms	214,00 ms
NUC-U575ZI-Q	38,58 ms	4,85 ms	71,56 ms	128,21 ms
DISCO-F746NG	17,75 ms	1,71 ms	34,08 ms	62,88 ms

- Güç Tasarrufu

  Mikro denetleyiciler, enerji kaynağı kısıtlı olan uygulamalarda son derece düşük güç tüketen cihazlarda bulunmaktadır. Böylece Tiny-ML işlemlerini gerçekleştirmek için gerekli olan gücü azaltır. Ayrıca bu işlemler, veri transferleri ve giriş-çıkış (IO) gerektirmediğinden dikkate değer derecede verimlidir. Bu nedenle Tiny-ML uygulamalarının diğer makine öğrenmesi uygulamalarına göre daha düşük enerji tüketimi vardır. Şekil 2.3 'te anahtar kelime algılama, görsel uyandırma, görüntü sınıflandırma ve anormallik tespiti referans uygulamaları için sırasıyla enerji ve gecikme sonuçları gösterilmektedir [19]. Yapılan bu testler NUCLEO-L4R5ZI kartında geçekleştirilmiştir.



Şekil 2.3 4 Referans uygulama için gecikme ve enerji sonuçları

Ek olarak ST firmasına ait mikro denetleyicilerinin bağımsız bir kuruluş tarafından yapılan enerji değerleri karşılaştırması aşağıdaki Tablo 2.2 'de verilmiştir [18].

Tablo 2.2 Tiny-ML modellerinin enerji değerleri

Geliştirme Kartı	DSCNN	FC AutoEnc	MobNetV1	ResNetV1
NUC-H7A32ZI-Q	1.869,38 uj	221,22 uj	3.699,35 uj	6.310,69 uj
NUC-L4R5ZI	2.833,14 uj	293,91 uj	5.618,28 uj	7.019,09 uj
NUC-U575ZI-Q	1.004,22 uj	121,45 uj	1.920,64 uj	3.383,82 uj

- Bant Genişliği Tasarrufu

  Bulut çözümlü yapay zeka uygulamalarının aksine Tiny-ML uygulamalarında model çıkarımı için internet bağlantısı kullanmaz. Veriler, bir sunucuya transfer edilmek yerine cihazda toplanır ve işlenir. Bu nedenle, Tiny-ML sistemleri bant genişliği açısından son derece verimlidir.

- Daha Güçlü Gizlilik ve Veri Koruma

  Tiny-ML uygulamaları sunucularda veri bulundurmaz. Böylece kötü niyetli insanların sunucuları ele geçirmesi durumunda hassas bilgilere erişilmesini imkansız kılar. Genel olarak gömülü bir cihazda makine öğrenimi modellerinin yürütülmesi daha güçlü veri gizliliği sağlamaktadır.

Gömülü cihazlarda ve mikro denetleyicilerde makine öğrenimi geliştirmek, dağıtmak ve çalıştırmakla ilgili bazı zorluklar ve kısıtlamalar bulunmaktadır. Ana zorluklardan biri veri toplama süreçleri ile ilgilidir. Sınırlı enerji miktarı nedeniyle Tiny-ML eğitim görevleri için büyük miktarda veri toplamak oldukça zordur. Bu sebep, gömülü makine öğrenimi ve Tiny-ML görevleri için sınırlı veri setlerinin bulunmasına neden olur.

Başka bir zorluk, büyük miktarda bilgiyi işlemek için hesaplama kapasitesi ve belleğin eksikliğidir. Bu durum genellikle, mikro denetleyiciler ve diğer gömülü sistemler için de belirli görevleri (örneğin, video sahne analizi) gerçekleştirmekte bir engeldir. Ek olarak Tiny-ML uygulaması geliştirmek ve üretmek tecrübe ile ilişkilidir. Gömülü sistemler ve veri bilimi uzmanlığını birleştiren çok disiplinli ekipler gerektirir. Bu nedenle uygulamanın geliştirilmesi zorlayıcı olabilmektedir.

2.1.2 Uygulama Alanları

Tiny-ML, sıkıştırılmış ve optimize edilmiş makine öğrenimi modellerini mikro denetleyiciler ve gömülü sistemler gibi küçük ve düşük güçlü cihazlardaki uygulanmalara odaklanmaktadır. Ayrıca düşük gecikme süresine, sınırlı işlem gücüne ve düşük maliyete sahip uygulamalarda sıklıkla kullanılmaktadır. IoT cihazları ve mikro denetleyicilerin kullanım kolaylığının artmasıyla uygulama alanları yaygınlaşmaktadır.

Tiny-ML modelleri, fabrika otomasyonu ve endüstriyel süreçlere veri analizi ve karar verme yetenekleri ekleyerek daha akıllı ve verimli sistemler oluşturabilir. Ek olarak ev otomasyonu, akıllı enerji yönetimi ve akıllı tarım alanlarında da kullanılabilir.

Sağlık izleme cihazları, hasta takibi ve tıbbi teşhis sistemlerinde Tiny-ML modelleri kullanılabilir. Özellikle taşınabilir cihazlar aracılığıyla gerçek zamanlı sağlık verilerini analiz etmek ve anlamak için kullanışlıdır.

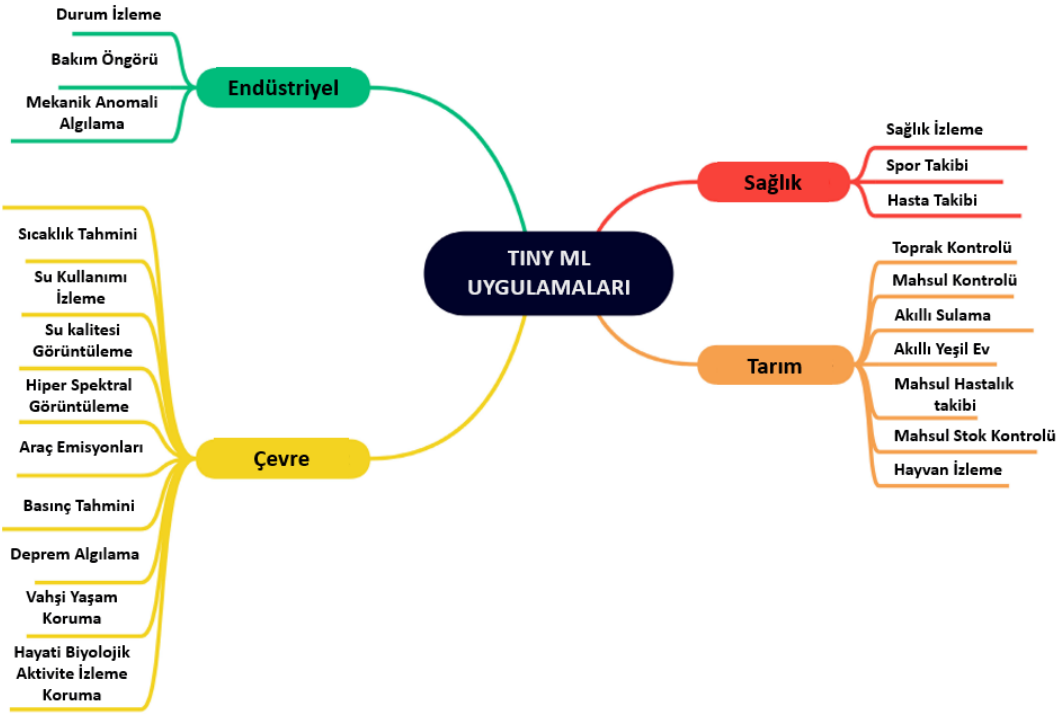
Giyilebilir teknoloji ile kullanıcı davranışlarını ve verilerini analiz etmek için Tiny-ML modelleri kullanılabilir. Böylece gerçek zamanlı geri bildirimler

sağlanabilir ve kişiselleştirilmiş hizmetler sunulabilir.

Otomotiv endüstrisinde sürücü asistanı sistemleri, araç içi ses kontrolü ve sürücü davranışı analizi gibi uygulamalarda Tiny-ML kullanılabilir.

Gözetim kameraları ve güvenlik sistemleri, Tiny-ML modellerini kullanarak algılama, tanıma ve sınıflandırma yeteneklerini artırabilir. Bunun sonucunda daha akıllı ve etkili güvenlik çözümleri sağlayabilir.

Yukarıda verilen bu örnekler Tiny-ML uygulama alanlarının birkaçıdır. Birçok farklı endüstride ve uygulamada potansiyel olarak kullanılabilir ve giderek daha yaygın hale gelmektedir (Şekil 2.4) [17].



Şekil 2.4 Tiny-ML uygulama alanları

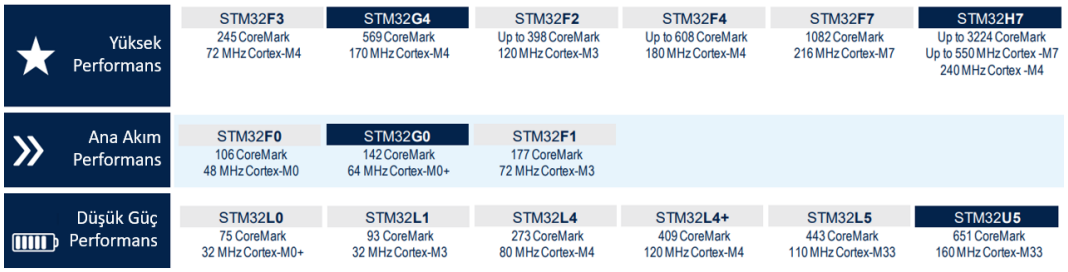
2.1.3 Tiny-ML Geliştirmeye Yönelik Araçlar ve Teknikler

Tiny-ML; düşük güç tüketimi, sınırlı bellek ve işlemci kaynaklarına sahip gömülü sistemler için optimize edilmiş yapay zeka modellerinin geliştirilmesini sağlayan bir alanı ifade eder. Bu alandaki mikro denetleyicilerde (örneğin, Arduino, STM32, NXP, Renesas, ESP) veya diğer gömülü cihazlarda (Raspberry Pi, Pico, mini bilgisayar) çalışacak şekilde tasarlanmış makine öğrenimi modelleri oluşturmak için özel araçlar ve teknikler kullanılır. Bu araçlar ve teknikler model boyutunu azaltma, hesaplama gereksinimlerini düşürme ve performans ölçme gibi zorluklarla başa çıkmak için tasarlanmıştır.

2.1.3.1 Donanım

Makine öğrenim modellerini mikro denetleyiciler üzerinde çalıştırmak için genellikle Google tarafından geliştirilen açık kaynaklı bir kütüphaneye sahip olan TensorFlow-Lite-Micro (TFLM) kullanılır. TFLM, sınırlı kaynaklara sahip mikro denetleyicilerde makine öğrenimi modellerini çalıştıran bir kütüphanedir. Gömülü sistemlerde yapay zeka uygulamalarını hızlı ve etkili bir şekilde entegre etmeyi mümkün kılar. ARM-Cortex-M3 mimarisine sahip olan bir mikro denetleyicide yalnızca 16 KB hafızaya sığar ve birçok temel modeli çalıştırabilir. İşletim sistemi desteğine ve herhangi bir C kütüphanesine ihtiyaç duymaz. Bu nedenle günümüzde kullanılan birçok 32 bit platforma sahip olan mikro denetleyiciyi desteklemektedir. Tiny-ML uygulamaları için geliştirme kitleri arasında en çok kullanılanlar aşağıdaki gibidir [20].

ST firmasına ait birçok farklı performansta mikro denetleyici ve geliştirme kiti bulunmaktadır. Şekil 2.5 'te görülen mikro denetleyici serileri makine öğrenmesi için önerdiği serilerdir.



Şekil 2.5 ST firmasının performansa göre mikro denetleyici serileri

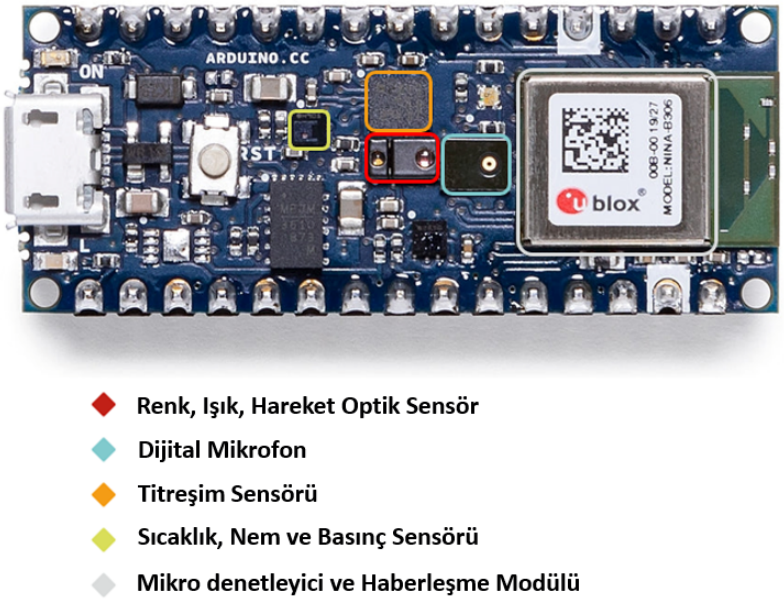


Şekil 2.6 OpenMV Cam H7 / STM32H747I / STM32F746G kitleri

ARM-Cortex-M7 mimarisine sahip mikro denetleyiciler üzerinde görüntü işleme amacıyla önerdiği kitler bulunmaktadır (Şekil 2.6).

Arduino firmasının ürettiği ve kendi geliştirme ortamında da programlanabilen kitte ARM-Cortex-M4 mikro denetleyici, birçok sensör ve haberleşme modülü

barındırmaktadır (Şekil 2.7). Bu sayede yapılacak olan uygulamalara alt yapı sağlamaktadır [21].



Şekil 2.7 Arduino Nano 33 BLE Sense geliştirme kiti

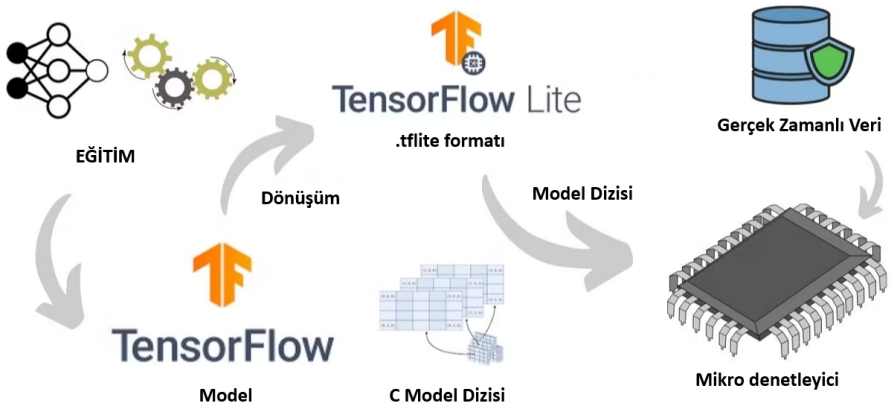
2.1.3.2 Yazılım Kütüphaneleri

Tiny-ML projelerini geliştirmek için öncelikle modelin oluşturulması, eğitilmesi, sıkıştırılması ve optimize edilerek düşük hafızaya sahip bir mikro denetleyici donanımına yüklenmesi gerekmektedir. Tüm bu proje akışındaki maddeler için birden fazla açık kaynak yazılım kütüphanesi bulunmaktadır [22][23]. Yaygın kullanımı olan yazılım kütüphaneleri aşağıda verilmiştir.

- TensorFlow Lite Micro (TFLM)

  TensorFlow Lite (TFL), makine öğrenimi görevlerini gerçekleştirmek için Google tarafından geliştirilen popüler bir açık kaynaklı yazılım kütüphanesidir. Mikro denetleyiciler için TFL yani TFLM, TFL alt yapısından türetilmiş bir projedir [24]. Böylelikle mikro denetleyiciler üzerinde makine öğrenmesi modellerini çalıştırmamızı sağlamaktadır. Bu yazılım kütüphanesi C++ 17 ve 32 bit mikro denetleyicileri desteklemektedir. Uygulama akışı Şekil 2.8 'de gösterilmiştir.

  Mikro denetleyicilerin hafızalarının limitli olması sebebi ile modelin eğitilmesi, bilgisayar veya uzak sunucular vasıtasıyla yazılım kütüphanesi kullanılarak gerçekleştirilmektedir. Eğitim bittikten sonra test teknikleriyle model değerlendirmesi yapılarak modelin doğruluğunu ve güvenilirliği test edilmelidir. Oluşturulan model, tflite uzantılı formata çevrilerek mikro

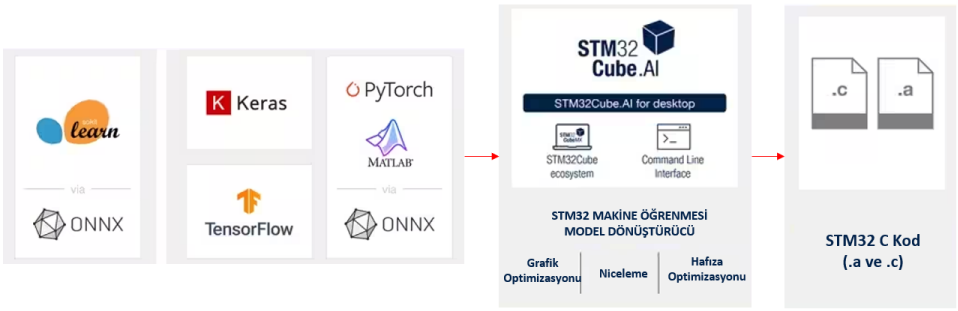


Şekil 2.8 TFLM uygulama akışı

denetleyicilerde çalışabilen C model dizisine dönüşümü yapılmaktadır. Tflite, TFL yazılım kütüphanesine ve donanıma uyumlu olan bir dizi ara bellek dosyasıdır. Elde edilen tflite uzantılı model dosyası, kullanılan derleyici ile mikro denetleyiciye uygun hale getirilip yüklenebilmektedir. Sonuç olarak mikro denetleyici üzerinde çalışabilen model, belirtilen giriş verilerinin gerçek zamanlı olarak makine öğrenmesi modeline aktarıldığında çıktı üretebilmektedir.

- STM32Cube AI

  STM32Cube AI, mikro denetleyici üretici firması olan ST 'nin geliştirdiği mikro denetleyici ailesi üzerinde makine öğrenmesi modellerini çalıştırmayı sağlayan bir yazılım paketidir. Keras ve TensorFlow Lite dahil çeşitli çerçevelerden eğitimli modellerini Şekil 2.9 'da gösterildiği gibi destekler [25].



Şekil 2.9 STM32Cube AI akışı

STM32Cube-AI, STM32CubeMX ve STM32CubeIDE gibi diğer ST firmasının geliştirdiği araçlarla tamamen uyumlu bir yazılım paketidir. Diğer yazılımlar ile uyumlu olması basit ve kullanımı kolay bir akış sağlar. Ayrıca tek satır kod yazmaya gerek kalmadan, çıkarım süresi ve hafıza kullanımına ilişkin ölçümler sağlamaktadır. Oluşturulan modeli, gerçek bir test veri

seti ile hedefe yönelik olarak değerlendirmek için bir test uygulaması da bulunmaktadır.

STM32Cube-AI ayrıca CMSIS-NN 'ye üzerinden optimize edilmiş bir çıkarım motoruna da sahiptir. Ancak kapalı bir yapısı olduğundan nasıl optimize edildiği belli değildir. Bu sebeple, bu yazılım kütüphanesini değiştirmek veya genişletmek mümkün değildir. Ek olarak ST firmasına ait olmayan mikro denetleyicilerde kullanılamaz.

- MicroTensor

  MicroTensor, bulut uygulama tabanlı cihazlarda prototip oluşturmaya ve hızlı model geliştirmeye yardımcı olan ücretsiz bir yerleşik öğrenme ortamıdır [26]. Bir çıkarım motoru ve grafik işleme aracı içerir. Eğitim için Keras kullanarak bir sinir ağı modeli oluşturur ve eğitilen modeli C / C++ diline dönüştürür. MicroTensor modeli Mbed, ST ve K64 kartlarında kullanılabilir. Ayrıca hafızada yalnızca 2 KB gerektiren küçük boyutlu bir modüldür.

- Embedded Learning Library (ELL)

  Microsoft, Tiny-ML ekosistemini desteklemek amacıyla ELL 'i geliştirmiştir. Raspberry Pi, Arduino ve Microbit platformları için destek sağlamaktadır. Eğitim için Scikit-learn ve Keras 'ı desteklemektedir. Eğitilen modeli C diline ve emlearn-micropython entegrasyonu ile micropython diline de dönüştürebilmektedir [27]. Birçok eğitim modelini desteklemektedir [28].

- MicroTVM

  MicroTVM; TFL, ONNX, PyTorch gibi birçok popüler yazılım kütüphanelerinden modelleri kabul edebilen ve bunları mikro denetleyicilere kadar derleyen açık kaynaklı bir derin öğrenme derleyicisidir [29]. Çeşitli platformlarda geliştirilen modelleri desteklemesi ile ön plana çıkmaktadır. Ek olarak RISC-V, Cortex-M mimarisi tabanlı 8 bit ve 32 bitlik mikro denetleyicileri desteklemektedir. Dolayısıyla desteklediği donanım çeşitliliği yüksektir.

- Artificial Intelligence for Embedded Systems (AIfES)

  TensorFlow, Keras veya PyTorch gibi iyi bilinen birçok yazılım kütüphanesi ile uyumludur [30]. Mevcut sürümde, tamamen serbestçe yapılandırılabilen ileri beslemeli ve evrişimli sinir ağları desteklenmektedir. ReLU, Sigmoid veya Softmax gibi genel aktivasyon fonksiyonları da entegre edilmiştir. 100 KB 'ın biraz üzerinde RAM 'e sahip kaynak kısıtlı cihazlarda bile CNN modellerini verimli bir şekilde eğitebilmektedir. Bu sayede neredeyse her donanımda çalışabilmektedir. Mikro denetleyiciler için TFLM ile

karşılaştırılmış ve evrişimli sinir ağları kullanıldığında bellek tüketimini %54'e kadar azaltmıştır [31].

- Neural Network on Microcontroller (NNoM)

  NNoM, özellikle mikro denetleyiciler için yüksek seviyeli, katman tabanlı bir sinir ağı kütüphanesidir [32]. NNoM 'nin amacı, hızlı model geliştirilmesi için kullanıcı dostu ve esnek bir arayüz sağlamaktır. NNoM; geliştirici için yapıyı, belleği ve diğer parametreleri yönetmesine izin vermektedir. NNoM yazılım kütüphanesi, Keras modellerini kaynak koduna çevirebilmektedir [32]. NNoM kütüphanesinin TensorFlow Lite ve STM32Cube AI araçlarıyla karşılaştırmaları bulunmaktadır [33].

2.1.3.3 Platformlar

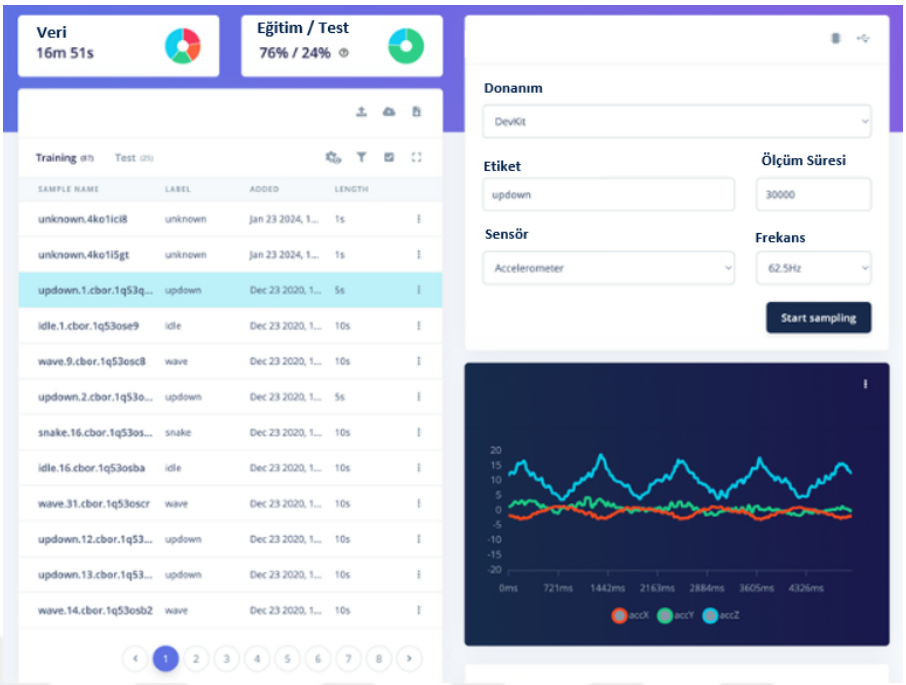
Tiny-ML platformları; üst başlıkta aktarılan yazılım kütüphanelerini sıklıkla kullanan, toplanan verilerin aktarıldığı veya veri toplayabilen, öğrenme modelinin seçilebildiği, eğitimin yapıldığı, çıkan modelin test edildiği ve ihtiyaç duyulan donanıma uygun bir şekilde çıktı veren sistemlerdir. Kullanıcının ihtiyaç duyduğu tüm alt yapıyı sağlamayı amaçlayan sistemlerdir. Yaygın kullanımı olan platform detayları aşağıdaki gibidir.

- Edge Impulse

  Gömülü sistemlerde çalışacak şekilde optimize edilmiş yapay zeka ve makine öğrenimi modelleri oluşturmayı ve dağıtmayı sağlayan bir platformdur. Bu platform; IoT cihazları, akıllı nesneler, giyilebilir teknolojiler ve diğer gömülü sistemlerde yapay zeka uygulamaları geliştirmek için ideal bir çözüm sunar [34] [35].

  Kullanıcıların sensör verilerini kolayca toplamasına, etiketlemesine, analiz etmesine ve model oluşturmasına olanak tanır (Şekil 2.10). Ek olarak, veri aktarma ve toplama için birçok format desteklenmektedir [36].

  Toplanan verileri kullanarak makine öğrenimi modelleri oluşturabilir ve bu modelleri eğitebilirler. Edge Impulse, geliştiricilerin oluşturdukları bu modelleri çeşitli gömülü sistemlere entegre etmelerini sağlar [37]. Bu platform, C++ kütüphanesi olarak modeli gömülü cihazlara entegre edilmesinin ilk aşamasını sağlar. Sonrasında kullanılan gömülü sisteme uygun bir derleme ortamına kütüphanenin eklenmesi ve derlemesi ile modelin gömülü sisteme transferi tamamlanmış olur [38].

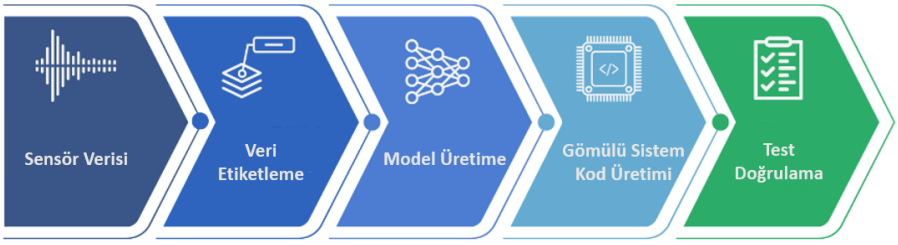


Şekil 2.10 Edge Impulse veri görüntüleme

Edge Impulse, kullanıcıların oluşturdukları modelleri test edebilmektedir. Test sonucuna göre iyileştirme yaptıktan sonra tekrar eğitebilir ve modellerinin performanslarını arttırabilir.

- SensiML

QuickLogic'in bir yan kuruluşu olan SensiML, uygulama geliştiricilerinin hızlı bir şekilde akıllı cihazları oluşturmasına olanak tanıyan ve ham sensör verilerini işlemek için makine öğrenmesi yazılım araçları sunmaktadır [39]. Platformu kullanarak yapılacak uygulamanın akışı Şekil 2.11 'de gösterilmektedir.



Şekil 2.11 SensiML akışı

Sensör verilerinin gerçek zamanlı toplanması ve aktarılması, görüntülenmesi ve etiketlenmesi sağlayan bir arayüzü bulunmaktadır (Şekil 2.12) [40]. Tüm arayüzün çalışma sistemini ve alt detaylarını aktaran kapsamlı dokümantasyonu mevcuttur [41].

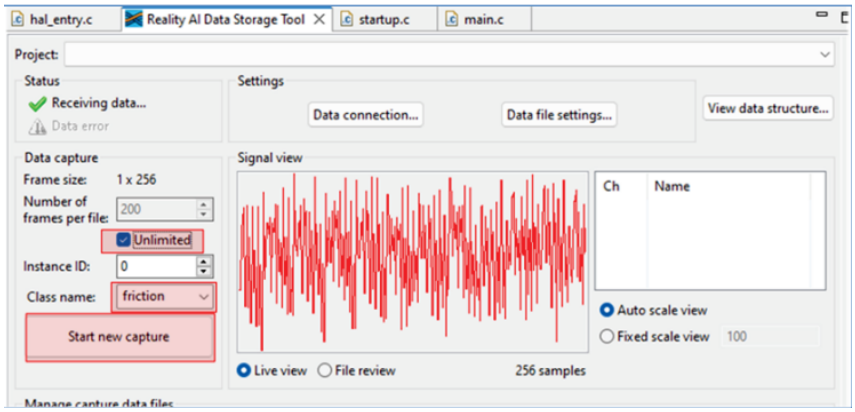


Şekil 2.12 SensiML etiketleme arayüzü

Üzerinde sensör bulunduran veya dışarıdan bağlanan sensörlerin hızlı bir şekilde okunmasının yapılabildiği arduino, raspberry pi, ST gibi büyük firmalara ait donanımları desteklemektedir. Platform üzerinde birden fazla veri seti ve uygulama kütüphane örnekleri bulunmaktadır [42].

- Reality AI

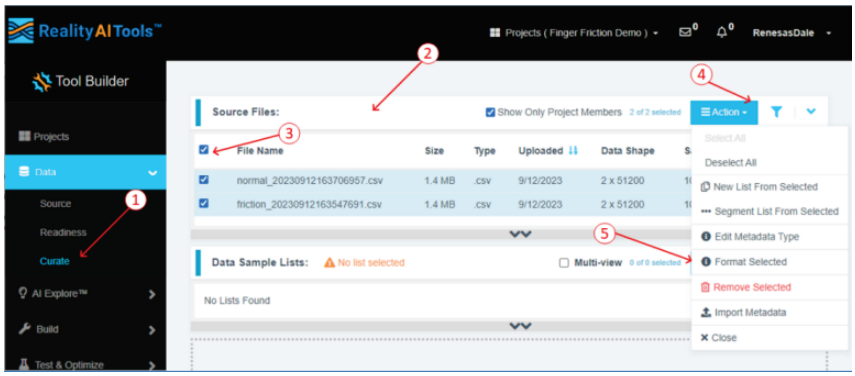
  Reality AI; Renesas firmasına ait gerçek zamanlı analitik, görüntü ve ses uygulaması geliştirmeyi amaçlayan kendi derleyicisi ve internet üzerinde bulunan bir portalda çalışabilen Tiny-ML platformudur [43].



Şekil 2.13 Reality AI veri toplanma

Derleyici içinde çalışan Şekil 2.13' te görüldüğü gibi AI uygulamasında kullanılması amacı ile veri toplama ve görüntüleme yapılabilmektedir. Bu özellik büyük avantaj sağlamaktadır.

Gerçek zamanlı veri toplayan derleyici, veriyi CSV veya WAVE formatında kaydedebilmektedir. Kaydedilen bu veriler Şekil 2.14 'de gösterildiği gibi model oluşturmak için portala aktarılabilmektedir.

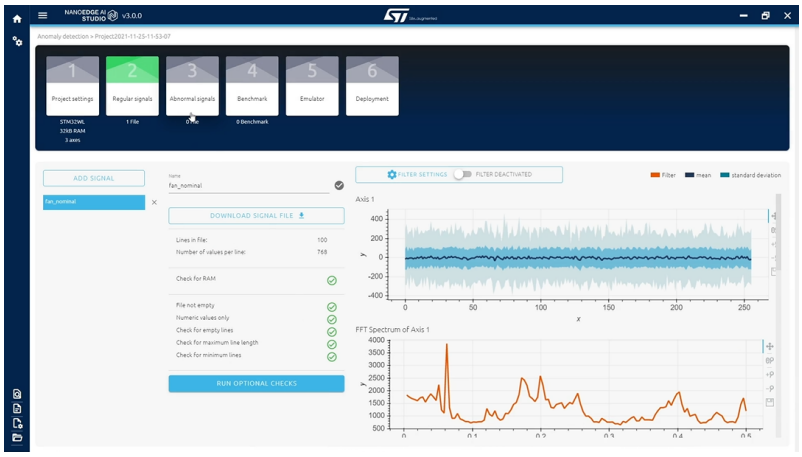


Şekil 2.14 Reality AI portal arayüzü

Platform sınıflandırma ve anormallikleri algılayan modelleri desteklemektedir. Seçilen modele göre etiketlenmiş veriler eğitildikten sonra model çıktısı alınabilmektedir. Model çıktısı, 8/32 bit ve C/C++ kütüphanesi olarak üretilebilmektedir. Üretilen model sadece renesas firmasına ait donanımlara uygulanabilmektedir. Ek olarak kullanıcılar için verilen birçok uygulama örneği ve dokümanı da bulunmaktadır [44].

- NanoEdge AI Studio

  ST firmasının ücretsiz olarak sunduğu diğer platformlar gibi veri toplamak, doğrulamak ve projenize entegre edilecek modeli C koduna çeviren bir Tiny-ML platformudur [45]. Tek bir arayüz üzerinden tüm işlemleri basite indirgeyerek sunmaktadır (Şekil 2.15). Anormallik tespiti, sınıflandırma veya regresyon modellerine destek vermektedir.



Şekil 2.15 NanoEdge AI Studio uygulama akışı ve arayüzü

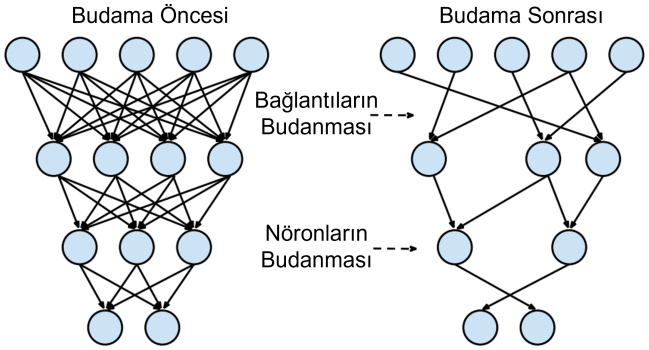
Platformun kullanımı ve detayların aktarıldığı bir doküman sayfası bulunmaktadır [46]. Platform tüm ST mikro denetleyicileri ve kitleri desteklemektedir. Platform içerisinde birçok uygulama tipi ve örnek yer almaktadır [47].

2.1.3.4 Sıkıştırma ve Optimizasyon

Tiny-ML uygulamalarındaki en büyük kısıtlar sınırlı hafıza ve hesaplama gücünün düşük oluşudur. Bu kısıtların aşılması için modelin sıkıştırılması ve optimize edilmesi şarttır. Bu işlemlerin yapılmadığı modellerde hız ve hafıza problemi görülmektedir. Bu nedenle, model boyutunu ve hesaplama karmaşıklığını azaltmak için en temel iki teknik budama ve nicelemedir.

- Budama

  Budama, model mimarisindeki nöronlar ve arasındaki önemsiz bağlantıların kaldırılması işlemidir (Şekil 2.16). Bu, bir modelin bellek ayak izini azaltmanın en doğrudan yöntemidir. Düğümler arasındaki etkileşimleri hesaplamak için daha az iş yapılması sayesinde gecikmeyi önlemek ve hesaplama yükünü azaltma gibi ek faydalara sahiptir. Bu nitelikler, budanmış modellerin hem konuşlandırılmasını hem de daha iyi sonuçlar için daha fazla eğitilmesini kolaylaştırır. Aynı zamanda doğruluğu da arttırmaktadır.

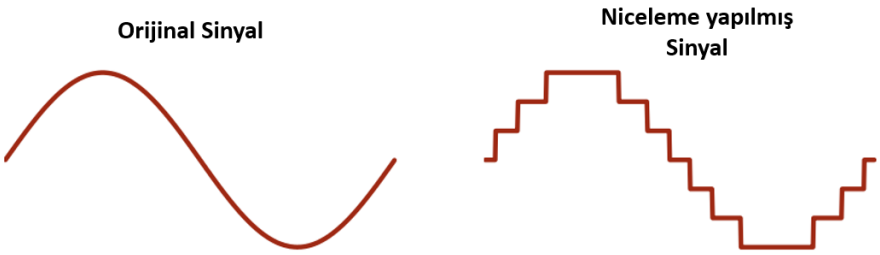


Şekil 2.16 Budama örneği

  Çoğu durumda önemli bir performans düşüşü meydana gelmeden önce bir modelin parametrelerinin yüzde 80 kadarı kaldırılabilir. Budama işlemi genellikle optimizasyon aşamasının bir parçası olarak bir modelin eğitilmesinden sonra gerçekleşir. Budama teknikleri karmaşıklık ve etkililik açısından farklılık gösterir. Standart bir yöntem, ağırlıkların her bir sinir ağı katmanında aşması gereken bir eşiği hesaplayan ve daha sonra bu eşiğin altındaki ağırlıkları budayan büyüklüğe dayalı budamadır. Tensoflow yazılım kütüphanesi ile yapılmış budama işlemiyle yakın doğruluklar elde edilerek 3 kat daha düşük model sonuçlarına ulaşılmıştır [48] [49].

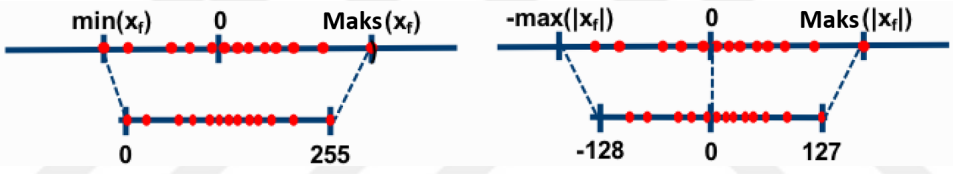
- Niceleme

  Niceleme, genellikle Tiny-ML modellerinde 32 bitlik ağırlık değerlerinin çözünürlüğünü azaltarak çalışır. Şekil 2.17 'de sembolik olarak gösterilmiştir.



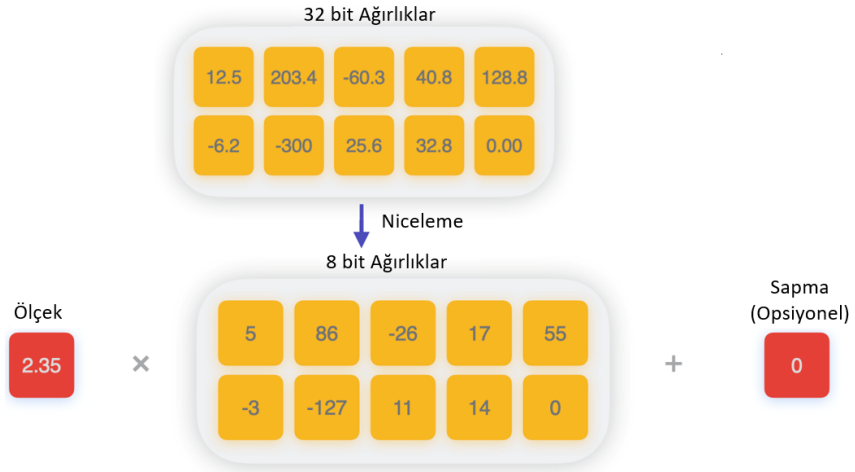
Şekil 2.17 Niceleme örneği

Yapılan bu işlem, daha küçük bir model boyutuna ve daha hızlı hesaplamaya neden olur. Tiny-ML platformlarında modelin optimizasyonu için hem eğitim aşamasında hem de eğitim sonrası modelin parametrelerine uygulanabilir. Bu şekilde model boyutu 4 kat ve gecikme süresi 1,5 - 4 kat arası iyileşmektedir [50]. Görüntü sınıflandırma araçları üzerinde yapılan modellerde yakın doğruluklar elde edilmiştir [51]. Birden fazla niceleme tipi bulunmaktadır. Bunlar; simetrik, simetrik olmayan, tek biçimli ve tek biçimli olmayan niceleme tipleridir (Şekil 2.18).



Şekil 2.18 Simetrik ve simetrik olmayan niceleme

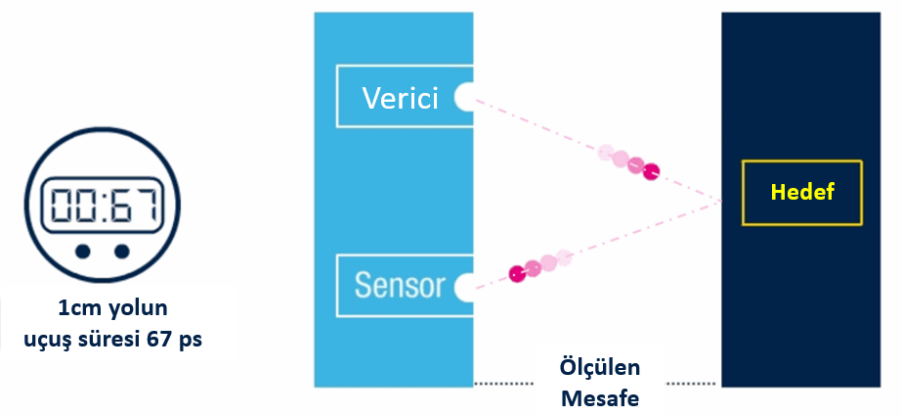
Model ağırlıkları üzerinde uygulanan niceleme işlemi örneği Şekil 2.19 'da gösterilmiştir. 32 bit tipine sahip ağırlıklar 4 kat sıkıştırılarak 8 bit tipinde veriye çevrilmiştir. Bu şekilde 4 kat sıkıştırma sağlanabilmektedir.



Şekil 2.19 Model ağırlıkları nicelemesi

2.2 UÇUŞ SÜRESİ SENSÖRÜ (ToF)

Belirli dalga boyuna sahip optik kaynak kullanılarak gönderilen foton hüzmesinin bir engele çarpması ve optik kaynağa tekrar ulaşması için geçen süreyi ölçebilen sensörlere uçuş süresi sensörleri denir. Hesaplanan süre ile engelin mesafesi tespit edilebilmektedir (Şekil 2.20). Farklı dalga boyuna sahip kaynaklar ile çalışabildiği gibi ses dalgalarını da kullanılabilir.



Şekil 2.20 Uçuş Süresi (ToF) ölçüm prensibi

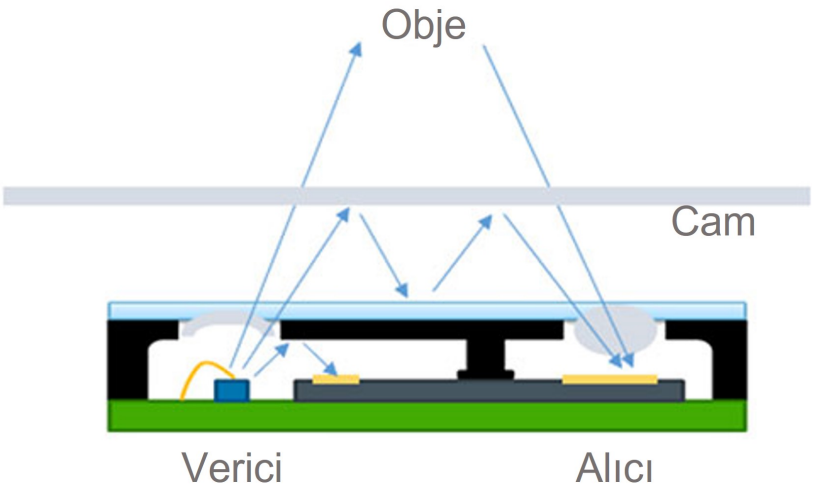
ToF sensörleri çeşitli uygulamalarda kullanılmaktadır. Özellikle mesafe ölçümü, hareket tespiti, 3D modelleme ve görüntü odaklama gibi alanlarda yaygın olarak kullanılırlar. ToF teknolojisi diğer mesafe ölçme yöntemlerine (ultrasonik sensörler, lazer mesafe ölçerler vb.) göre daha hızlı yanıt süresi, daha yüksek doğruluk ve daha iyi performans sağlayabilir.

ToF sensörleri; düşük güç tüketimi, yüksek hassasiyet, geniş ölçüm aralığı ve hızlı tepki süreleri gibi avantajlara sahiptir. Bu özellikler, ToF sensörlerinin birçok alanda kullanılmasını sağlamaktadır. Uygulama alanına göre ToF ölçüm altyapısına sahip birçok 3 boyutlu kamera, sensör modülleri ve entegreleri bulunmaktadır.

Bu tez çalışmasında düşük güçlü, düşük maliyetli, alıcısı ve vericisi bir entegre içerisinde yer alan ve işlem gücünü düşük tutmak için az piksele sahip sensörler incelenmiştir.

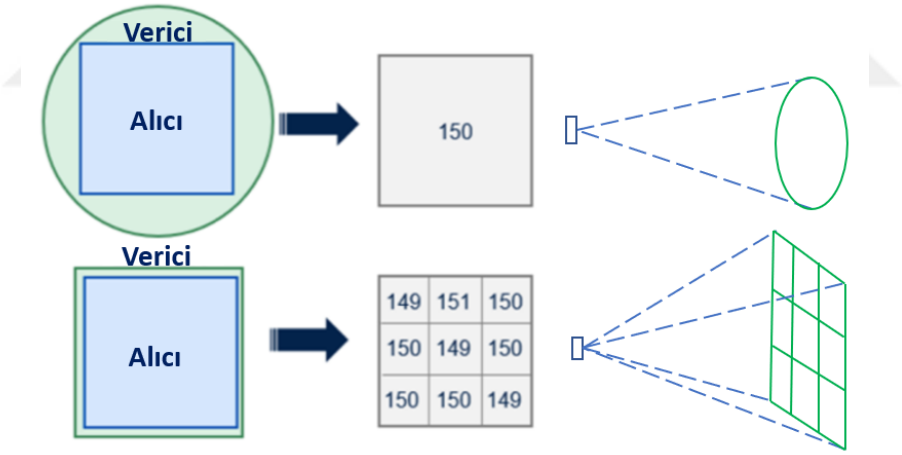
2.2.1 Çalışma Prensibi

Tek bir entegre içerisinde yer alan alıcı ve verici tipindeki ToF sensörlerin çalışma prensibi incelenmiştir. Şekil 2.21 'de gösterildiği gibi verici 940nm dalga boyuna sahip VCSEL (Dikey Kavite Yüzey Yayan Lazer) teknolojisini kullanmaktadır. Alıcı tarafında ise birçok foto-diyot içeren bir piksel dizisine sahip SPAD (Tek Fotonlu Çığ Diyotu) teknolojisi yer almaktadır.



Şekil 2.21 Uçuş Süresi (ToF) sensör alıcı ve vericisi

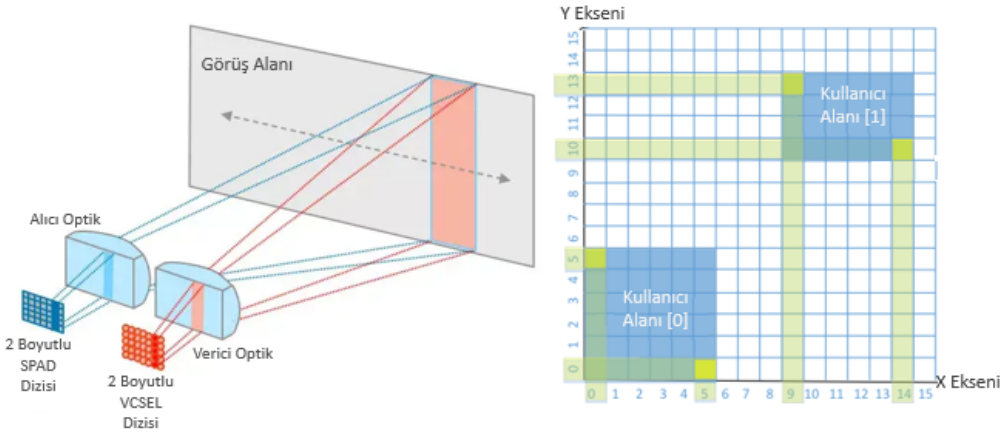
ToF sensörler, tek piksel veya çoklu piksel yapısına sahiptir. Çoklu piksel yapısı derinlik haritası oluşturabilmektedir (Şekil 2.22). Tekli yapılarda genellikle verici konik bir yapıya sahipken çoklu piksel yapısına sahip olanlarda ise üçgen prizma şeklindedir.



Şekil 2.22 Tekli ve çoklu piksel gösterimi

Çoklu piksel yapısı için Şekil 2.23 'de gösterildiği gibi SPAD ve VCSEL yapısı senkron ölçüm alarak tarama gerçekleştirmektedir. Bu teknoloji üreticiye göre değişiklik göstermektedir.

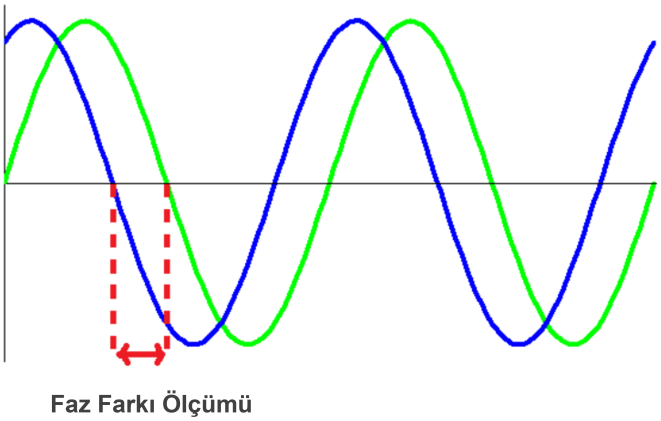
Sensörlerin kontrol arayüzlerinde ölçüm süreleri ve piksel konfigürasyonları gerçekleştirilerek uygulamaya adapte edilebilmektedir. Ayrıca bazı sensörlerde görüş açısı (FoV) da programlanabilmektedir (Şekil 2.23). Böylece sensörün algılama mesafesini ve ölçüm kalitesi arttırılabilmektedir [52].



Şekil 2.23 Senkron ölçüm ve programlanabilir görüş açısı (FoV)

ToF sensörler çalışma prensibi olarak ışık kaynağının sürüş şekline göre doğrudan (dToF) ve dolaylı (iToF) olarak ikiye ayrılmaktadır.

Dolaylı uçuş süresi (iToF) yöntemi, sürekli dalga genliği modüle edilmiş ışığın ölçülecek nesneye gönderilmesi ile yansıyan dalganın arasındaki fazın ölçülmesi yöntemine dayanır (Şekil 2.24). Bu yönteme sürekli dalga (CW) modülasyonu, faz ölçümü veya genlik modülasyonlu sürekli dalga (AMCW) ismi de verilmektedir [53].



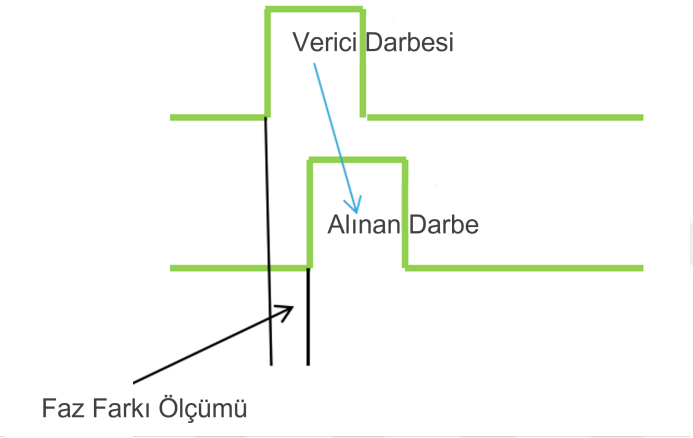
Şekil 2.24 Dolaylı uçuş süresi (iToF) sinyali

Dolaylı ToF yönteminde gönderilen sinyal ile yansıyan sinyal arasındaki faz farkı 2.1 'de gösterilen formülle hesaplanır. Formülde d mesafe, c ışığın boşluktaki hızı ($3 * 10^8 m/s$), f_m modülasyon frekansı ve $\Delta\Phi$ ise faz kaymasını ifade etmektedir.

$$d = c * \frac{\Delta\Phi}{4 * \pi * f_m} \tag{2.1}$$

Yaygın olarak kullanılan doğrudan uçuş süresi (dToF) yönteminde, vericiden çok kısa süreli bir darbe sinyali gönderilir ve engelden yansıyarak geri gelen sinyal ile gönderilen sinyal arasındaki süre ölçülür (Şekil 2.25). Bu süreye bağlı olarak mesafe aşağıda yer alan 2.2 'deki denklem ile hesaplanır [53].

$$d = t * \frac{c}{2} \tag{2.2}$$



Şekil 2.25 Doğrudan uçuş süresi (dToF) sinyali

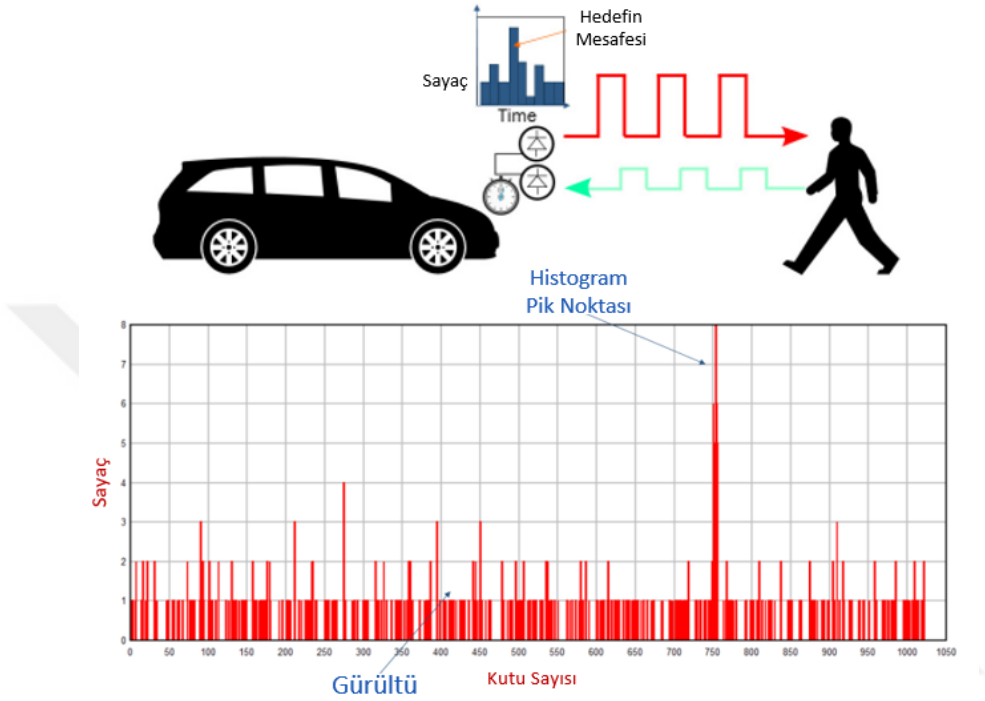
Bu yöntemin avantajları arasında yüksek enerjili darbelerin kullanılmasıyla güneşin bozucu etkisinin az oluşu gösterilebilir. Yoğun ışık darbeleri, göz güvenlik riski oluşturduğundan gözün korunması için çok hassas dedektörlerin kullanılması gerekmektedir. Güneş etkisinin az olması, algıladığı mesafe aralığını arttırmaktadır. Bu nedenle dToF yöntemi üreticiler tarafından sıkça kullanılır [53].

İki sürüş yönteminin karşılaştırması Tablo 2.3 'te verilmiştir [54]. iToF, daha kısa menzilli derinlik algılama uygulamaları ve sensöre doğrudan güneş ışığının gelmediği kapalı ortamlarda kullanıma uygundur. dToF hem kısa hem de uzun menzilli derinlik algılama uygulamaları için uygundur. Daha hızlı sinyal toplama oranları ve birden fazla yankıyı ölçme yeteneği sunarak dönüş yolunda çoklu nesnelerin algılanmasına olanak tanır.

Tablo 2.3 iToF ve dToF karşılaştırması

Parameter	**iToF**	**dToF**
Toplama Hızı	Uzun Entegrasyon Süresi	Hızlı Toplama
Uzaklık Belirsizliği	Evet	Hayır
Çoklu Yankıları Algılama	Hayır	Evet
Piksel Sayısı	Büyük	Küçük
Veri Hacmi	Küçük	Büyük
Güçlü Ortam Işığında Çalıştırma	İyi	Çok iyi

Doğrudan uçuş sensörü (dToF) yönteminde sıklıkla kullanılan çoklu darbe modu bulunmaktadır. Bu modda, bir seri darbe gönderilir ve algılanan dönüş darbelerinin zaman dağılımları incelenir. Şekil 2.26 'da darbelerin zaman dağılımları histogram grafiği olarak gösterilmiştir. Histogramda ölçülen hedefin mesafesine göre pik noktası oluşmaktadır. Bu şekilde ölçümün derinliği arttırılır ve sinyal-gürültü oranını (SNR) iyileştirilir.



Şekil 2.26 Dolaylı uçuş süresi histogram modu

2.2.2 Kullanım Alanları

ToF sensörleri, geniş bir kullanım alanına sahiptir ve birçok farklı uygulama için önemli bir rol oynamaktadır. Uygulamaya göre ToF sensör tipleri değişiklik gösterebilmektedir. Bu değişiklikler çözünürlük, piksel sayısı ve görüş açısı olarak sıralanabilir . ToF sensörlerin bazı kullanım alanlarının aşağıdaki gibidir.

- Robotik

  ToF sensörleri; robotik sistemlerde nesneleri algılamak, kişi takibi, duvar takibi, çevreyi haritalandırmak ve uçurum algılama gibi güvenlik koşulları için de kullanılır. Bu sensörler mobil robotlar, insansız hava araçları ve otonom araçlar gibi uygulamalarda önemli rol oynar.

- Mobil Cihazlar

  Telefonlar, tabletler ve akıllı saatler gibi mobil cihazlarda yüksek çözünürlüklü ToF sensörleri genellikle kamera sistemleriyle entegre edilir.

Bu sensörler derinlik algılama, portre fotoğrafçılığı ve yüz tanıma gibi özellikleri desteklemek için kullanılır.

- Dayanıklı tüketim aletleri

  İçecek hazırlama makinelerinde sıvı yüksekliği, bardak şekli ve derinliğini algılamak için kullanılır. Ek olarak kamera uygulaması içeren beyaz eşyalarda derinlik bilgisini algılamak için de kullanılmaktadır.

- Otomotiv

  Otomotiv endüstrisinde ToF sensörleri, araç içi algılama sistemlerinde ve sürücü destek sistemlerinde (ADAS) kullanılır. Bu sensörler araç içi ortamı izlemek, yolcuları ve sürücüyü algılamak, çarpışma önleme ve park yardımı gibi özellikleri desteklemek için kullanılabilir.

- Endüstriyel Otomasyon

  Endüstriyel robotlar ve otomasyon sistemlerinde ToF sensörleri nesnelerin konumunu ve mesafesini belirlemek için kullanılır. Bu sensörler robotların nesneleri algılaması ve hedeflere doğru hareket etmesi için önem taşır.

- Görüntü Odaklama

  Fotoğraf makinelerinde ve kamera sistemlerinde ToF sensörleri hızlı ve doğru odaklama için kullanılır. Bu sensörler nesnelerin uzaklığını algılar ve kameranın odaklama mekanizmasını buna göre ayarlar.

- Akıllı Ev ve Güvenlik

  Akıllı ev cihazlarında ToF sensörleri genellikle hareket algılama, mesafe ölçümü ve insanların konumunu belirleme gibi görevlerde kullanılır. Ayrıca güvenlik sistemlerinde insan algılama, kişi sayısını ve hareketleri algılama için de kullanılabilirler.

- Sanal ve Artırılmış Gerçeklik

  Sanal ve artırılmış gerçeklik uygulamalarında ToF sensörleri genellikle kullanıcıların hareketlerini ve çevrelerini algılamak için kullanılır. Bu sensörler kullanıcıların etkileşimini artırmak ve daha doğal deneyimler sağlamak için önemlidir.

2.2.3 ToF Entegre Sensör Üreticileri

Tek bir entegre içerisinde alıcı ve vericisini içeren, çoklu piksel yapısına sahip ve elektronik kart üzerine dizilerek bir yazılımla birlikte okunabilen sensör entegreleri

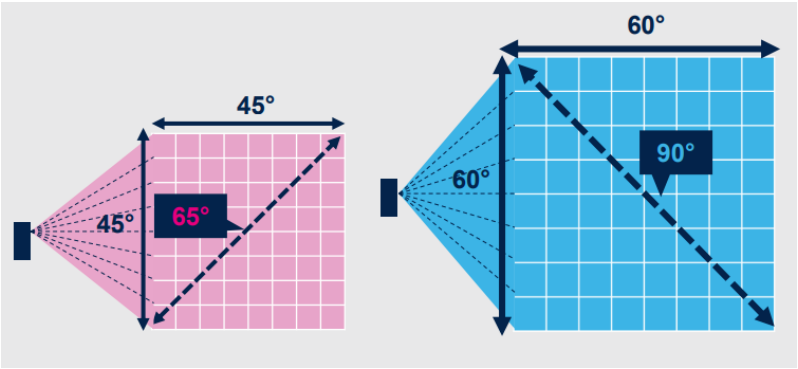
incelenmiştir. ToF sensör üreticilerden biri olan ST firması, farklı uygulamaları hedef almak amacıyla ürün çeşitliliğini zenginleştirmiş ve üreticiler arasında popüler bir yere ulaşmıştır. Çoklu piksel yapısına sahip olan sensörler Şekil 2.27 'de verilmiştir. Ek olarak, kullanıcılar için birçok sensör geliştirme kiti de bulunmaktadır.

	VL53L1CB Uzak Mesafe ve Çoklu Hedef Sensörü	VL53L5CX Geniş Görüş Açılı Çok bölgeli Mesafe Sensörü	VL53L7CX 90° Görüş Açılı Çok bölgeli Mesafe Sensörü	VL53L8CX Dayanıklı, Düşük Güç, Performansı geliştirilmiş Çok bölgeli Mesafe Sensörü	VL53L8UX 256 bölgeli Mesafe Sensörü
Çözünürlük **Maks Piksel Mod**	-	8x8 & 4x4	8x8 & 4x4	8x8 & 4x4	16x16 & 8x8
Maksimum Ölçüm Mesafesi	250cm	400cm	350cm	400cm	290cm
Görüş Açısı	27°	45° x 45° (Ø65°)	60° x 60° (Ø90°)	45° x 45° (Ø65°)	45° x 45° (Ø65°)
Yoğun Işık altındaki Performansı		+++ (170cm)	+ (65cm)	+++++ (285cm)	++ (100cm)
Yakın Mesafe Hassasiyeti	+ >2.5cm	++ >2.0cm	++ >2.0cm	+ >2.5cm	+ >2.5cm
Minimum Ölçüm Mesafesi	1.0cm	1.0cm	1.0cm	tbc	tbc
Histogram	Evet	Evet	Evet	Evet	Evet
Gürültü \ Leke bağışıklığı	80 cm	60 cm	60 cm	60 cm	60 cm
Güç Tüketimi Ölçüm	16mA	216mW	216mW	215mW	
Güç Tüketimi Otonom Mod	-	2.8mW / 8mW	2.8mW / 8mW	1.6mW / 7mW	
Hafıza	ROM 47.6kB RAM 9.8kB	ROM 6kB RAM 89kB	ROM 6kB RAM 89kB	ROM 6kB RAM 89kB	
Ürün Çeşitliliği	-	-	VL53L7CH Yapay Zeka Çözümü	VL53L8CH Yapay Zeka Çözümü	-
Boyutlar	4.9 x 2.5 x 1.56 mm	6.4 x 3.0 x 1.5 mm	6.4 x 3.0 x 1.6 mm	6.4 x 3.0 x 1.75 mm	6.4 x 3.0 x 1.75 mm

Şekil 2.27 ST çok bölgeli ToF sensörleri

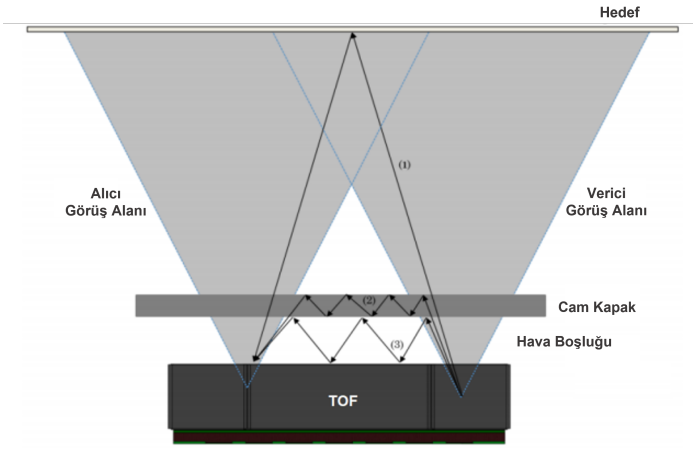
- VL53L5-L7-L8

  Bu sensörler 8x8 veya 4x4 alanı tarayacak şekilde programlanabilir. Farklı görüş açılarına, farklı ortam ışığı performanslarına ve farklı ölçüm mesafelerine sahiplerdir. Yapılacak uygulamaya göre sensör seçimine olanak sağlamaktadır. Örneğin; L5 ve L8 sensörleri çapraz görüş açısı ϕ 65°, L7 sensörün ϕ 90° dir (Şekil 2.28).



Şekil 2.28 L5/L8 ve L7 sensör görüş açıları

Sensörlerin çevresel etkilerden korunması için önüne kızılötesi geçirgen bir malzeme yerleştirilmektedir. Bu malzeme; toz, su , buhar gibi sensörü bozabilecek etkileri sıfırlar. Ancak önüne yerleştirilen bu malzeme sensörün ölçümünü etkileyebilmektedir.

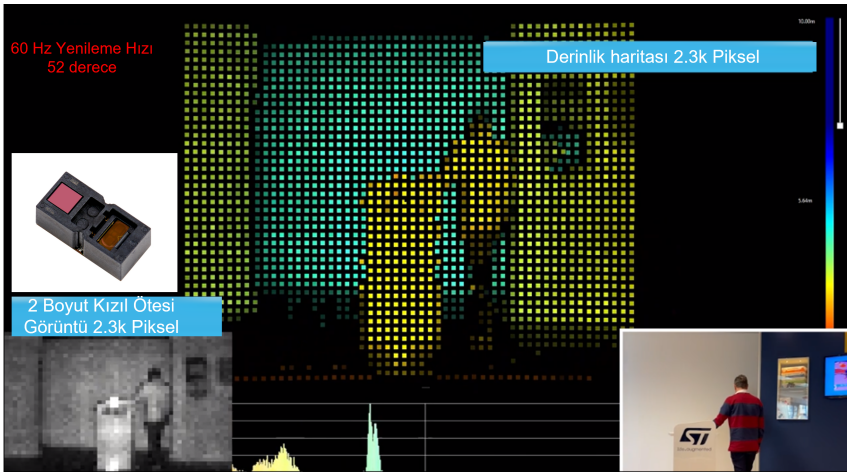


Şekil 2.29 Koruyucu malzeme (Cam kapak) ve sinyal gürültüsü

Şekil 2.29 'da görüldüğü gibi sensör - malzeme arasındaki hava boşluğu, malzemenin kalınlığı ve malzemenin asimetrik olması durumları ölçüme gürültü eklemektedir. Gürültü dalgaları, vericiden çıkan foton dalgalarının koruyucu malzeme içerisinden ve yüzeyinden alıcıya ulaşır. Bu durum gürültü kalibrasyonu yapılarak minimuma seviyeye indirilebilir. Kalibrasyon genellikle ışık seviyesi az olan, sabit mesafede ve koyu renk bir hedefte yapılmaktadır. Bu tarz sensörlerin kalibrasyon fonksiyonları mevcuttur. Yazılımda kalibrasyon fonksiyonu çağrıldıktan sonra otomatik bir şekilde sensör kendini kalibre etmektedir.

- VL53L9CA

  ST firmasının son ürünü olan bu sensör, 54 x 42 çözünürlüğünde ve 10 metreye kadar ölçüm sağlayabilen iToF teknolojisine sahip 3 boyutlu bir ToF sensörüdür. Şekil 2.30 'da görüldüğü gibi 2 boyut ve 3 boyutlu olarak ölçüm alabilmektedir [55].



Şekil 2.30 VL53L9CA

Diğer popüler üreticilerden biri de AMS-OSRAM firmasıdır. Bu üreticinin çok bölgeli ToF sensörü sayısı çeşitliliği 3 ürün ile sınırlıdır. Ancak görüş açısı ve ölçüm modunu programlanarak uygulama için hızlı bir şekilde adapte edilebilmektedir. Ek olarak, güneş altındaki ölçüm performansı ST ürünlerine oranla daha iyidir [56].

- TMF8820

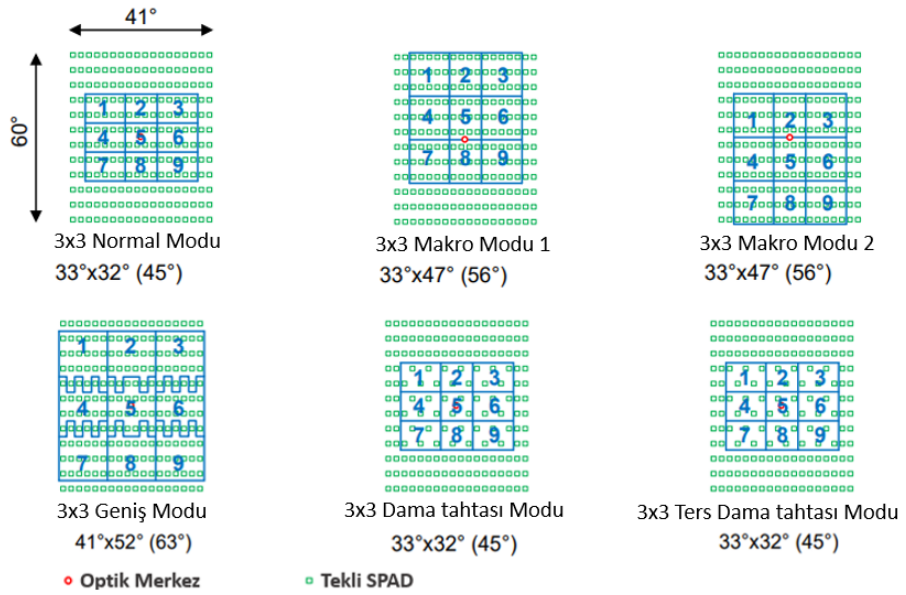
  Bu sensör AMS-OSRAM firmasının en temel modelidir. 3x3 piksel boyutunda ölçüm almaktadır. Mesafe verisinin dışında histogram ve yansıma değerlerini de vermektedir.

- TMF8821

  3x3, 4x4 ve 3x6 piksel boyutunda programlanabilen esnek bir modeldir. Temel modeldeki gibi mesafe verisinin dışında histogram ve yansıma değerlerini de içermektedir.

- TMF8828

  3x3, 4x4, 3x6, 8x8 piksel boyutlarında programlanabilen en üst modeldir. Bu ürün de mesafe verisinin dışında histogram ve yansıma değerlerini de verir. Tüm ürünlerinde SPAD programlanarak bir maske yaratılmakta ve görüş açısı değiştirilebilmektedir (Şekil 2.31). Ayrıca 40 cm mesafeden kalibrasyon yapılarak koruyucu malzemenin gürültü etkisi azaltılabilir.



Şekil 2.31 TMF8820/21/22 SPAD bölge konfigürasyonu ve modları

2.3 VL53L7CX ToF Sensör ve Veri Seti

Bu tezde kullanılan yöntem, otonom robotun hareketi sırasında önüne veya hareket yoluna çıkan engellerin algılanmasının sağlanmasıdır. Bu nedenle hareket halinde iken sensörün taradığı alan ve toplanan verilerin önemi büyüktür. Ayrıca bu tez çalışmasında çapraz görüş açısı ϕ 90° ve 8x8 çoklu piksel yapısına sahip Şekil 2.32 'de gösterilen VL53L7CX sensörü kullanılmıştır [57].



Şekil 2.32 VL53L7CX

VL53L7CX ToF sensöründen okunan her piksel değerinin birden çok ölçüm çıktısı bulunmaktadır. Bu ölçüm çıktıları Tablo 2.4 'de verilmiştir. Tabloda bulunan ölçüm çıktıları 64 (8x8) piksel için vererek tek bir ölçümde 384 değer sensör üzerinden alınabilmektedir. Sensörün ölçüm frekansı maksimum 15 Hz olarak belirlenmiştir. Sonuçta sensör 1 saniyede 5760 (384x15) değer üretebilmektedir.

Tablo 2.4 VL53L7CX sensör çıktıları

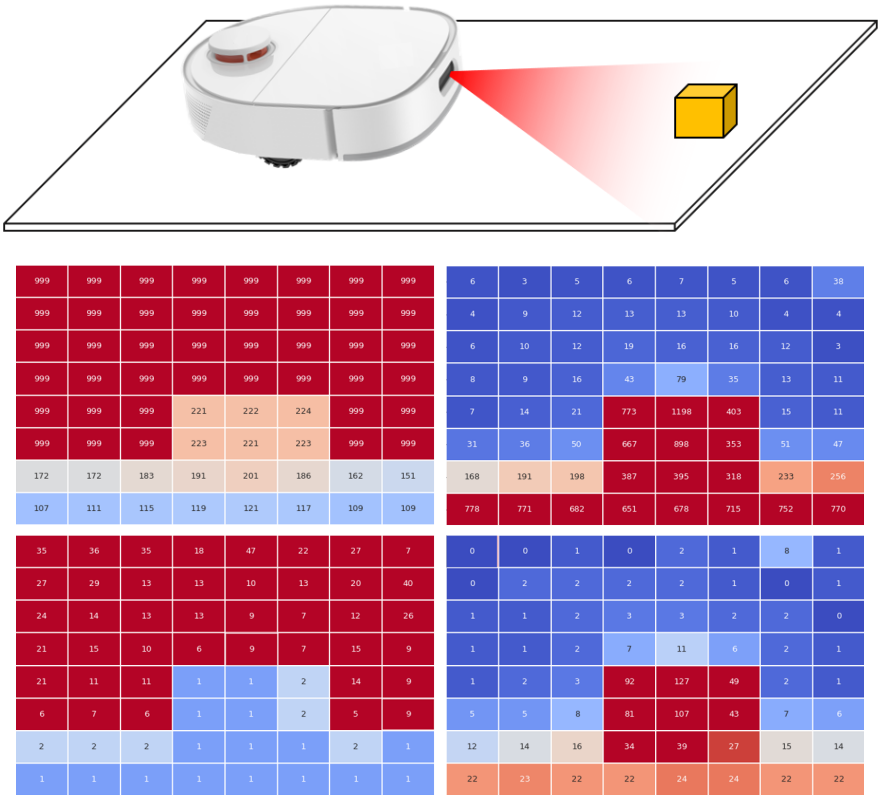
Ölçüm	**Birim**	**Açıklama**
Signal seviyesi	Kcps/SPAD	VCSEL sırasında ölçülen foton darbesi miktarı
Mesafe	Millimetre	Hedef Mesafe
Mesafe değişimi	Millimetre	Hedef için mesafe değişimi
Yansıma	Yüzde	Hedefin yansıma oranı
Ortam aydınlık seviyesi	Kcps/SPAD	SPAD aktif foton emisyonu olmadan ölçülen ortam sinyal oranı
Ölçüm durumu	-	Ölçüm geçerliliği, her piksel için hata durumu

Bu tez çalışmasında otonom cihaz olarak ev tipi robot süpürge kullanılmıştır. Kullanılan bu robotta sensör konumu Şekil 2.33 'te gösterildiği gibi ön tarafta ve yerden 5 cm yükseklikte bulunmaktadır.



Şekil 2.33 Örnek robot üzerinde ToF sensörün konumu

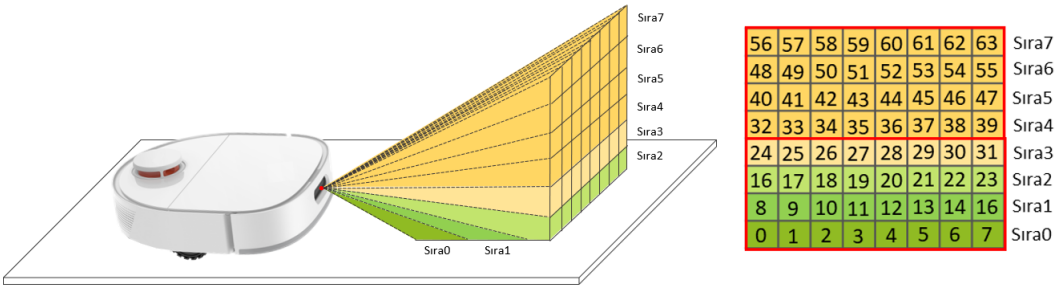
Robot üzerine yerleştirilen ToF sensörüyle küp bir engelin ölçümü alınmıştır. Alınan mesafe, sinyal seviyesi, mesafe değişimi ve yansıma verileri Şekil 2.34 'te gösterilmiştir.



Şekil 2.34 8x8 Mesafe, sinyal seviyesi, mesafe değişimi ve yansıma verileri

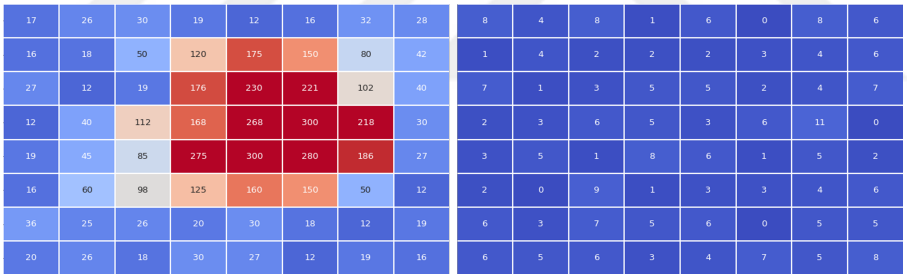
Şekil 2.34 'teki veriler incelendiğinde nesne formunun seçilebildiği görülmektedir. Nesne yani engel küçüldükçe veya uzaklaşıldıkça formu, 1 piksele kadar düşmekte ve kaybolmaktadır. Robot süpürge üzerine konumlandırılan sensörün görüş açısının

geniş olması nedeniyle zemini de görmektedir (Şekil 2.34, Şekil 2.35). Şekillerde görüldüğü üzere alttan 4 sıra piksel serisi robotun önünü ve zemini görmektedir. Yukarı tarafta kalan 4 sıra ise robotun üst tarafını görmektedir. Ev robotunun hareketini etkileyecek engeller zeminde bulunmaktadır. Bu nedenle 64 piksel verinin üst yarısı veri kümesine dahil edilmemiştir. Böylece veri sayısı yarıya indirilerek hafıza kullanımı azaltılmıştır.



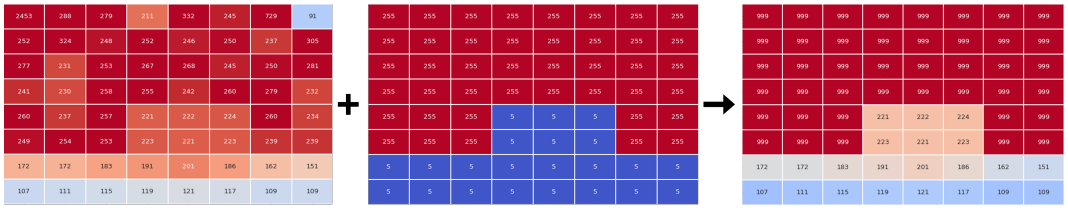
Şekil 2.35 ToF görüş açısı ve piksellerin konumu

Kullanılan ToF sensör, kızılötesi dalga boyuna sahip darbe sinyalleri ile çalışmaktadır ve güneşin vurduğu ortamda sensör verileri gürültülü olmaktadır. Bu nedenle ortamın aydınlık seviye verisi önemlidir. Sensör, güneş alan bir pencereye doğru tutulduğundaki ve karalık ortamda alınan veriler Şekil 2.36 'da gösterilmiştir.



Şekil 2.36 Aydınlık ve karanlık koşulda ortam aydınlık verisi

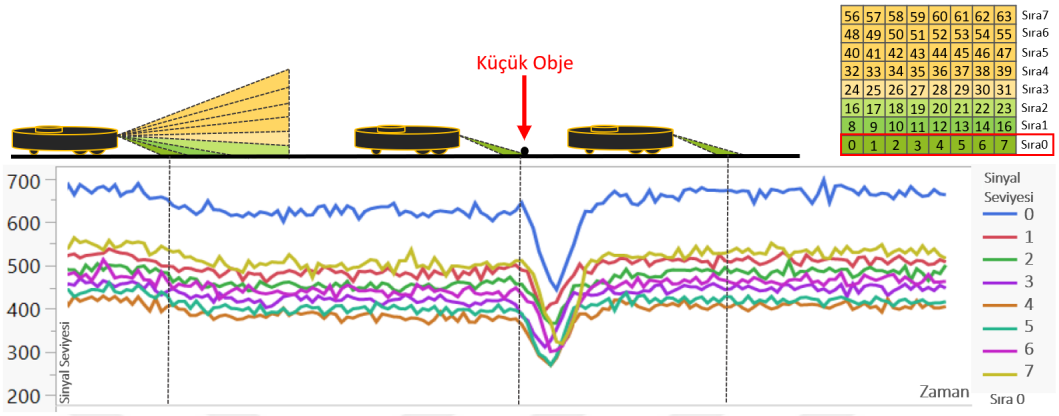
Her piksel verisinin kalitesine karşılık gelen ölçüm durumu verisine bakılarak alınan diğer veriler filtre edilebilmektedir. Örneğin Şekil 2.37 'de alınan mesafe verisi ve ölçüm durumu verisi kullanılarak final nesnenin yani hedefin mesafe verileri elde edilebilmektedir.



Şekil 2.37 Ölçüm durumu bilgisi ile mesafe filtre örneği

Küçük nesnelerin, zemine yakın olmaları ve mesafenin de uzak olması gibi durumlarda algılama çok zorlaşmaktadır. Bu nedenle algılama yöntemi olarak

robotun engele doğru hareketi sırasında alınan bir seri veri değerlerinin değişine bakılmaktadır. Bu değişimler; yansıma, sinyal seviyesi, mesafe ve mesafe değişimi verilerinde de görülmektedir. Şekil 2.38'de siyah obje veya nesnenin sinyal seviyesinin ilk sıra piksel değerleri görülmektedir. Bu değişimi yakalamak için sensörden alınan art arda 5 veri kümesi kullanılmaktadır.



Şekil 2.38 Hareket halindeki robot üzerinden alınan sinyal seviyesi değeri

Belirlenen yöntem için kullanılacak veri kümesinde sinyal seviyesi, mesafe, mesafe değişimi ve yansıma verileri model girişi olarak belirlenmiştir. Diğer veriler ise filtreleme için kullanılacaktır. Tüm veri tipleri aynı anda değerlendirildiğinde donanımın belleği yetmeyeceğinden veriler ayrı ayrı değerlendirilecektir. Ancak test yükünün azalması için tüm veri tipleri aynı anda toplanacaktır (Tablo 2.5). Bir veri paketi toplamda 192 değişkenden oluşmaktadır.

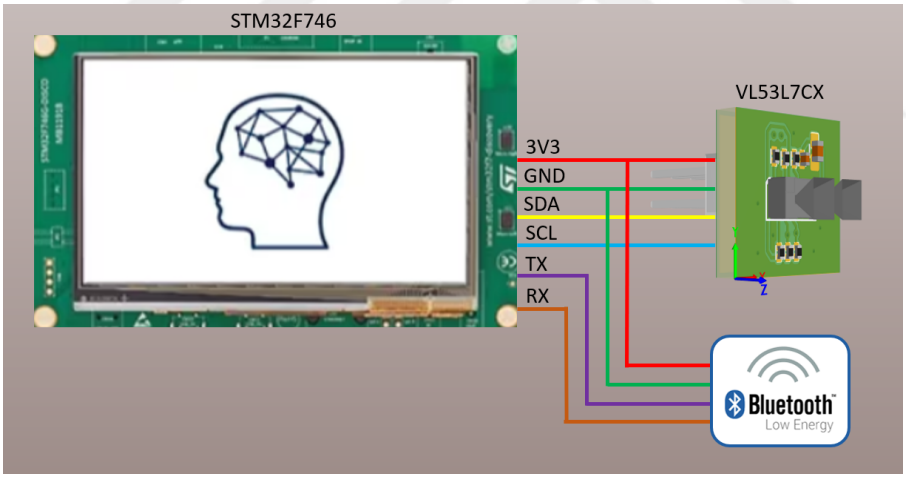
Tablo 2.5 Yöntem için kullanılan sensör veri kümesi

Sensör verisi	**Boyut**	**Kullanım amacı**
Signal seviyesi	5(8x4)	ML Model giriş verisi
Mesafe	5(8x4)	ML Model giriş verisi
Mesafe değişimi	5(8x4)	ML Model giriş verisi
Yansıma	5(8x4)	ML Model giriş verisi
Ortam aydınlık seviyesi	5(8x4)	Verinin filtrelemesi
Ölçüm durumu	5(8x4)	Verinin filtrelemesi

3
UYGULAMA

3.1 Kullanılan Donanım

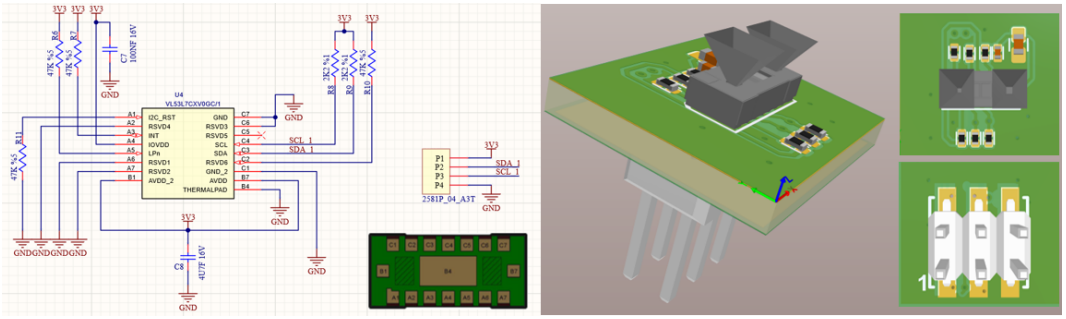
Bu tez çalışmasında ST firmasına ait STM32F746 geliştirme kartı kullanılmıştır (Şekil 3.1). Bu kart üzerinde ARM-Cortex-M7 mimarisine sahip mikro denetleyici bulunmaktadır. Sağladığı 200 MHz çalışma frekansı ve 1 MB hafızası ile Tiny-ML uygulamalarında kullanımı yaygındır. Ayrıca kit üzerinde 4.3" RGB 480×272 dokunmatik ekran , kamera soketi, mikro SD kart yuvası ve arduino kiti bağlanabilen soket yapısına sahiptir [58].



Şekil 3.1 Kullanılan donanımlar

Kullanılan VL53L7CX sensör entegresi SMD tipinde yani pinleri pad şeklinde alttan lehimlenmektedir. Sensörlerde sıklıkla kullanılan I2C yani dijital bir haberleşmeye sahiptir [59]. Robot üzerinde sensörün uygun bir şekilde yerleştirilebilmesi amacıyla sensör kartı tasarımı yapılmıştır. Bu kart, mikro denetleyici karta uygun olması için 3,3V 'luk bir tasarıma sahiptir (Şekil 3.2).

Ek olarak sensör üzerinden toplanan verilerin uzak kullanıcıya aktarılması için UART haberleşme protokolüne sahip BLE modülü kullanılmıştır (Şekil 3.1).

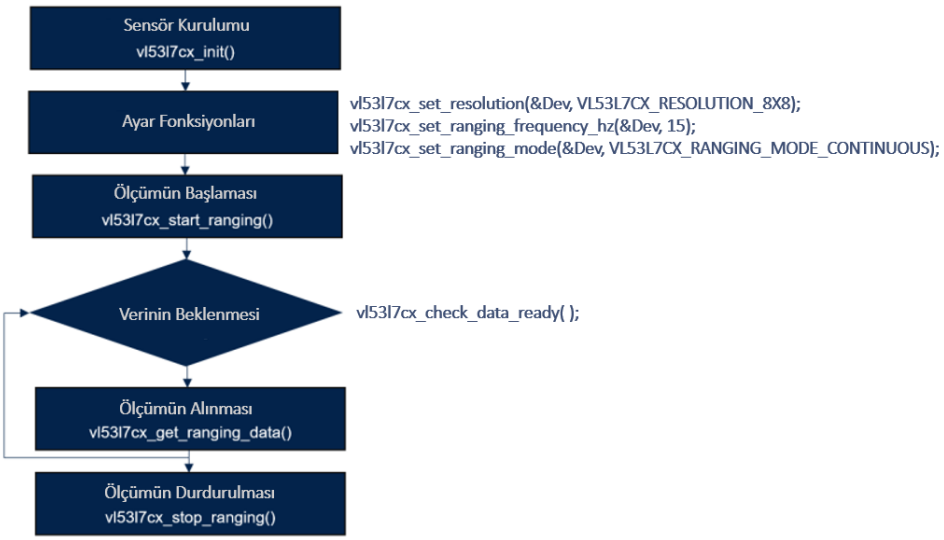


Şekil 3.2 VL53L7CX sensör devre şematiği ve PCB tasarımı

3.2 Kullanılan Yazılımlar

Bu tezde kullanılan sensör için ST firmasının yazılım kütüphanesi bulunmaktadır [60]. Bu kütüphane içerisinde sensörün kurulum ve sürüşünü içeren birçok fonksiyon yer almaktadır [61]. Ayrıca bu kütüphane içerisinde I2C fonksiyonları ayrı bir şekilde verilmiştir. Böylece farklı mikro denetleyici platformlarına sahip kullanıcılar da hızlıca yazılım adaptasyonu gerçekleştirebilir.

Bu tez çalışmasında, 8x8 piksel ve 15 Hz frekansıyla sürekli okuma olacak şekilde sensör kurulumu yapılmıştır. Şekil 3.3 'te sensörün kurulumu, okuma akışı ve kullanılan fonksiyonlar verilmiştir.



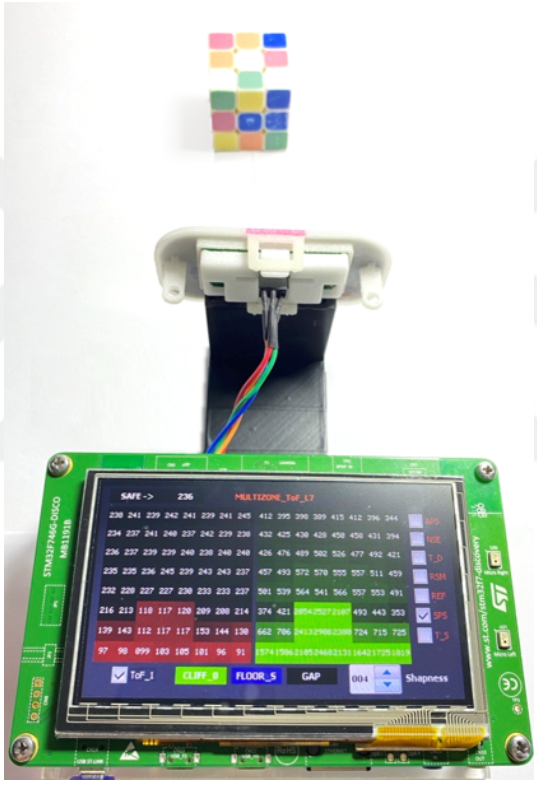
Şekil 3.3 VL53L7CX sensör akış diyagramı

Çalışmada kullanılan bir başka yazılım ise geliştirime kartı içindir. Bu kart üzerinde STM32F746NGH7 kodlu mikro denetleyici yer almaktadır. Bu yazılımın ihtiyacı olan I2C, UART ve üzerinde bulunan TFT ekranını sürmek için gerekli pin tanımlamaları ve uygulama örnekleri ST firmasının paylaştığı açık kaynak sayfası üzerinden alınmıştır [62]. Ek olarak ST programlama arayüzününde de programın temel hali üretilebilmektedir [63].

Mikro denetleyici kit üzerinde C ve C++ dili kullanılmıştır. Bu yazılım dilinin derlenmesi ve hata ayıklamaları için IAR firmasına ait EWARM derleyici kullanılmıştır [64]. Ayrıca TFT ekran üzerinde verinin görüntülenmesini sağlayan emWin kütüphanesinden yararlanılmıştır [65].

Kart üzerinden veri toplamak ve Tiny-ML modelinin çalıştırılması için iki gömülü sistem yazılımı gerçekleştirilmiştir.

Birincisi, sensörü okuyan TFT ekran üzerinde verinin gösterilmesini sağlayan ve veri toplanması için UART üzerinden log atabilen bir yazılımdır. Bu yazılımın uygulanmış görsel hali Şekil 3.4 'te gösterilmiştir.



Şekil 3.4 Veri toplama ve anlık görüntüleme yazılımı

Bu yazılım ile haberleşme modülü üzerinden gönderilen veri, sensör ölçüm frekansı olan 15 Hz 'de 8x8 boyutunda her piksel için sinyal seviyesi, mesafe, mesafe değişimi, yansıma, ortam aydınlık seviyesi ve ölçüm durumunu içermektedir. Alınan veriler bilgisayar üzerinde CSV formatında kaydedilmektedir.

İkinci yazılım ise Tiny-ML platformundan çıkartılan makine öğrenmesi modelini çalıştıran bir yazılımdır. Bu yazılımda modele giriş yapılan sensör verilerinin toplanması önemlidir. Belirlenen yöntem olarak art arda 5 veri alınmaktadır. 5 veri kümesini içeren dizi doldurulduktan sonra sınıflandırma modeli çağrılır ve sonucu alınır.

Yeni bir veri alındığında 5 veri kümesini içeren diziden ilk veri silinip diğer veriler kaydırılarak 5. veri olarak yeni veri eklenir ve tekrar model çağrılır. Özetle modelin çağrılma frekansı sensörün ölçüm frekansı(15 Hz) ile aynı olmaktadır.

Tiny-ML platformu üzerinden alınan model kütüphanesinin derleyici içerisine eklenmesi için örnek projeler ve yönlendirme dokümantasyonları bulunmaktadır [66][67]. Model kütüphanesi eklendikten sonra "run classifier()" fonksiyonu çağırılıp model çalıştırılabilmektedir.

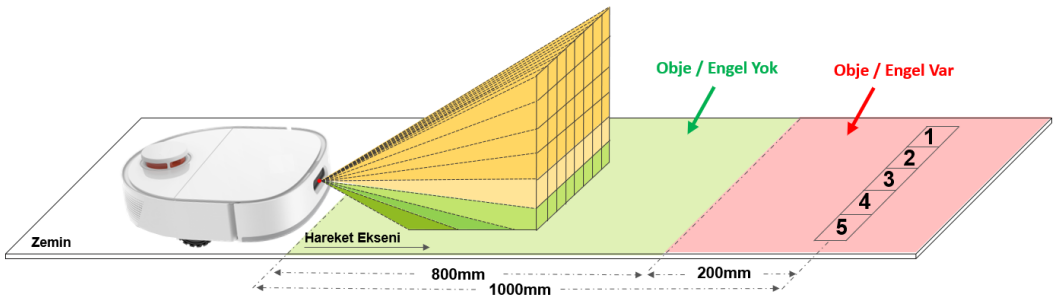
3.3 Veri Setinin Toplanması ve Etiketlenmesi

Ev tipi temizlik robotlarının ev içerisinde karşılaşabileceği küçük engel oluşturabilecek objeler Şekil 3.5 'te gösterildiği gibi seçilmiştir. Bu objeler farklı büyüklüklerde, farklı renk ve farklı malzeme tiplerinden oluşmaktadır.



Şekil 3.5 Veri toplama için kullanılan objeler

Hazırlanan deney düzeneğinde robotun karşısında olacak şekilde objelerin yerleştirilmesi için 5 bölge oluşturulmuştur. Şekil 3.6 'da görüldüğü gibi robot ve objeler arasında 100 cm olacak şekilde ayarlama yapılmıştır. Robot objelere doğru hareket ettirilmiş ve bu hareket sırasında veri toplanmıştır.



Şekil 3.6 Veri toplama deney düzeneği

Her bölge için yerleştirilen objeler farklı açılarda konumlandırılarak veri alınmıştır. Toplamda 270 veri dosyası oluşturulmuştur. Bu veri dosyaları oluşturulurken dosya ismine hangi bölgede hangi objenin olduğu işlenmiştir. Şekil 3.6 'da gösterildiği

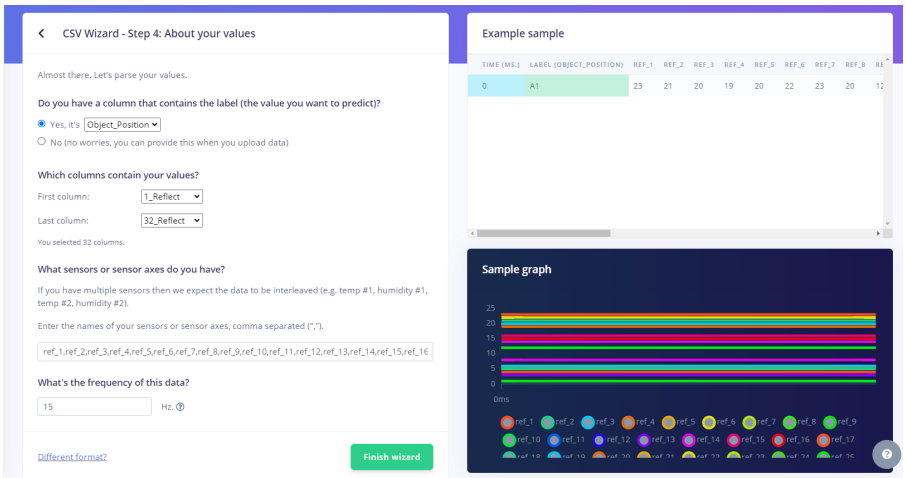
gibi ilk 80 cm 'lik ve son 20 cm 'lik olan alanlar obje var ve obje yok olarak 2 dosyaya ayrılmıştır. Ayrılan bu dosyalar ile 540 veri dosyası elde edilmiştir. Veri dosyası isimleri; "YeşilTop-A5-0","Terlik-A1-1" şeklinde objenin tipi, hangi bölgede olduğu ve objenin varlığı olarak kaydedilmiştir. Sonuç olarak Tablo 3.1 'de verildiği gibi 6 sınıftan oluşan bir model oluşturmak için veri toplanmıştır.

Tablo 3.1 Veri sınıfları

Sınıf - Model Çıktısı
Engelin Varlığı (A0)
Engelin Alanı 1 (A1)
Engelin Alanı 2 (A2)
Engelin Alanı 3 (A3)
Engelin Alanı 4 (A4)
Engelin Alanı 5 (A5)

3.4 Kullanılan Tiny-ML Platformu ve Önerilen CNN Modeli

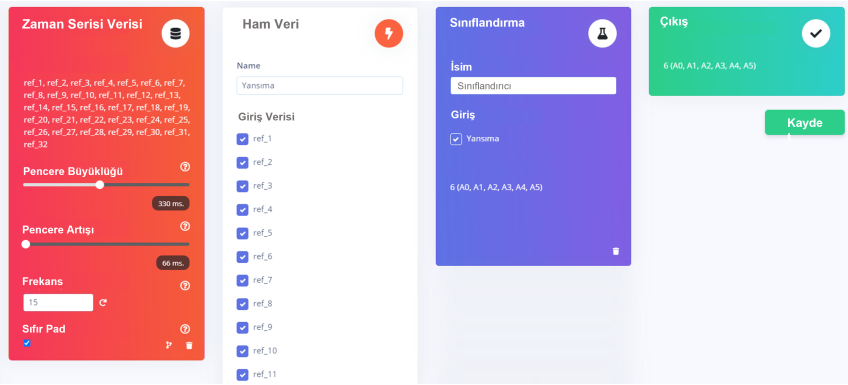
Bu tez çalışmasında Tiny-ML platformu olarak "EdgeImpulse" kullanılmıştır [34]. Bu platformun seçilme sebepleri desteklediği yazılım kütüphaneleri (TensorFlow ve Keras), donanımlar ve kullanışlı olmasıdır. Platformda öncelikle sınıflandırılmış verilerin aktarılması yapılmıştır. Veri dosyaları CSV formatında bir dosya olarak tanıtılmaktadır ve Şekil 3.7 'deki gibi ayar yapılabilmektedir.



Şekil 3.7 Veri aktarımı

Sonraki aşamada, veri klasörü seçilerek tüm verilerin otomatik olarak platforma belirtilen konfigürasyonda aktarımı tamamlanır. Platform üzerine veri eklemek için detaylı dokümantasyon bulunmaktadır [68].

Aktarılan sınıflandırılmış verilerin model girişi Şekil 3.8 'de gösterildiği gibi zaman serisi olarak veri pencere genişliğin 5 küme yani 330 ms seçilmiştir.



Şekil 3.8 Sınıflandırma modeli

Toplanan 8x8 boyutundaki veriler kavramsal olarak gri tonlu bir resim gibi düşünülebilir. Bu nedenle evrişimsel sinir ağı (CNN) modelleri iyi sonuçlar verebilmektedir. Kullanılan platform üzerinde CNN katmaları seçilebilmekte ve kodlanabilmektedir. Bu tez çalışmasında kullanılan ağ katmanları Şekil 3.9 'da basit bir biçimde gösterilmiştir.

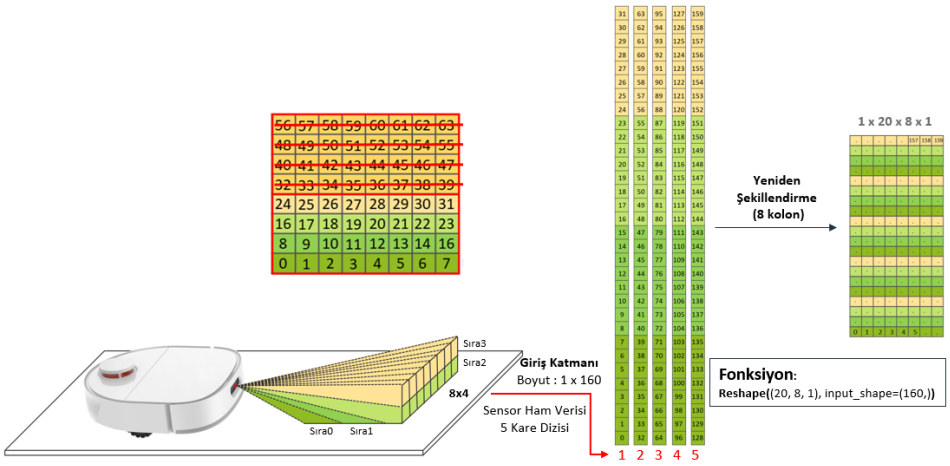


Şekil 3.9 Ağ katmanları

Kullanılan evrişimsel sinir ağının detayları aşağıdaki maddelerde verilmiştir. Tüm CNN katmanları Şekil 3.14 'te gösterilmiştir.

- **Giriş ve Yeniden Şekillendirme Katmanları**

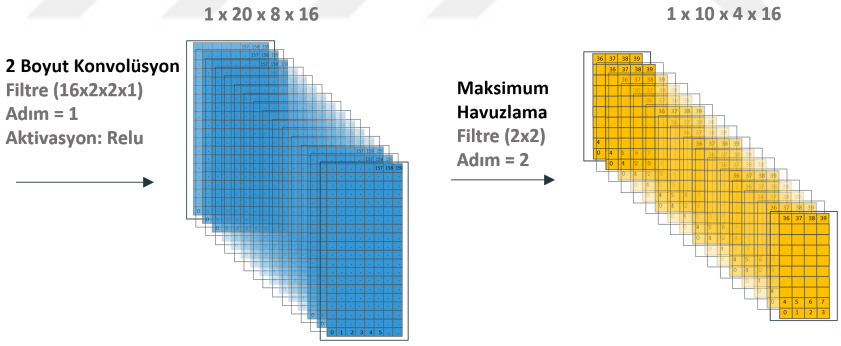
  Önerilen yöntemde, sensör verisinin 5 veri kümesinin kullanılması ve ilk 4 sıra sensör verisinin değerli olduğu belirlenmiştir. Şekil 3.10 'da gösterildiği gibi 5 veri kümesi 8x4 boyutundaki sensör verisi 1x160 olarak giriş katmanına verilmektedir. Bu sensör verisi; sinyal seviyesi, mesafe, mesafe değişimi veya yansıma verilerinden biridir. İkinci aşama olarak tek boyutlu veri yeniden şekillendirilir. Bu aşamada veri 20x8 boyutuna çevrilmektedir.



Şekil 3.10 Giriş ve yeniden şekillendirme katmanları

- **1. 2 Boyutlu Evrişim Katmanı**

  Yeniden şekillendirilen 20x8 boyutundaki veri kümesine Şekil 3.11 'de görüldüğü gibi sıfır pad yapısı, 2x2 filtre boyutu, 16 filtre sayısı ve 1 adıma göre 2 boyutlu evrişimsel katman uygulanmıştır. Ek olarak bu katmanın peşine sıfır pad yapısına sahip, 2x2 boyutunda ve 2 adıma göre uygulanmış maksimum havuzlama katmanı gelmektedir. Bu katman sonrasında veri 10x4x16 boyutuna ulaşmıştır.



Fonksiyon:

Conv2D(16, kernel_size=2, kernel_constraint=tf.keras.constraints.MaxNorm(1), padding='same', activation='relu'**)**

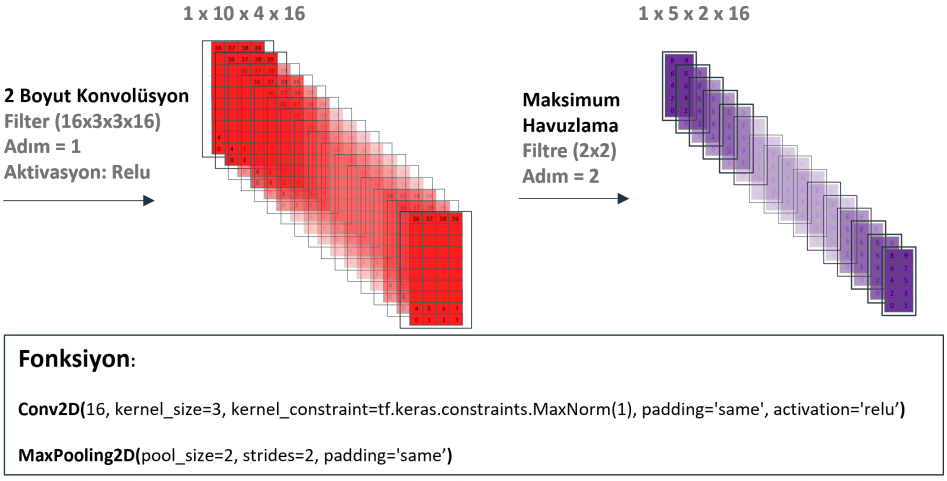
MaxPooling2D(pool_size=2, strides=2, padding='same'**)**

Şekil 3.11 1. 2 Boyutlu evrişim katmanı

- **2. 2 Boyutlu Evrişim Katmanı**

  1. evrişimsel ve havuzlama katmanından çıkan 10x4x16 boyutundaki veri kümesine Şekil 3.12 'de görüldüğü gibi sıfır pad yapısı, 3x3 filtre boyutu, 16 filtre sayısı ve 1 adıma göre 2 boyutlu evrişimsel katman uygulanmıştır. Ek olarak bu katmanın peşine sıfır pad yapısına sahip, 2x2 boyutunda ve

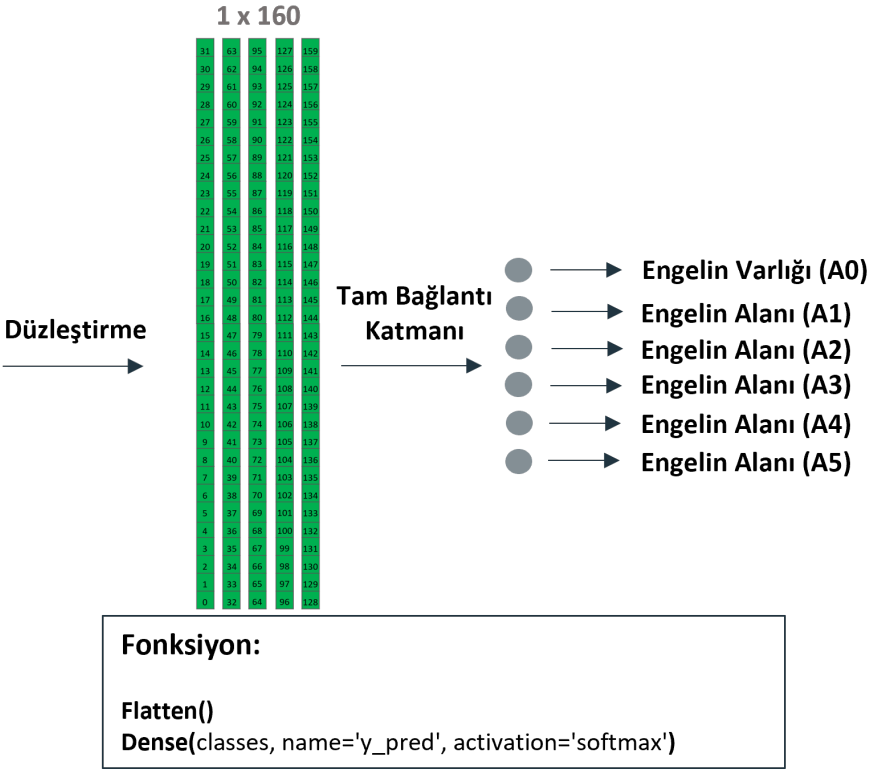
2 adıma göre uygulanmış maksimum havuzlama katmanı gelmektedir. Bu katman sonrasında 5x2x16 boyutuna ulaşılmıştır.



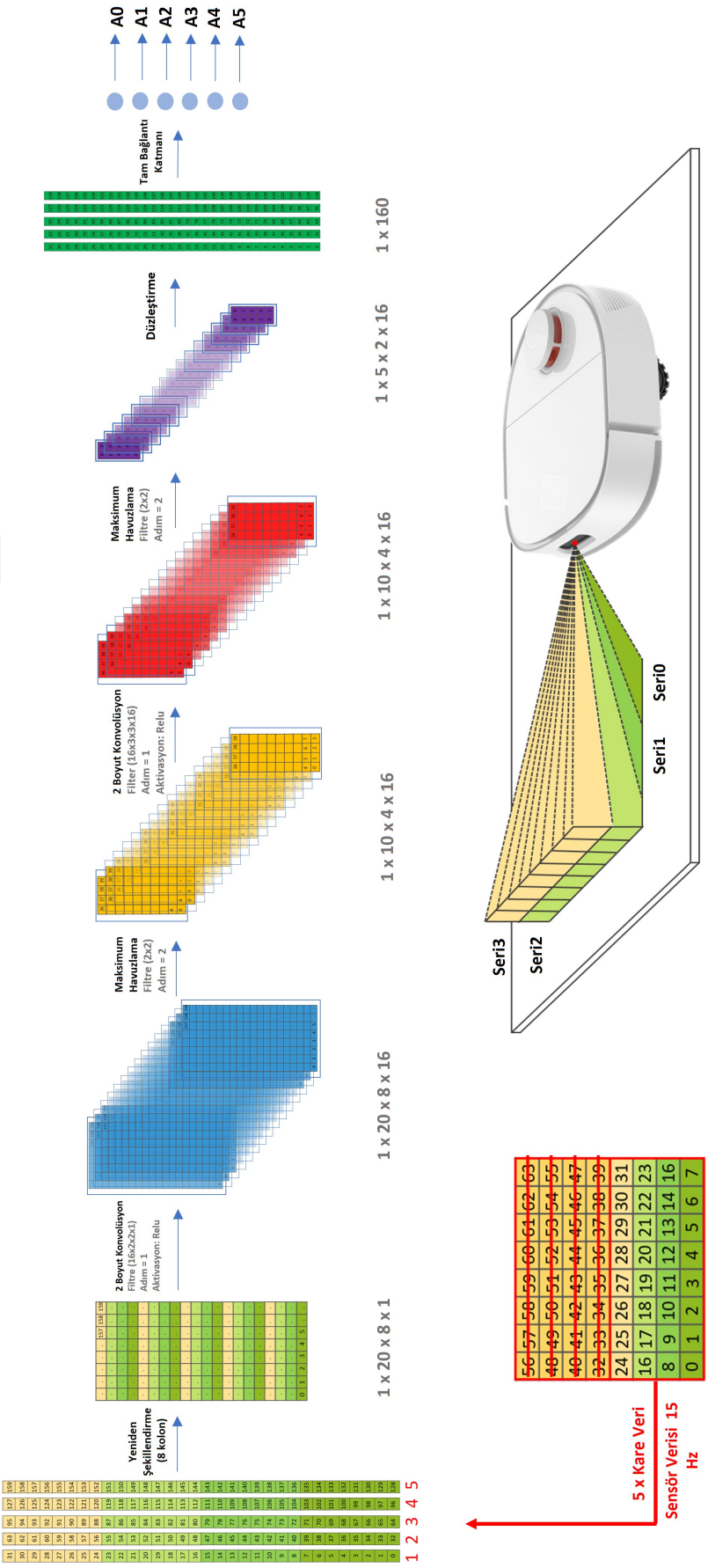
Şekil 3.12 2. 2 Boyutlu evrişim katmanı

- **Düzleştirme ve Tam Bağlantı Katmanı**

2. evrişimsel ve havuzlama katmanından çıkan 5x2x16 boyutundaki veri kümesine Şekil 3.13 'te görüldüğü gibi düzleştirme katmanı uygulanmaktadır. Son aşamada aktivasyon fonksiyonu olarak softmax kullanan ve 6 sınıf çıktısı veren tam bağlantı katmanı bulunmaktadır.



Şekil 3.13 Düzleştirme ve tam bağlantı katmanı



Şekil 3.14 Tüm CNN modeli

4
UYGULAMA SONUÇLARI

Tiny-ML platformu üzerinde kurulan CNN modeli için birden fazla ölçüme sahip olan sensör çıktılarının sonuçları alt başlıklarda değerlendirilmiştir. Yazılım içerisinden çekilen model özeti Şekil 4.1 'de gösterilmiştir. Modelde test edilen sinyal seviyesi, mesafe, mesafe değişimi ve yansıma sensör verileri kullanılmıştır. Ek olarak platform üzerinden TFL kütüphanesinden yararlanılarak 2 Tiny-ML modeli oluşturulmuştur. Bunlardan ilki niceleme ve budama işlemi görmemiş yani optimizasyon yapılmamış 32 bitlik modeldir. İkincisi ise niceleme ve budama işlemi görmüş yani optimizasyon yapılmış 8 bitlik modeldir. Bu iki modele göre sensör verilerinin başarım sonuçları alt başlıklarda karşılaştırılmıştır.

Layer (type)	Output Shape	Param #
reshape (Reshape)	(None, 20, 8, 1)	0
conv2d (Conv2D)	(None, 20, 8, 16)	80
max_pooling2d (MaxPooling2D)	(None, 10, 4, 16)	0
dropout (Dropout)	(None, 10, 4, 16)	0
conv2d_1 (Conv2D)	(None, 10, 4, 16)	2320
max_pooling2d_1 (MaxPooling2D)	(None, 5, 2, 16)	0
dropout_1 (Dropout)	(None, 5, 2, 16)	0
flatten (Flatten)	(None, 160)	0
y_pred (Dense)	(None, 6)	966

Total params: 3,366
Trainable params: 3,366

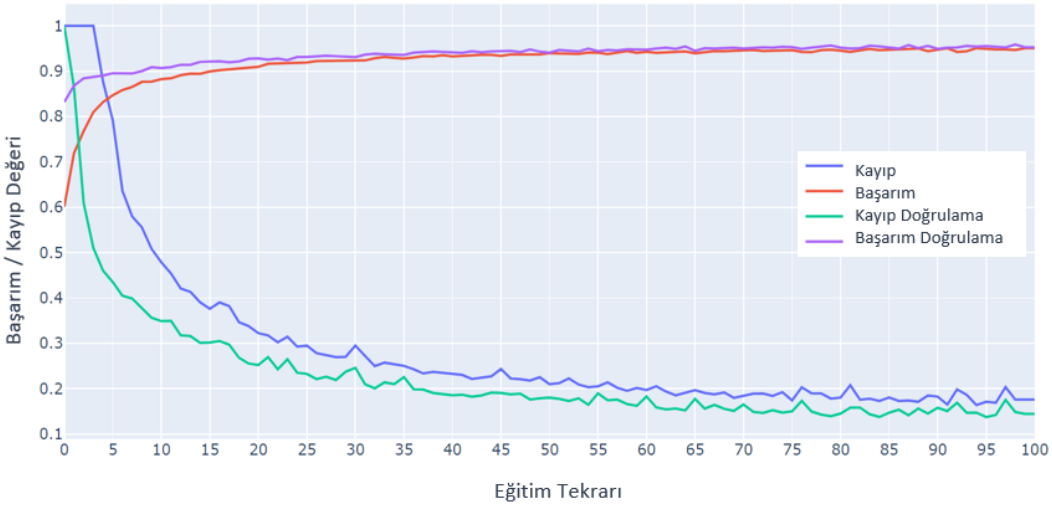
Şekil 4.1 Model özeti

Eğitim ve test verileri %80 ve %20 oranlarında ayrılmıştır. Eğitim verisi içerisinde de %20 doğrulama verisi belirlenmiştir. Model eğitilirken daha önce görmediği doğrulama verisi üzerinde değerlendirme yapılır. Eğitim ve doğrulama verisi birbirine yakın ise bu durum iyiye işarettir. Son olarak eğitim parametreleri ise aşağıdaki gibidir.

- Eğitimin parti büyüklüğü : 32
- Eğitim tekrar sayısı : 100
- Öğrenme oranı : 0,0005

4.1 Sinyal Seviyesi Verisi Model Sonucu

Eğitim ve doğrulama verilerinden oluşan öğrenim grafiği Şekil 4.2 'de verilmiştir. Görüldüğü gibi model başarımı ve doğrulama verisi birbirine yakındır. Eğitimin sonucunda doğrulama verilerine bakıldığında %95,6 başarım gerçekleşmiştir.



Şekil 4.2 Sinyal seviyesi verisi öğrenim grafiği

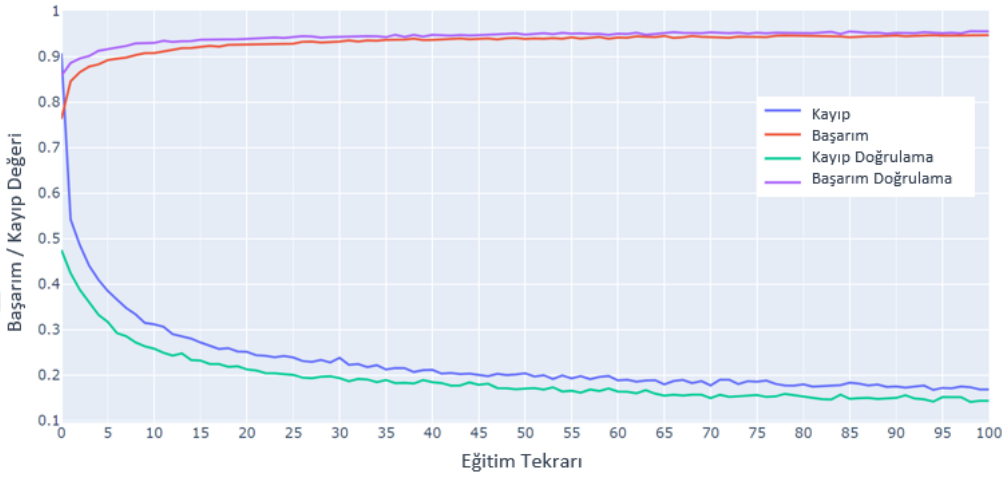
32 bitlik modelin doğrulama verilerine göre %95,6 başarım gerçekleşmiştir. 8 bitlik modelde ise %20,9 'luk bir düşüş ile %74,7 başarım sağlanmıştır. 32 bitlik modelin test verisi kullanıldığındaki test başarımı %94,85 olarak gerçekleşmiştir. Bu denemeye ait karışıklık matrisi Tablo 4.1 'de verilmiştir.

Tablo 4.1 Sinyal seviyesi modeli karışıklık matrisi

	Tahmin Değerleri						
Gerçek Değerler	**A0**	**A1**	**A2**	**A3**	**A4**	**A5**	**Belirsiz**
A0	99,4%	0,1%	0%	0,2%	0,1%	0%	0,3%
A1	12,3%	81,4%	0,7%	0%	0%	0%	5,6%
A2	8,8%	1,3%	85,9%	0%	0%	0%	4,0%
A3	4%	0%	0%	95,2%	0%	0%	0,8%
A4	6%	0%	0%	0%	83,3%	3%	7,7%
A5	10,6%	0,6%	0%	0%	0%	86,1%	2,8%
F1 Skoru	0,98	0,89	0,92	0,97	0,90	0,90	

4.2 Yansıma Verisi Model Sonucu

Eğitim ve doğrulama verilerinden oluşan öğrenim grafiği Şekil 4.3 'te verilmiştir. Görüldüğü gibi model doğruluğu ve doğrulama verisi birbirine yakındır. Eğitimin sonucunda doğrulama verilerine bakıldığında 32 bitlik modelin %95,6 başarımı gerçekleşmiştir.



Şekil 4.3 Yansıma verisi öğrenim grafiği

8 bitlik model için doğrulama verilerine göre %95,5 başarım gerçekleşmiştir. 8 bit ve 32 bitlik modellerin başarım oranları çok yakındır. 32 bitlik modelin test verisi kullanıldığındaki test başarımı %94,85 olarak gerçekleşmiştir. Bu denemeye ait karışıklık matrisi Tablo 4.2 'de verilmiştir.

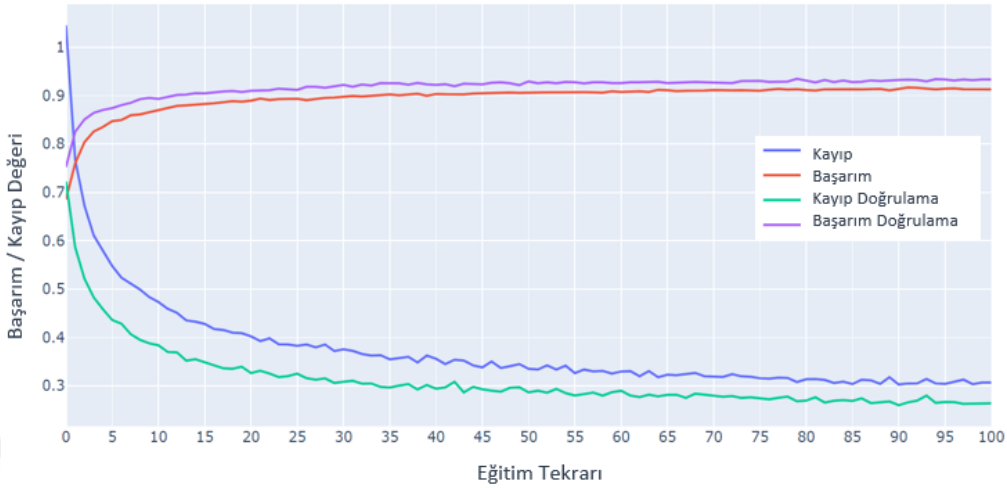
Tablo 4.2 Yansıma verisi modeli karışıklık matrisi

	Tahmin Değerleri						
Gerçek Değerler	**A0**	**A1**	**A2**	**A3**	**A4**	**A5**	**Belirsiz**
A0	98,6%	0,1%	0,1%	0,0%	0,3%	0,1%	0,8%
A1	14,1%	84,4%	0%	0%	0%	0%	1,5%
A2	1,8%	2,7%	89,0%	0%	0%	0%	6,4%
A3	3,1%	0%	0%	86,2%	2,7%	0%	8,1%
A4	4,3%	0%	0%	0%	90,1%	1,7%	3,9%
A5	13,4%	0%	0%	0%	1,2%	81,4%	4,1%
F1 Skoru	0,98	0,90	0,94	0,92	0,91	0,88	

4.3 Mesafe Değişimi Verisi Model Sonucu

Eğitim ve doğrulama verilerinden oluşan öğrenim grafiği Şekil 4.4 'te verilmiştir. Model doğruluğu ve doğrulama başarımı, yansıma verisini kullanan modeldeki gibi

birbirine yakındır. Eğitimin sonucunda doğrulama verilerine göre 32 bitlik modelin %93,2 başarımı gerçekleşmiştir.



Şekil 4.4 Mesafe değişimi verisi öğrenim grafiği

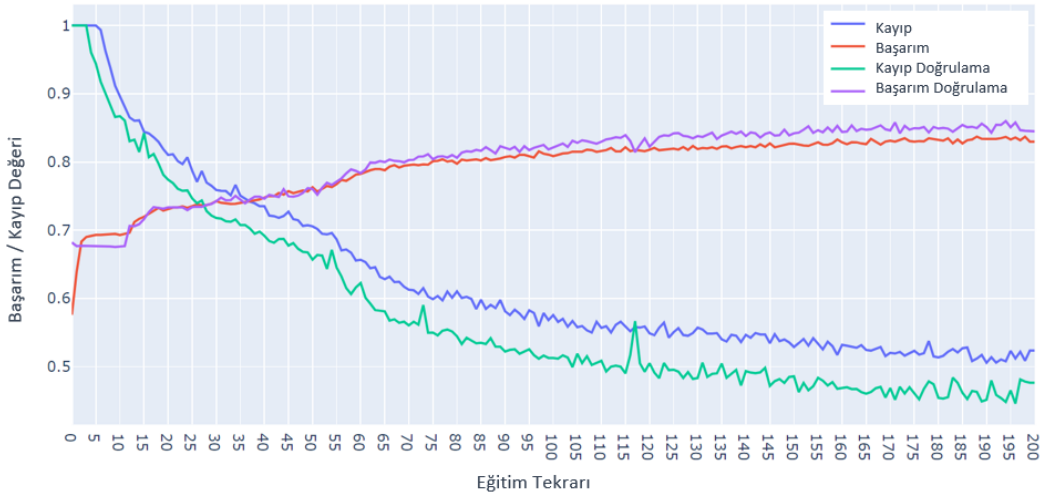
8 bitlik modelin doğrulama verilerine göre %3 'lük bir düşüşle %90,3 başarım gerçekleşmiştir. 32 bitlik modelin test verisi kullanıldığındaki test başarımı %91,67 olarak çıkmıştır. Bu denemeye ait karışıklık matrisi Tablo 4.3 'te verilmiştir.

Tablo 4.3 Mesafe değişimi verisi modeli karışıklık matrisi

	Tahmin Değerleri						
Gerçek Değerler	**A0**	**A1**	**A2**	**A3**	**A4**	**A5**	**Belirsiz**
A0	99,3%	0,2%	0,0%	0%	0%	0%	0,5%
A1	28,1%	60,1%	0,4%	0%	0%	0%	11,5%
A2	12,0 %	2,3%	75,6%	0%	0%	0%	10,1%
A3	5,4 %	0%	0%	89,7%	0,8%	0%	4,1%
A4	13,2 %	0%	0%	0,9%	76,6%	3,8%	5,5%
A5	16,4%	0%	0%	0%	0,5%	77,6%	5,5%
F1 Skoru	0,96	0,73	0,86	0,94	0,86	0,85	

4.4 Mesafe Verisi Model Sonucu

Eğitim ve doğrulama verilerinden oluşan öğrenim grafiği Şekil 4.5 'te verilmiştir. Eğitim tekrar sayısı 100, 200, 300 ve öğrenme oranı 0,0001, 0,0005, 0,001 olarak test edilmiştir. En yüksek başarım eğitimin tekrar sayısı 200 ve öğrenme oranı 0,0005 olarak gerçekleştirilmiştir. Ancak eğitimin sonucunda doğrulama verilerine bakıldığında 32 bitlik modelde %73,21 başarım gerçekleşmiş ve diğer veri modellerinin gerisinde olduğu görülmüştür.



Şekil 4.5 Mesafe verisi öğrenim grafiği

8 bitlik modele bakıldığında %73,3 başarım gerçekleşmiştir. Model test edildiğinde ise test başarımı %73 olarak alınmıştır. Bu denemeye ait karışıklık matrisi Tablo 4.4 'te verilmiştir.

Tablo 4.4 Mesafe verisi modeli karışıklık matrisi

	Tahmin Değerleri						
Gerçek Değerler	**A0**	**A1**	**A2**	**A3**	**A4**	**A5**	**Belirsiz**
A0	90,4%	0,2%	0,0%	0%	0%	0%	0%
A1	13,5%	39,5%	1,4%	0%	0%	0%	0%
A2	3,8 %	0%	44,6%	2,9%	0%	0%	0%
A3	2,7 %	0,4%	0%	61,6%	0%	0%	0%
A4	7,9 %	0%	0,4%	5,2%	34,5%	0%	5,9%
A5	12,3%	0%	0,5%	0%	10,3%	5,9%	0,11%
F1 Skoru	0,93	0,56	0,60	0,73	0,48	0,11	

4.5 Tüm Sensör Verilerinin Özet Sonuçları

Tüm sensör verileri için model başarımları Tablo 4.5 'te verilmiştir. Yansıma verisi ile oluşturulan 32 bit ve 8 bitlik modellerin sonuçları yakın ve yüksek doğrulukta gerçekleştirilmiştir. Ek olarak bu modelde arayüz zamanı diğer modellere göre oldukça iyi sonuç vermiştir. Modeller aynı CNN yapısını kullandığı için kapladıkları hafızalar aynıdır.

Tablo 4.5 Özet model sonuçları

Veri Tipi	32 Bit Model Test Başarısı	8 Bit Model Test Başarısı	Arayüz Zamanı	Dinamik Hafıza	Hafıza
Sinyal Seviyesi	95,38%	74,7%	13 ms	8,1K	53,4K
Yansıma	94,85%	95,5%	1 ms	8,1K	53,4K
Mesafe Değişimi	91,67%	90,3%	13 ms	8,1K	53,4K
Mesafe	73,21%	73,3%	13 ms	8,1K	53,4K

5
SONUÇ

Günümüzde otonom hareket edebilen robotların varlığı evlerde gün geçtikçe artmaktadır. Özellikle temizlik robotları yaygın bir şekilde kullanılmaktadır. Robotlar sürekli hareket halinde olduklarından dolayı hareketlerini sınırlayabilecek engeller ile sıkça karşılaşmaktadırlar. Bu engelleri algılama ve etrafından dönme gibi aksiyonlara sahip olmaları gerekmektedir. Bu problemdeki kısıtlar, algılama metodu ve maliyetli olmasıdır. Engel olabilecek nesnelerin yüksek çözünürlükte algılanması hem işlem hacmini hem de ürün maliyetini arttırmaktadır. Bu tez çalışmasında entegre ToF sensörü ile birlikte mikro denetleyici üzerinde çalışabilen Tiny-ML modeli geliştirilmiştir. Bu sayede işlem hacmi azaltılmıştır. Aynı zamanda maliyet ve enerji kullanımını düşürülmüştür.

Kullanılan ToF sensöründen alınan birçok ölçüm verisi bulunmaktadır. Oluşturulan bir deney düzeneği ile kontrollü bir şekilde veri toplanmış ve bir Tiny-ML platformu olan "EdgeImpulse" üzerinden makine öğrenmesi modelleri oluşturulmuştur. Oluşturulan modellerde evrişimsel sinir ağı (CNN) kullanılmıştır. Bu modellerin mikro denetleyici üzerinde daha az alan kaplaması ve hızlı bir şekilde çalışması için niceleme işlemi yapılmıştır. Niceleme işlemi, 32 bitlik modeli sıkıştırarak 8 bitlik optimize edilmiş modele çevirmektedir. 32 bitlik ve 8 bitlik bu model sonuçları karşılaştırılmıştır. Tüm sonuçlara bakıldığında yansıma verisi kullanılan 8 bitlik modelde %95 başarıma ulaşılmıştır.

Bu çalışmada sadece ev tipi temizlik robotların karşılaşabileceği objeler kullanılmıştır. Çalışmayı ileriye taşımak için farklı ortam koşullarında, farklı zeminlerde ve farklı engellerde testlerin gerçekleştirilmesi önerilmektedir. Ayrıca Tiny-ML modelinin daha seri bir şekilde çalışması için robotun hareket hızına bağlı olarak sensör ölçüm frekansının arttırılması önem arz etmektedir. Bu nedenle, model başarısını arttırmak için çalışılan robot uygulamasına göre işlem hacmi ve hafızanın izin verdiği kadarıyla veri toplama hızının arttırılması önerilmektedir.

KAYNAKÇA

[1] R. A. Deshmukh, M. A. Hasamnis, "A novel multi-sensor tactile detachable bumper design for obstacle avoidance, collision detection, and recovery in autonomous mobile robots," in *2023 4th International Conference for Emerging Technology (INCET)*, 2023, pp. 1–5. doi: 10.1109/INCET57972.2023.10170494.

[2] T. R. Ohm, E. Amaral, A. Deschenes, "Floating bumper in autonomous cleaning robot]," U.S. Patent 11213181B2, 20.11.2019.

[3] B. Seth, D. Samuel, D. Kyle, "Contact sensors for a mobile robot," European Patent 3100661B1, 20.11.2015.

[4] M.-C. Kang, K.-S. Kim, D.-K. Noh, J.-W. Han, S.-J. Ko, "A robust obstacle detection method for robotic vacuum cleaners," *IEEE Transactions on Consumer Electronics*, vol. 60, no. 4, pp. 587–595, 2014. doi: 10.1109/TCE.2014.7027291.

[5] R. Zhang, Y. Fan, H. Jin, R. Zhao, D. Liu, "Visual navigation for fast trajectory planning and accurate trajectory following based on legged robots in unknown environments," in *2023 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 2023, pp. 1–7. doi: 10.1109/ROBIO58561.2023.10354595.

[6] A. R. García, L. R. Miller, C. F. Andrés, P. J. N. Lorente, "Obstacle detection using a time of flight range camera," in *2018 IEEE International Conference on Vehicular Electronics and Safety (ICVES)*, 2018, pp. 1–6. doi: 10.1109/ICVES.2018.8519501.

[7] G. Mora-Martín *et al.*, "High-speed object detection with a single-photon time-of-flight image sensor," *Optics Express*, vol. 29, no. 21, p. 33 184, 2021, issn: 1094-4087. doi: 10.1364/oe.435619.

[8] A. Kongpecth, N. Kwankeo, V. Manthamkarn, "360 degrees object detection using multiple tof sensors for educational robot," in *2022 19th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, 2022, pp. 1–4. doi: 10.1109/ECTI-CON54298.2022.9795626.

[9] STMicroelectronics, *2d lidar using multiple vl53l1x time-of-flight long distance ranging sensors (1.0.0)*, Erişim: 24 Aralık 2023. [Çevrimiçi]. Erişim adresi: https://www.st.com/en/embedded-software/stsw-img017.

[10] G.-Y. Kim, S.-H. Eom, C.-W. Lee, W.-S. Kim, E.-H. Lee, "Vehicle wheel edge detection using tof sensor for low cost of vehicle specification measurement system," in *2023 International Conference on Electronics, Information, and Communication (ICEIC)*, 2023, pp. 1–4. doi: 10.1109/ICEIC57457.2023.10049860.

[11] S. Irigaray, *"Time-of-Flight sensor solutions and applications"*, Erişim: 24 Aralık 2023. [Çevrimiçi]. Erişim adresi: https://www.st.com/Presentations/TimeofFlightsensor/Applications.pdf.

[12] S. Boner, C. Vogt, M. Magno, "Tiny tcn model for gesture recognition with multi-point low power tof-sensors," in *2022 IEEE 4th International Conference on Artificial Intelligence Circuits and Systems (AICAS)*, 2022, pp. 356–359. doi: 10.1109/AICAS54282.2022.9869848.

[13] STMicroelectronics, *Turnkey gesture recognition solution based on vl53l5cx, vl53l7cx and vl53l8cx multizone time-of-flight ranging sensors (1.9.2)*, Erişim: 24 Aralık 2023. [Çevrimiçi]. Erişim adresi: https://www.st.com/en/embedded-software/stsw-img035.html.

[14] S. Salerno, *8x8 tof gesture classification - arduino rp2040 connect*, Erişim: 24 Aralık 2023. [Çevrimiçi]. Erişim adresi: https://docs.edgeimpulse.com/experts/novel-sensor-projects/tof-gesture-classification-arduino-rp2040-connect.

[15] STMicroelectronics, *Hand posture recognition without camera module*, Erişim: 24 Aralık 2023. [Çevrimiçi]. Erişim adresi: https://stm32ai.st.com/use-case/hand-posture-recognition-without-camera-module/.

[16] STMicroelectronics, *Floor type detection for vacuum cleaners*, Erişim: 24 Aralık 2023. [Çevrimiçi]. Erişim adresi: https://stm32ai.st.com/use-case/floor-type-detection-for-vacuum-cleaners.

[17] Y. Abadade, A. Temouden, H. Bamoumen, N. Benamar, Y. Chtouki, A. S. Hafid, "A comprehensive survey on tinyml," *IEEE Access*, vol. 11, pp. 96 892–96 922, 2023. doi: 10.1109/ACCESS.2023.3294111.

[18] MLCommons, *Machine learning performans benchmark (1.2)*, Erişim: 10 Mart 2024. [Çevrimiçi]. Erişim adresi: https://mlcommons.org/benchmarks/inference-tiny/.

[19] C. R. Banbury, "Mlperf tiny benchmark," *CoRR*, vol. abs/2106.07597, 2021. doi: 10.48550/arXiv.2106.07597.

[20] Tensorflow, *Tiny ml supported platforms*, Erişim: 17 Mart 2024. [Çevrimiçi]. Erişim adresi: https://www.tensorflow.org/lite/microcontrollers/#supported_platforms.

[21] Arduino, *Arduino nano 33 ble sense*, Erişim: 19 Mart 2024. [Çevrimiçi]. Erişim adresi: https://docs.arduino.cc/hardware/nano-33-ble-sense.

[22] A. Osman, U. Abid, G. Lepri, M. Perotto, D. Brunelli, "Tinyml platforms benchmarking," *arXiv (Cornell University)*, Nov. 2021. doi: 10.48550/arxiv.2112.01319.

[23] P. P. Ray, "A review on tinyml: State-of-the-art and prospects," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 4, pp. 1595–1623, 2022, issn: 1319-1578. doi: 10.1016/j.jksuci.2021.11.019.

[24] P.-E. Novac, G. Boukli Hacene, A. Pegatoquet, B. Miramond, V. Gripon, "Quantization and deployment of deep neural networks on microcontrollers," *Sensors*, vol. 21, no. 9, p. 2984, Apr. 2021. doi: 10.3390/s21092984.

[25] STMicroelectronics, *Stm32 cube ai (9.0.0)*, Erişim: 19 Mart 2024. [Çevrimiçi]. Erişim adresi: https://www.st.com/en/embedded-software/x-cube-ai.html#overview.

[26] UTensor, *Micro tensor*, Erişim: 19 Mart 2024. [Çevrimiçi]. Erişim adresi: https://utensor.github.io/website/.

[27] J. Nordby, M. Cooke, A. Horvath, *emlearn: Machine Learning inference engine for Microcontrollers and Embedded Devices (0.20.4)*, 2024. doi: 10.5281/zenodo.10995032.

[28] J. Nordby, *Emlearn: Machine learning inference engine for microcontrollers and embedded devices*, version 0.8.2, 2019. doi: 10.5281/zenodo.2589394.

[29] Apache Software Foundation, *Apache tvm (0.17)*, Erişim: 19 Mart 2024. [Çevrimiçi]. Erişim adresi: https://tvm.apache.org/.

[30] Fraunhofer IMS, *Ai for embedded systems*, Erişim: 20 Mart 2024. [Çevrimiçi]. Erişim adresi: https://www.ims.fraunhofer.de/en/Business-Unit/Industry/Industrial-AI/Artificial-Intelligence-for-Embedded-Systems-AIfES.html.

[31] L. Wulfert, J. Kühnel, L. Krupp, J. Viga, C. Wiede, P. Gembaczka, "Aifes: A next-generation edge ai framework," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–16, 2024. doi: 10.1109/TPAMI.2024.3355495.

[32] J. Ma, *A higher-level Neural Network library on Microcontrollers (NNoM)*, version v0.4.2, 2020. doi: 10.5281/zenodo.4158710.

[33] J. Ma, *Neural network on microcontroller github*, Erişim: 20 Mart 2024. [Çevrimiçi]. Erişim adresi: https://github.com/majianjia/nnom.

[34] Edge Impulse, *Edge impulse getting started*, Erişim: 20 Mart 2024. [Çevrimiçi]. Erişim adresi: https://docs.edgeimpulse.com/docs.

[35] S. Hymel *et al.*, "Edge impulse: An mlops platform for tiny machine learning," *arXiv preprint*, 2022. doi: 10.48550/arXiv.2212.03332.

[36] Edge Impulse, *Edge impulse data acquisition*, Erişim: 22 Mart 2024. [Çevrimiçi]. Erişim adresi: https://docs.edgeimpulse.com/docs/edge-impulse-studio/data-acquisition.

[37] Edge Impulse, *Edge impulse supported mcus*, Erişim: 20 Mart 2024. [Çevrimiçi]. Erişim adresi: https://docs.edgeimpulse.com/docs/edge-ai-hardware/mcu.

[38] Edge Impulse, *Edge impulse run interface*, Erişim: 22 Mart 2024. [Çevrimiçi]. Erişim adresi: https://docs.edgeimpulse.com/docs/run-inference/cpp-library.

[39] SensiML, *Sensiml website*, Erişim: 22 Mart 2024. [Çevrimiçi]. Erişim adresi: https://sensiml.com.

[40] SensiML, *Sensiml data studio*, Erişim: 22 Mart 2024. [Çevrimiçi]. Erişim adresi: https://datastudiobysensiml.com.

[41] SensiML, *Sensiml documantation*, Erişim: 22 Mart 2024. [Çevrimiçi]. Erişim adresi: https://sensiml.com/documentation/data-capture-lab/overview.html.

[42] SensiML, *Sensiml application example*, Erişim: 22 Mart 2024. [Çevrimiçi]. Erişim adresi: https://sensiml.com/resources/app-library.

[43] Renesas, *Reality-ai website*, Erişim: 22 Mart 2024. [Çevrimiçi]. Erişim adresi: https://www.renesas.com/us/en/products/microcontrollers-microprocessors/reality-ai.

[44] Renesas, *Reality ai e2 studio entegrasyonu*, Erişim: 22 Mart 2024. [Çevrimiçi]. Erişim adresi: https://www.renesas.com/us/en/document/apn/e2-studio-integration-getting-start-guide-using-ra6t2?r=1626801.

[45] STMicroelectronics, *Nano edge ai studio website*, Erişim: 22 Mart 2024. [Çevrimiçi]. Erişim adresi: https://stm32ai.st.com/nanoedge-ai.

[46] STMicroelectronics, *Nano edge ai studio*, Erişim: 22 Mart 2024. [Çevrimiçi]. Erişim adresi: https://wiki.st.com/stm32mcu/wiki/AI:NanoEdge_AI_Studio.

[47] STMicroelectronics, *How to create a current sensing classifier using nanoedge ai studio*, Erişim: 22 Mart 2024. [Çevrimiçi]. Erişim adresi: https://wiki.st.com/stm32mcu/wiki/AI:How_to_create_a_current_sensing_classifier_using_NanoEdge_AI_Studio.

[48] Tensorflow, *Pruning in keras example*, Erişim: 22 Mart 2024. [Çevrimiçi]. Erişim adresi: https://www.tensorflow.org/model_optimization/guide/pruning/pruning_with_keras.

[49] Tensorflow, *Tensorflow pruning results*, Erişim: 22 Mart 2024. [Çevrimiçi]. Erişim adresi: https://www.tensorflow.org/model_optimization/guide/pruning#results.

[50] Tensorflow, *Tensorflow quantization*, Erişim: 22 Mart 2024. [Çevrimiçi]. Erişim adresi: https://www.tensorflow.org/model_optimization/guide/quantization/training.

[51] Tensorflow, *Tensorflow quantization image classification*, Erişim: 22 Mart 2024. [Çevrimiçi]. Erişim adresi: https://www.tensorflow.org/model_optimization/guide/quantization/training#image_classification_with_tools.

[52] STMicroelectronics, *Partial fov performances vl53l1 datasheet*, Erişim: 22 Mart 2024. [Çevrimiçi]. Erişim adresi: https://www.st.com/resource/en/datasheet/vl53l1.pdf.

[53] S. Royo, M. Ballesta-Garcia, "An overview of lidar imaging systems for autonomous vehicles," *Applied Sciences*, vol. 9, p. 4093, 2019. doi: 10.3390/app9194093.

[54] OnSemi, *Direct time of flight depth sensing reference designs*, Erişim: 22 Mart 2024. [Çevrimiçi]. Erişim adresi: https://www.onsemi.com/pub/Collateral/TND6341-D.PDF.

[55] STMicroelectronics, *3d dtof all-in-one lidar module with on-chip processing*, Erişim: 26 Mart 2024. [Çevrimiçi]. Erişim adresi: https://www.st.com/en/imaging-and-photonics-solutions/vl53l9ca.html.

[56] AMS-OSRAM, *Tmf8820/21/28 multizone time of flight sensors datasheet*, Erişim: 26 Mart 2024. [Çevrimiçi]. Erişim adresi: https://look.ams-osram.com/m/52236c476132a095/original/TMF8820-21-28-Multizone-Time-of-Flight-Sensor.pdf.

[57] STMicroelectronics, *Vl53l7cx time of flight (tof) 8x8 multizone ranging sensor with 90 degrees fov*, Erişim: 26 Mart 2024. [Çevrimiçi]. Erişim adresi: https://www.st.com/en/imaging-and-photonics-solutions/vl53l7cx.html.

[58] STMicroelectronics, *Stm32f746g discovery kit*, Erişim: 17 Mart 2024. [Çevrimiçi]. Erişim adresi: https://www.st.com/en/evaluation-tools/32f746gdiscovery.html.

[59] STMicroelectronics, *Vl53l7cx datasheet*, Erişim: 26 Mart 2024. [Çevrimiçi]. Erişim adresi: https://www.st.com/resource/en/datasheet/vl53l7cx.pdf.

[60] STMicroelectronics, *Ultra lite driver api (1.2.10)*, Erişim: 26 Mart 2024. [Çevrimiçi]. Erişim adresi: https://www.st.com/en/embedded-software/stsw-img036.html.

[61] STMicroelectronics, *A guide to using the vl53l7cx time-of-flight multizone ranging sensor with 90° fov*, Erişim: 26 Mart 2024. [Çevrimiçi]. Erişim adresi: https://www.st.com/resource/en/user_manual/um3038-a-guide-to-using-the-vl53l7cx-timeofflight-multizone-ranging-sensor-with-90-fov-stmicroelectronics.pdf.

[62] STMicroelectronics, *Stm32746g discovery application examples and template github (1.17.1)*, Erişim: 31 Mart 2024. [Çevrimiçi]. Erişim adresi: https://github.com/STMicroelectronics/STM32CubeF7/tree/master/Projects/STM32746G-Discovery.

[63] STMicroelectronics, *Stm32cube init code generator for windows (6.11.1)*, Erişim: 31 Mart 2024. [Çevrimiçi]. Erişim adresi: https://www.st.com/en/development-tools/stm32cubemx.html.

[64] IAR, *Iar embedded workbench for arm (9.50)*, Erişim: 31 Mart 2024. [Çevrimiçi]. Erişim adresi: https://www.iar.com/products/architectures/arm/iar-embedded-workbench-for-arm/.

[65] SEGGER Microcontroller, *Emwin for stm32*, Erişim: 31 Mart 2024. [Çevrimiçi]. Erişim adresi: https://www.segger.com/products/user-interface/emwin/add-ons/emwin-for-stm32/.

[66] Edge Impulse, *Edgeimpulse iar library*, Erişim: 31 Mart 2024. [Çevrimiçi]. Erişim adresi: https://docs.edgeimpulse.com/docs/run-inference/iar-library#deploy-to-an-iar-embedded-workbench-library.

[67] Edge Impulse, *Iar export project for stm32cubef4*, Erişim: 31 Mart 2024. [Çevrimiçi]. Erişim adresi: https://github.com/edgeimpulse/iar-export-stm32.

[68] Edge Impulse, *Edgeimpulse csv wizard (time series)*, Erişim: 31 Mart 2024. [Çevrimiçi]. Erişim adresi: https://docs.edgeimpulse.com/docs/edge-impulse-studio/data-acquisition/csv-wizard.

TEZDEN ÜRETİLMİŞ YAYINLAR

Makale

1. S. E. ARSLAN ve B. BOLAT, “Tiny machine learning model for obstacle detection with multi-zone time of flight sensors,” Recent Advances in Science and Engineering, pp. 9–14, 2023, doi: 10.14744/rase.2023.0002.