

## İÇİNDEKİLER

Şekil listesi .....	iii
ÖZET .....	iv
ABSTRACT .....	v
1. GİRİŞ.....	1
1.1. PROJENİN AMACI VE ÖNEMİ.....	1
1.2. PROBLEM TANIMI .....	2
1.3. PROJENİN KAPSAMI VE SINIRLARI.....	2
1.4. KULLANILAN YÖNTEMLER VE TEKNOLOJİLER.....	3
2. LİTERATÜR TARAMASI .....	3
2.1. GÖRÜNTÜ İŞLEME VE NESNE TESPİTİ TEKNOLOJİLERİ .....	3
2.1.1. Nesne Tanıma (Nesne Tespiti) Nedir .....	4
2.1.2. Nesne Tanıma Nereelerde Kullanılır? .....	4
2.1.3. Görüntü İşleme Çözümleri ile Nesne Tanıma Nasıl Yapılıyor? .....	5
2.2. MAKİNE ÖĞRENMESİ VE DERİN ÖĞRENME MODELLERİ.....	5
2.2.1. Makine Öğrenimi Nedir? .....	6
2.2.2. Derin Öğrenme Nedir? .....	6
2.3. YOLO(YOU ONLY LOOK ONCE) ALGORİTMASI VE NESNE ALGILAMA(OBJECT DETECTION) .....	8
2.3.1. Nesne Algılama(Object Detection) Nedir? .....	8
2.3.2. YOLO (You Only Look Once) Nedir?.....	8
2.4. SENSÖR VE MİKRODENETLEYİCİ SİSTEMLERİ (ESP8266 VE MPU6050).....	9
2.4.1. NodeMCU V3 ESP8266 ESP-12E Geliştirme Kartı - CP2102.....	9
2.4.2. MPU6050 6 Eksen İvme ve Gyro Sensörü - GY-521.....	10
2.4.3. HW-479 RGB Led Modülü .....	10
2.4.4. Buzzer .....	10
2.5. ADAFRUIT IO PLATFORMU VE KULLANIM ALANLARI .....	11
2.5.1. Öne Çıkan Özellikler.....	11
2.6. VERİ İLETİMİ VE KAYIT.....	12
2.6.1. MQTT Broker Konfigürasyonu.....	12
2.6.2. MQTT İletişimi Kodu .....	12
2.7. BENZER PROJE VE ÇALIŞMALARIN İNCELENMESİ .....	13
3. PROJE YÖNTEMİ.....	14
3.1. VERİ TOPLAMA SÜRECİ .....	14
3.1.1. Hedef Görüntülerin Çekimi .....	14
3.1.2. Roboflow ile Veri Seti Oluşturma ve Etiketleme .....	15
3.2. MODEL EĞİTİMİ.....	16
3.2.1. Ultralytics YOLOv8 Kullanımı.....	16
3.2.2. Model Performansının Değerlendirilmesi .....	16
3.3. DONANIM TASARIMI VE BAĞLANTILAR .....	17
3.3.1. Sensör ve ESP8266 Entegrasyonu .....	17
3.3.2. MPU6050 ile Eğim Algılama.....	17
3.3.3. Buzzer ve LED Bağlantıları .....	17
3.4. YAZILIM GELİŞTİRME.....	18

3.4.1. Python ile Hedef Tespiti .....	18
3.4.2. ESP8266 Üzerinde Ana Kod (main.cpp) .....	18
3.4.3. Adafruit IO ile Veri Gönderimi ve Görselleştirme .....	19
<b>4. PROTOTİP UYGULAMA VE TESTLER.....</b>	<b>20</b>
4.1. SİSTEM TESTLERİ .....	20
4.1.1. Hedef Tespit Doğruluğu Testi.....	20
4.1.2. Sensör Tepki Süreleri ve Performans Testi.....	21
4.2. GERÇEK ZAMANLI ÇALIŞMA VE GERİ BİLDİRİM TESTLERİ .....	22
4.3. TESTLERDEN ELDE EDİLEN SONUÇLAR .....	22
4.4. SONUÇ .....	23
<b>5. SONUÇlar ve tartışma.....</b>	<b>23</b>
5.1. ELDE EDİLEN BULGULARIN DEĞERLENDİRİLMESİ .....	23
5.2. PROJENİN KATKILARI VE SINIRLAMALARI .....	24
5.3. GELECEKTEKİ ÇALIŞMALAR İÇİN ÖNERİLER .....	26
<b>6. kaynaklar.....</b>	<b>27</b>

## ŞEKİL LİSTESİ

Şekil 1	.....	14
Şekil 2	.....	14
Şekil 3	.....	14
Şekil 4	.....	14
Şekil 5	.....	15
Şekil 6	.....	15
Şekil 7	.....	15
Şekil 8	.....	16
Şekil 9	.....	16
Şekil 10	.....	18
Şekil 11	.....	19
Şekil 12	.....	19
Şekil 13	.....	19
Şekil 14	.....	21
Şekil 15	.....	21
Şekil 16	.....	21

## ÖZET

### **AimGuard/MPU6050 Sensörü ile Silah Pozisyon Algılama**

Bu çalışma, askeri eğitim süreçlerinde güvenliği artırmak ve kullanıcı hatalarını en aza indirmek amacıyla geliştirilmiş bir sistem sunmaktadır. Projede, MPU6050 sensörü ve ESP tabanlı mikrodenetleyici kullanılarak silah pozisyonunun hassas bir şekilde algılanması ve hatalı durumlarda LED ve buzzer yardımıyla kullanıcıya görsel ve işitsel geri bildirim verilmesi sağlanmıştır. Ayrıca, bilgisayarlı görüş teknikleriyle, Roboflow kullanılarak etiketlenen veri seti üzerinden YOLOv8 modeli eğitilmiş ve düşman tespiti gerçekleştirilmiştir. Tespit edilen düşman bilgisi, ESP üzerinden sisteme iletilerek kullanıcıya LED yardımıyla uyarı verilmiştir. Gerçek zamanlı veri analizi ve görselleştirme, Adafruit IO platformu aracılığıyla sağlanmıştır. Bu proje, askeri eğitimlerde güvenliği artıran ve teknoloji destekli bir çözüm sunmaktadır.

**Anahtar sözcükler:** Askeri Eğitim, MPU6050, ESP Mikrodenetleyici, YOLOv8, Bilgisayarlı Görü, Derin Öğrenme, Gömülü Sistemler.

## **ABSTRACT**

### **AimGuard/ Weapon Position Detection with MPU6050 Sensor**

This study presents a system developed to enhance safety and minimize user errors during military training processes. The system employs an MPU6050 sensor and ESP-based microcontroller to precisely detect weapon positions and provide visual and auditory feedback through LEDs and a buzzer in case of incorrect positions. Additionally, computer vision techniques were utilized to train the YOLOv8 model on a dataset labeled using Roboflow for enemy detection. Detected information was transmitted to the system via the ESP, triggering LED warnings for the user. Real-time data analysis and visualization were achieved through the Adafruit IO platform. This project offers a technology-supported solution to improve safety in military training.

**Keywords:** Military Training, MPU6050, ESP Microcontroller, YOLOv8, Computer Vision, Deep Learning, Embedded Systems.

# 1. GİRİŞ

Son yıllarda teknolojinin hızla gelişmesi, askeri eğitim ve uygulamalarda yenilikçi sistemlerin kullanılmasını mümkün kılmıştır. Özellikle gömülü sistemler, sensör teknolojileri ve yapay zeka alanındaki ilerlemeler, bu alanda güvenliği arttıran ve kullanıcı hatalarını minimum seviyeye indiren çözümlerin geliştirilmesine olanak sağlamaktadır. Bu proje, askeri eğitim süreçlerinde kullanıcı hatalarını önlemek, güvenliği arttırmak ve performansı geliştirmek amacıyla geliştirilmiştir. Proje, bir silahın doğru pozisyonda tutulup tutulmadığını algılamak ve kullanıcıya anlık geri bildirim sağlamak üzerine odaklanmaktadır. Bununla birlikte, bilgisayarlı görüş teknikleri kullanılarak düşman tespiti yapılması ve bu bilginin sisteme iletilerek kullanıcıya uyarılar verilmesi amaçlanmıştır.

## 1.1. Projenin Amacı ve Önemi

Bu proje, askeri eğitim süreçlerinde **güvenliği artırmak** ve eğitim verimliliğini iyileştirmek amacıyla geliştirilmiştir. Silah eğim algılama ve hedef tanıma sistemlerinin kullanımıyla kullanıcı hatalarının önlenmesi ve performans takibi hedeflenmiştir. Araştırmanın önemini şu başlıklar altında özetleyebiliriz:

- **Doğru Silah Pozisyonunun Sağlanması:** Silahın sağa veya sola yanlış açılarla tutulduğu durumları algılayarak, kullanıcıyı doğru pozisyonda kalmaya yönlendirmek.
- **Hataların Önlenmesi:** Kullanıcıya görsel (LED) ve işitsel (buzzer) geri bildirimler sunarak, hatalı silah tutuşlarının yol açabileceği **kazaların ve yaralanmaların** önüne geçmek.
- **Eğim Faktörünün Yönetimi:** Uzun mesafeli atışlarda, yüzey bozukluklarından kaynaklanan eğimin silah üzerindeki etkisini algılayarak **daha isabetli atışlar** gerçekleştirilmesini sağlamak.
- **Uzaktan Veri Takibi:** MPU6050 sensöründen elde edilen verilerin **Adafruit IO** platformu üzerinden gerçek zamanlı olarak iletilmesi ve analiz edilmesi ile eğitim süreçlerinde daha etkili geri bildirim sağlamak ve verimi arttırmak.
- **Teknolojik Entegrasyon:** Donanım (ESP8266 mikrodenetleyici, MPU6050 ivme sensörü) ve yazılım (Python, Arduino) bileşenleriyle hassas bir şekilde pozisyon takibi sağlamak.
- **Veri Analizi ve Görselleştirme:** Toplanan verilerin bulut tabanlı sistemler üzerinde görselleştirilerek **kullanıcı dostu bir arayüz** ile analiz edilmesini sağlamak.

## 1.2. Problem Tanımı

Silah kullanımında doğru pozisyonun sağlanamaması, askeri eğitimlerde ve saha operasyonlarında güvenlik sorunlarına, kazalara ve performans kayıplarına yol açmaktadır. Özellikle **eğitim faktörlerinin** dikkate alınmaması, uzun mesafeli atışlarda hedefi vurma oranını olumsuz yönde etkilemektedir. Bu sorunların çözümü için geliştirilen sistemin temel problem tanımı şu şekilde özetlenebilir:

1. Silahın sağa veya sola **hatalı açılarla** tutulması, atış isabetini düşürür.
2. Kullanıcıların hatalı silah tutuşlarının fark edilmemesi, **güvenlik sorunlarına** yol açabilir.
3. Eğitim sırasında **geri bildirim mekanizmalarının** eksikliği, hataların tekrarlanmasına neden olur.
4. Anlık veri takibi ve analizinin yapılamaması, kullanıcı performansının etkin şekilde değerlendirilmesini engeller.

Bu problemler doğrultusunda, bir algılama ve uyarı sistemi geliştirilmiş, kullanıcıya **anlık görsel ve işitsel geri bildirimler** sağlanarak hataların önüne geçilmesi amaçlanmıştır.

## 1.3. Projenin Kapsamı ve Sınırları

Proje, yazılım ve donanım entegrasyonu ile silah kullanımında güvenliği artırmaya yönelik bir çözüm sunmaktadır. Projenin kapsamı şu bileşenlerden oluşmaktadır:

1. **Donanım Geliştirme:**
  - ESP8266 tabanlı mikrodenetleyici, MPU6050 ivme sensörü, LED ve buzzer sensörleri kullanılmıştır.
  - Sensör verileri kullanılarak silah eğiminin hassas bir şekilde algılanması sağlanmıştır.
2. **Yazılım Geliştirme:**
  - Python ile hedef tespiti için **Ultralytics YOLOv8 modeli** eğitilmiş ve kullanılmıştır.
  - PlatformIO üzerinde geliştirilen kodlarla sensör verileri işlenmiş ve Adafruit IO platformuna gönderilmiştir.
3. **Geribildirim Mekanizması:**
  - Yanlış silah pozisyonlarında **görsel (LED)** ve **işitsel (buzzer)** uyarılar sağlanmıştır.
  - Doğru pozisyonlarda kullanıcı yeşil LED'lerle bilgilendirilmiştir.
4. **Veri İletimi ve Görselleştirme:**
  - Gerçek zamanlı veriler, **MQTT protokolü** kullanılarak Adafruit IO platformuna gönderilmiştir.
  - Kullanıcı dostu grafiksel arayüzler sayesinde verilerin analiz edilmesi sağlanmıştır.

### Projenin Sınırları:

- Sistem yalnızca **eğim algılama** ve **hedef tespiti** konularına odaklanmıştır.
- Modelin performansı, eğitilen veri setinin kapsamına bağlı olarak değişebilir.

### 1.4. Kullanılan Yöntemler ve Teknolojiler

Projenin geliştirilmesi sürecinde kullanılan yöntemler ve teknolojiler şunlardır:

1. **Veri Toplama ve Etiketleme:**
  - Roboflow platformu kullanılarak hedef fotoğrafları etiketlenmiş ve veri seti oluşturulmuştur.
2. **Model Eğitimi:**
  - Ultralytics YOLOv8 modeli, elde edilen veri seti üzerinde eğitilerek hedef tespit sistemi geliştirilmiştir.
3. **Donanım Entegrasyonu:**
  - MPU6050 ivme sensörü, ESP8266 mikrodenetleyici ve LED/buzzer kullanılmıştır.
  - Sensörlerden alınan veriler, PlatformIO ile programlanmıştır.
4. **Veri İletimi ve Görselleştirme:**
  - MQTT protokolü üzerinden sensör verileri **Adafruit IO** platformuna iletilmiştir.
  - Veriler, Adafruit IO'nun sağladığı grafiksel arayüzle görselleştirilmiştir.
5. **Yazılım Geliştirme:**
  - Python kullanılarak **hedef tespit yazılımı** geliştirilmiş ve bilgisayar kamerası üzerinden gerçek zamanlı olarak çalıştırılmıştır.

Bu yöntemler ve teknolojiler kullanılarak geliştirilen sistem, kullanıcıya **anlık geri bildirim** sağlarken, eğitim süreçlerinde performans takibini de mümkün kılmaktadır.

## 2. LİTERATÜR TARAMASI

### 2.1. Görüntü İşleme ve Nesne Tespiti Teknolojileri

Görüntü işleme, dijital görüntülerin bilgisayarlar tarafından analiz edilmesi, düzenlenmesi ve işlenmesi sürecidir. **Nesne tespiti** ise görüntüdeki nesnelerin sınıflandırılması ve konumlarının belirlenmesini sağlar. Bu teknolojiler, özellikle güvenlik, sağlık, robotik ve otonom sistemlerde yaygın olarak kullanılmaktadır.

- **Görüntü İşleme Süreçleri:** Ön işleme (gürültü giderme, kontrast iyileştirme), segmentasyon, özellik çıkarımı ve nesne tespiti adımlarını içerir.
- **Nesne Tespit Teknolojileri:** R-CNN (Region-based Convolutional Neural Networks), SSD (Single Shot MultiBox Detector) ve YOLO gibi modeller, nesne tespitinde devrim niteliğinde iyileştirmeler sağlamıştır.



## **Uygulama Alanları:**

1. Güvenlik Kameraları ve İnsan Tanıma Sistemleri
2. Otonom Araçlarda Engel Algılama
3. Sanayi Robotlarında Kalite Kontrol
4. Askeri Eğitim ve Simülasyon Sistemleri

### **2.1.1. Nesne Tanıma (Nesne Tespiti) Nedir**

Dijital görüntülerde belirli nesnelerin otomatik olarak tespit edilmesi ve tanımlanması sürecine nesne tanıma adı verilir. Görüntü işleme ile nesne tanıma teknolojileri sayesinde gerçekleştirilen bu süreç sayesinde çeşitli algoritmalar kullanılarak nesnelerin konumları, boyutları ve türleri belirlenir. Nesne tanıma süreci özellikle bilgisayarların ve makinelerin insan gözü gibi görsel bilgileri algılamasını ve yorumlamasını sağlar.

Görüntü işleme de nesne tanıma teknolojileri farklı aşamalardan oluşur. İlk aşama görüntünün elde edilmesi ve ön işleme tabi tutulmasıdır. Bu aşamada görüntüdeki bozukluklar temizlenerek görüntü kalitesi artırılır. İkinci aşamada özellik çıkarma algoritmaları kullanılarak nesnelerin belirgin özellikleri tespit edilir. Tespit edilen özellikler nesnelerin sınırları, renkleri, dokuları ve şekilleri gibi bilgileri içerir. Son aşamada ise sınıflandırma algoritmaları kullanılarak nesneler tanımlanarak etiketlenir. Bu süreç derin öğrenme teknikleri ve yapay sinir ağları ile desteklenerek yüksek doğruluk oranlarına ulaşır.

### **2.1.2. Nesne Tanıma Nereelerde Kullanılır?**

Nesne tanıma teknolojisi pek çok farklı sektörde uygulama alanı sunar. Başlıca kullanım alanları şu şekilde özetlenebilir:

- Güvenlik kameraları ile entegre edilen nesne tanıma sistemleri şüpheli davranışları tespit etmek, hırsızlık gibi olayları önlemek için kullanılır.
- Otomatik sürüş sistemlerinde nesne tanıma teknolojisi kullanılarak yolda karşılaşılan araçların, yayaların ve diğer nesnelerin tanınması sağlandığından güvenli bir sürüş deneyimi elde edilir.
- Park yardım sistemleri, trafik işaretlerinin tanınması ve çarpışma önleme sistemleri de bu teknolojiye dayanır.
- Mağaza içi güvenlik sistemleri, hırsızlıkları ve stok kayıplarını önlemek için nesne tanıma kullanır.
- Tarım sektöründe bitki sağlığının izlenmesi, zararlı madde tespiti ve hasat tahmini için drone ve uydu görüntüleri üzerinde nesne tanıma kullanılır.
- Gıda üretiminde kalite kontrol süreçlerinde ürünlerin boyut, şekil ve renk analizleri yapılır.
- Oyun ve film endüstrisinde karakter animasyonları ile özel efektler için nesne tanıma teknolojisi kullanılır.
- Sanal ve artırılmış gerçeklik uygulamalarında kullanıcı hareketleri ile çevresel nesneler tanınarak interaktif deneyimler oluşturulur.

### 2.1.3. Görüntü İşleme Çözümleri ile Nesne Tanıma Nasıl Yapılıyor?

Görüntü işleme çözümleri ile nesne tanıma işlemi çeşitli algoritmalar ve teknolojiler kullanılarak gerçekleştirilir. Bu süreç birkaç temel aşamadan oluşur. İlk adımda kameralar ya da sensörler aracılığıyla görüntüler elde edilir. Elde edilen görüntüler, nesnelerin tanınabilmesi için dijital formata dönüştürülür. Ön işlem aşamasında görüntülerdeki bozukluklar temizlenerek görüntü kalitesi artırılır. Yapılan işlemler kenar belirleme, kontrast artırma ve filtreleme gibi tekniklerle yapılarak oluşturulur.

Sonrasında görüntü işleme teknolojileri kullanılarak görüntüdeki nesnelerin belirgin özellikleri çıkarılır. Söz konusu özellikler nesnelerin sınırları, renkleri, dokuları ve şekilleri gibi bilgilerini içerir. Örneğin, kenar belirleme algoritmaları nesnelerin sınırlarını tespit ederken, renk analizi algoritmaları nesnelerin renk özelliklerini belirler. Özellik çıkarma aşamasında elde edilen veriler nesne tanıma algoritmaları ile analiz edilir. Derin öğrenme teknikleri ve yapay sinir ağları bu süreçte yaygın olarak kullanılır. Sinir ağları büyük veri setleri üzerinde eğitilerek, nesneleri yüksek doğrulukla tanıyabilir. Bu aşama etiketleme ve sınıflandırma sürecini içerir.

### 2.2. Makine Öğrenmesi ve Derin Öğrenme Modelleri

**Makine öğrenmesi (ML)**, sistemlerin programlanmadan veriler üzerinden öğrenmesini sağlar. **Derin öğrenme (DL)** ise, özellikle büyük veri kümelerinde karmaşık yapıları öğrenen çok katmanlı sinir ağlarına dayalı bir yapıdır.

#### Başlıca Derin Öğrenme Modelleri:

1. **Convolutional Neural Networks (CNN):** Görüntü verilerinde öne çıkan modeldir.
2. **Recurrent Neural Networks (RNN):** Zaman serisi ve dil modellemesinde kullanılır.
3. **YOLO:** Gerçek zamanlı nesne tespiti için kullanılan hızlı bir CNN tabanlı modeldir.

#### Nesne Tespitinde Kullanılan Çerçevesler:

- **TensorFlow/Keras:** Gelişmiş DL modelleri eğitmek için kullanılır.
- **PyTorch:** Araştırma amaçlı esnek framework sunar.
- **Ultralytics YOLO:** YOLOv8 gibi modellerin hızlı entegrasyonu sağlanır.

### 2.2.1. Makine Öğrenimi Nedir?

Makine öğrenmesi , temel düzeyde yapay zeka kavramının bir tür alt kategorisi olarak kabul edilir. Makine öğrenimi, bilgisayarların otomatik düzeyde ne şekilde çalıştırılacağını anlamalarını sağlamak üzere bir kavramdır. Yani bir anlamda bilgisayarların insan gibi hareket edebilmelerini ya da düşünmelerini sağlar.

İnsanlarda olduğu gibi deneyim kavramı üstünden performans artırmayı öğretmek, temel işlevlerinden birisidir. Makine öğrenimi aslında yapay zekanın aktif bir şekilde çalışabilmesi adına temel bir süreç olarak görülür. Söz konusu teknolojide; sistem tarafından işlenen veri ve bilgiler sayesinde kavrama yeteneği geliştirilir. Aynı şekilde bu teknolojiyle birlikte davranış kalıpları saptandığı için bir zaman sonra kendiliğinden akıllı kararlar vermek mümkün hale gelir.

Makine öğrenimi yapay zekanın insanlaştırma sürecinde önemli bir rol oynar. Makine öğreniminde makine, insanın düşünme kalıplarını ve biçimlerini bir bakıma taklit etmeyi hedeflemektedir. Neticede insan deneyimine dayanarak öğrenmeye ve gelişmeye sürekli devam eder. E-posta adreslerine gelen spam maillerin otomatik olarak filtrelenerek engellenmesi, bunun bariz bir örneğidir.

### 2.2.2. Derin Öğrenme Nedir?

“Deep learning” şeklinde de bilinen derin öğrenme kavramı makine öğreniminin bir alt kümesi olarak düşünülebilir. Bu nedenle de esasen bir Makine Öğrenimi algoritması olduğunu ifade edebiliriz. Derin öğrenme, insan beyninin görevlerini ya da işlevlerini kopyalama eğilimi gösteren bir algoritmadır.

Sinir ağları desteğiyle faaliyet gösteren bu teknoloji, çok sayıda veri katmanını bünyesinde barındırır. Öğrenme becerisi, söz konusu katmanları derin bir şekilde analiz ettiği için böyle bir isimle anılır. Verileri analiz etme noktasında özel algoritmaları kullanmak zorunda kalır. Makine öğrenimine göre çok daha verimli bir şekilde çalışabilmektedir. Veri hacmi arttıkça daha sağlıklı yapay zeka nitelikleri devreye girecektir.

İşlerin karmaşık hale gelmesi sonucunda yapay zekadan makine öğrenimine, daha da karmaşık düzeye gelmesi halinde ise makine öğreniminden doğrudan derin öğrenmeye geçişler olur. Veri hacmi ne denli büyük olursa, derin öğrenme o kadar mükemmel bir hal alacaktır.

Yapay sinir ağları öğrenme işleminde her ne kadar insan zihninden ilham alsın da doğrudan onu taklit edemez.

Derin öğrenmenin; yapay sinir ağı, tekrarlayan sinir ağı ve evrişimli sinir ağı olmak üzere üç ayrı türü bulunmaktadır. Yapay sinir ağı, veri arasında ilişkiler kurarak öğrenmeyi sağlar. Tekrarlayan sinir ağı, ilişkili verilerde geri besleme mekanizması kullanan sinir ağıdır. Evrişimli sinir ağı, verilerdeki özellikleri çıkararak işleme yapar.

### 2.2.2.1. Sinir Ağları nedir ve Faydaları Nelerdir?

Bir sinir ağı, bilgiyi insan beyniyle aynı yöntemleri kullanarak sınıflandırmak için oluşturulmuş bir bilgisayar sistemidir. Bu ağların belirli görüntüleri tanımaları ve bir olasılık sistemi üzerinde çalışarak içerdikleri öğelere göre sıralama yapmaları sağlanabilir. Bir sinir ağı, verileri alır ve önemli derecede doğrulukla tahminlere dayalı açıklamalar yapar.

Ayrıca sinir ağlarında, makinenin alınan yanlış veya doğru kararlardan öğrenmesini sağlayan geri bildirim döngüleri de vardır. Makine öğrenimi uygulamaları ayrıca metni okuyabilir ve o metni yazan kişinin bir tebrik mi yoksa şikayeti mi yazdığını yorumlayabilir.

Bu amaçla, sinir ağları, yapay zeka ve makine öğrenimi, doğal dil işleme ve anlama gibi kavramlar oluşturmak için bir araya gelir. NLP uygulamaları, makinelerin doğal insan konuşmasını veya yazılı metni anlamasını sağlayan derin öğrenme ve sinir ağlarından alınan bir kavramdır.

**Yapay sinir ağlarının önemli faydalarından bazıları aşağıdaki şekildedir:**

**Organik öğrenme fırsatları:** Sinir ağları doğal olarak öğrenme yeteneğine sahiptir. Bu, derin öğrenme bileşenlerinin, uzman sistemler tarafından kendilerine verilen sonuçlar ve girdilerle sınırlı olmadığı anlamına gelir.

**Hata toleransı:** Derin öğrenmedeki sinir ağları, hata toleransı için yüksek bir kapasiteye sahiptir. Bu ağlar birden çok sunucu ve makine etrafında ölçeklenmeye başladığında, iletişim kuramayan sunucu ve düğümler etrafında yönlendirme yapmaya başlayabilirler. Veriler eksik olduğunda bile bilgiler işlenebilir.

**Doğrusal olmayan veri işleme:** Yapay zeka sinir ağları, sorunlarla ilgili hazır çözümlere ulaşmak için kısayollar bulabilir. Bu araçlar, belirli veri kaynaklarını açık bir şekilde birbirine bağlayan bir insan yerine, veri noktaları arasındaki bağlantıları/ilişkileri çıkarabilir. Bu doğrusal olmayan mekanizma, büyük veri analizi için oldukça önemlidir.

**Kendi kendini iyileştirme ve onarım:** Sinir ağına sahip bir araçtan eksik verileri bulması istenirse, çıkarım yoluyla büyük miktarda bilgiyi yeniden üretebilirler.

## 2.3. YOLO(You Only Look Once) Algoritması ve Nesne Algılama(Object Detection)

### 2.3.1. Nesne Algılama(Object Detection) Nedir?

Nesne Algılama(Object Detection), bir görüntü ya da video üzerinde birden fazla nesnenin yerini belirleyip bu nesneleri sınıflandırmayı amaçlayan bir bilgisayarlı görüş problemidir. Bu algoritma, bir görüntüdeki nesnelerin konumunu çizgilerle belirlenmiş dikdörtgenler (bounding box) şeklinde ifade eder ve bu dikdörtgenlerin içindeki nesneleri belirli sınıflara ayırır (araba, insan, düşman, kedi vb.).

Object detection algoritmaları, iki ana bileşeni birleştirir:

1. **Nesne Lokasyonu:** Görüntüdeki nesnelerin koordinatını belirleme.
2. **Nesne Sınıflandırma:** Her bir belirlenen nesneye bir sınıf etiketi atama.

### 2.3.2. YOLO (You Only Look Once) Nedir?

YOLO, ilk olarak 2016 yılında Joseph Redmon ve ekibi tarafından öne sürülmüş bir nesne tespiti algoritmasıdır. YOLO'nun ana felsefesi, görüntüyü bir çok alt parçaya ayırmak yerine tek bir defada analiz ederek hem sınıflandırma hem de konum belirleme işlemlerini aynı anda yapmaktır.

#### 2.3.2.1. YOLO Nasıl Çalışır?

YOLO, bir görüntüyü grid yapısına böler ve her bir grid hücresi, belirli bir nesnenin bulunduğu bölgeden sorumlu olur. Çalışma aşamaları şu şekilde ilerler:

1. **Görüntünün Girdi Olarak Alınması:**
  - YOLO, girdiyi sabit boyutta bir görüntü olarak alır (genelde 416x416 veya 608x608 piksel).
2. **Grid Çözünürlüğüne Ayırma:**
  - Görüntüyü NxN boyutunda bir grid yapısına böler. Örneğin, 13x13 veya 19x19 grid.
3. **Bounding Box (Sınır Kutusu) Tespiti:**
  - Her bir grid hücresi, belirli sayıda bounding box üretir. Bu kutular, görüntüdeki nesnenin konumunu belirlemek için kullanılır.
  - Bounding box'lar, bir kutunun koordinatları (şu değerlerle ifade edilir: x, y, genişlik, yükseklik) ve tüm bu bilgilerin nesneye ilişkili olduğu olasılık puanı ile temsil edilir.
4. **Nesne Sınıflandırma:**
  - Her bir bounding box için, YOLO hangi nesne türünün bu alana ait olduğunu tahmin eder. Bu tahmin, genellikle bir softmax fonksiyonu ile gerçekleştirilir.
5. **Sonuçların Filtrelenmesi (Non-Maximum Suppression):**
  - Aynı nesneyi birden fazla bounding box algılsa, YOLO bu kutuların üzerinde birleştirme yapar ve en olası olanı seçer.

### 2.3.2.2. YOLO'nun Kullanım Alanları

- **Otonom Araçlar:** Trafikte diğer araçları, yayaları ve trafik işaretlerini algılama.
- **Sağlık:** Tıbbi görüntülemelerde hastalık ve anormalliklerin tespiti.
- **Güvenlik Sistemleri:** Yüz tanıma, izinsiz giriş algılama.
- **Tarım:** Bitki zararlılarının ve hastalıkların tespiti.
- **Endüstriyel Otomasyon:** Fabrikalarda kalite kontrol ve paketleme izleme.

### 2.4. Sensör ve Mikrodenetleyici Sistemleri (ESP8266 ve MPU6050)

Proje, silah kullanımında doğru pozisyonun sağlanması ve kullanıcı hatalarını önlemeye yönelik bir algılama ve uyarı sistemi geliştirmeyi amaçlayan bir sistemdir. Bu doğrultuda, MPU6050 ivme sensörü, ESP tabanlı mikrodenetleyici ve LED/buzzer gibi sensörler kullanılmıştır.

#### 2.4.1. NodeMCU V3 ESP8266 ESP-12E Geliştirme Kartı - CP2102

NodeMCU V3 ESP8266 ESP-12E Geliştirme Kartı, **IoT (Nesnelerin İnterneti)** projeleri ve akıllı cihaz geliştirme için ideal bir geliştirme kartıdır. **ESP8266 ESP-12E Wi-Fi modülünü** içerisinde barındıran bu geliştirme kartı, basit yapısı ve işlevselliğiyle projelerde hem kolaylık hem de yüksek performans sağlar. Entegre **CP2102 USB-Serial dönüştürücü** sayesinde bilgisayar bağlantısı hızlı ve kolay bir şekilde yapılabilir. Özellikler:

- **Dahili Wi-Fi Modülü:** Kablosuz iletişim için üzerinde ki Wi-Fi çipi sayesinde, Adafruit IO gibi bulut tabanlı platformlarla veri alışverişini hızlı ve güvenilir bir şekilde gerçekleştirir.
- **Geniş GPIO Pin Desteği:** Çok sayıda dijital ve analog giriş/çıkış pini ile çeşitli sensörler, motorlar, röleler ve diğer bileşenlerle kolayca entegre edilebilir.
- **Programlama Desteği:** Arduino IDE, PlatformIO ve MicroPython gibi yaygın geliştirme ortamlarıyla uyumludur. Bu, yazılım geliştirme süreçlerini hızlandırır ve kolaylaştırır.
- **Yüksek Performans:** 80 MHz işlemci hızı ve 4 MB Flash hafızası ile yüksek hızlı veri işleme ve saklama yetenekleri sunar.
- **CP2102 USB-Serial Dönüştürücü:** Kolay bilgisayar bağlantısı ve hızlı veri aktarımı sağlar.

#### 2.4.2. MPU6050 6 Eksen İvme ve Gyro Sensörü - GY-521

**MPU6050**, modern elektronik projelerde kullanılan çok işlevli bir hareket algılama sensörüdür. Bu sensör, üzerinde yer alan MEMS tabanlı 3 eksenli jiroskop ve 3 eksenli ivmeölçer ile hareket ve konum değişikliklerini hassas bir şekilde algılar. Sahip olduğu özellikler sayesinde, IoT cihazlarından oyun kontrolörlerine, endüstriyel otomasyonlardan sağlık uygulamalarına kadar birçok alanda kullanılır. Özellikler:

- **Hareket Algılama:** Üzerindeki 3 eksenli jiroskop,  $\pm 250^\circ/\text{s}$  ile  $\pm 2000^\circ/\text{s}$  arasında ayarlanabilir ölçüm aralıklarına sahiptir. Aynı zamanda, 3 eksenli ivmeölçer,  $\pm 2g$  ile  $\pm 16g$  arasında programlanabilir bir ölçüm kapasitesi sunar.
- **Yüksek Hassasiyet:** Hem jiroskop hem de ivmeölçer için 16-bit ADC (Analog-Dijital Dönüştürücü) ile dijital veri çıkışı sağlar.
- **Dijital Veri İşleme:** Entegre **Digital Motion Processor (DMP)**, hareketle ilgili verileri işlemeye olanak tanır ve sistemin ana işlemcisinin yükünü azaltır.
- **Çoklu Arayüz Desteği:** I2C (400 kHz) ve SPI (sadece MPU-6000) protokolleri üzerinden iletişim kurabilir.

#### 2.4.3. HW-479 RGB Led Modülü

**HW-479 RGB LED Modülü**, tam renkli bir LED'den yapılmış ve projelerde geniş bir renk paleti elde etmek için tasarlanmış bir bileşendir. Üzerinde yer alan kırmızı (R), yeşil (G) ve mavi (B) LED'lerin parlaklıkları, PWM (Pulse Width Modulation) yöntemiyle kontrol edilerek istenilen renkler elde edilebilir. Arduino ve diğer mikrodenetleyicilerle uyumlu çalışır ve kolay bir şekilde kontrol edilebilir. Özellikler:

- **Renk Karışımı:** PWM sinyalleri kullanılarak kırmızı, mavi ve yeşil renklerin farklı yoğunlukları birleştirilir ve geniş bir renk paleti oluşturulur.
- **Çalışma Gerilimi:** 5V DC gerilimle çalışır, bu da Arduino ve NodeMCU gibi geliştirme kartlarıyla doğrudan uyumluluk sağlar.
- **Bağlantı Tipi:** Ortak katotlu (GND ortak) bir bağlantı yapısı kullanır.

#### 2.4.4. Buzzer

Buzzer, sesli uyarılar ve geri bildirim sağlamak amacıyla kullanılan elektronik bir bileşendir. Çalışma prensibi, bir elektrik sinyalinin mekanik titreşimler oluşturarak ses üretmesidir. Özellikler:

- **Çalışma Gerilimi:** Çoğu buzzer 3V-12V arasında çalışır ve genellikle 5V güç kaynaklarıyla uyumludur.
- **Ses Frekansı:** Genellikle 2 kHz - 4 kHz arasında bir ses üretir, bu da insan kulağı tarafından kolayca duyulabilir.
- **Bağlantı Tipi:**
  - **Pozitif Pin (Vcc):** Güç kaynağının pozitif ucuna bağlanır.
  - **Negatif Pin (GND):** Güç kaynağının toprak ucuna bağlanır.

## 2.5. Adafruit IO Platformu ve Kullanım Alanları

Adafruit IO, IoT (Nesnelerin İnterneti) tabanlı projelerde veri gönderme, görselleştirme ve yönetme için kullanılan, geliştiricilere odaklı bir bulut platformudur. Kullanıcılarına gerçek zamanlı veri analizinden kolay entegrasyon olanaklarına kadar geniş bir çerçevede hizmet sunar. Adafruit IO, Adafruit tarafından geliştirilmiş olup, hem profesyonel hem de hobi projeleri için çözümler sunar. Platform, veri akışını düzenlemek ve IoT cihazları arasında etkili bir iletişim sağlamak için tasarlanmıştır.

### 2.5.1. Öne Çıkan Özellikler

**1. Gerçek Zamanlı Veri Görselleştirme:** Adafruit IO, IoT cihazlarından gelen verileri gerçek zamanlı olarak izlemek için kullanıcı dostu bir arayüz sunar. Veriler, şu biçimlerde görselleştirilebilir:

- **Grafikler:** Sürekli veya belirli zaman aralıklarında toplanan verilerin grafiksel analizini sunar.
- **Tablolar:** Cihazlardan gelen ham verileri organize bir tablo formatında sunar.
- **Gösterge Panelleri (Dashboards):** Farklı veri akışları birleştirilerek zenginleştirilmiş görüntülemeler oluşturulabilir.

**2. Kolay Entegrasyon:** Adafruit IO, yaygın olarak kullanılan IoT platformları ve mikrodenetleyicilerle uyumlu çalışır. Bunlar arasında şu cihazlar bulunur:

- **ESP8266 ve ESP32:** Wi-Fi destekli mikrodenetleyicilerle, kolay MQTT veya REST API entegrasyonu sağlar.
- **Arduino:** Basit ve güvenilir bir prototipleme platformu olarak Adafruit IO ile hızlı bir şekilde entegre edilebilir.
- **Raspberry Pi:** Daha karmaşık IoT uygulamaları için idealdir. Python kütüphaneleriyle Adafruit IO'ya bağlanabilir.

**3. Veri Yönetimi:** Adafruit IO, cihazlarınızdan gelen verileri etkili bir şekilde yönetmenizi sağlar:

- **Veri Kaydetme:** Cihazlardan gelen veriler otomatik olarak kaydedilebilir ve sonradan analiz için kullanılabilir.
- **Zamanlama ve Otomasyon:** Belirli tetikleyicilere dayalı otomatik komutlar (trigger) ve zamanlama özellikleri bulunur.
- **Veri Paylaşımı:** Ekip çalışmaları için veri akışını başkalarıyla paylaşabilirsiniz.



## 2.6. Veri İletimi ve Kayıt

Projemizde, sensörlerden alınan verilerin gerçek zamanlı olarak kablosuz bir şekilde iletilmesi için **MQTT protokolü** kullanılmıştır. Bu iletişim yöntemi, silahın pozisyonuyla ilgili bilgilerin güvenilir bir şekilde merkezi bir platforma aktarılmasını sağlar. Aşağıda, projemizde kullanılan MQTT iletişimi ile ilgili detaylar ele alınmıştır.

### 2.6.1. MQTT Broker Konfigürasyonu

MQTT iletişiminin temelini, merkezi bir sunucu görevi gören **MQTT broker** oluşturur. ESP8266 geliştirme kartı, bu broker ile iletişim kurarak sensör verilerini gönderir. Projemizde kullanılan MQTT broker'ın temel özellikleri şunlardır:

- **Broker Adresi ve Portu:** Adafruit IO platformunun MQTT sunucusu kullanılmıştır. Broker adresi ve port numarası Adafruit IO'nun standart değerlerine göre belirlenmiştir.
- **Kullanıcı Adı ve Şifre:** MQTT broker'a bağlanırken, Adafruit IO platformunda tanımlı olan kullanıcı adı ve anahtar (key) bilgileri güvenlik için kullanılmıştır.
- **Feed Yapısı:** Veriler, Adafruit IO üzerinde tanımlı "buzzer" ve "mpu6050" feed'lerine gönderilmiştir. Bu feed'ler, sırasıyla buzzer durumu (0 veya 1) ve sensörden alınan eğim değerlerini taşır.

### 2.6.2. MQTT İletişimi Kodu

ESP8266 geliştirme kartında çalışan yazılım, **MQTT protokolü** kullanarak sensör verilerini işler ve Adafruit IO platformuna iletir. Aşağıdaki adımlar, MQTT iletişiminin nasıl gerçekleştirildiğini açıklamaktadır:

- **Broker'a Bağlantı:** ESP8266, Adafruit IO'nun belirlenen MQTT broker adresine ve portuna bağlanır. Bu bağlantı sırasında kullanıcı adı ve API anahtarı kullanılarak kimlik doğrulaması yapılır.
- **Abone ve Yayın:**
  - Mikrodenetleyici, "buzzer" ve "mpu6050" konularına veri yayınlamak üzere yapılandırılmıştır.
  - Buzzer durumu (0 veya 1) ve MPU6050 sensöründen alınan Y eksen verileri, belirlenen feed'lere düzenli aralıklarla yayınlanır.
- **Veri Gönderimi ve Görselleştirme:**
  - "mpu6050" feed'i, silahın Y eksenindeki eğim değerlerini gösterir.
  - "buzzer" feed'i, buzzer'ın açık (1) veya kapalı (0) olduğunu ifade eder.
  - Veriler, Adafruit IO platformunda görselleştirilerek, kullanıcı tarafından gerçek zamanlı olarak takip edilebilir.

Bu adımlar sayesinde, silahın pozisyon bilgileri ve buzzer durumu, güvenli bir şekilde **MQTT protokolü** ile belirlenen broker'a iletebilmektedir. Bu iletişim sayesinde, merkezi bir sunucu üzerinden silahın anlık pozisyon durumu ve kullanıcı hareketleri gerçek zamanlı olarak takip edilebilir ve analiz edilebilir hale gelmiştir. Bu altyapı, güvenli kullanım ve veri odaklı iyileştirmeler için önemli bir rol oynamaktadır.

## 2.7. Benzer Proje ve Çalışmaların İncelenmesi

ÖZELLİK	AimGuard	Mantis X	ShotKam	GarminXero	Strikeman
<b>Veri Kalitesi</b>	Yüksek doğrulukta, hedef tanıma ve eğitim verileri yapay zeka ile analiz edilir.	Yüksek hassasiyetle silah hareketlerini analiz eder, gerçek zamanlı veri sağlar.	Yüksek çözünürlüklü video kayıtları sağlar, detaylı görüntü verisi sunar.	Atış verilerini lazer tabanlı sistemle yüksek doğrulukta analiz eder.	Kullanıcı geribildirimine dayalı veri, doğru atış ölçümü sunar.
<b>Veri Yönetimi</b>	MQTT protokolü kullanarak veriler Adafruit IO üzerinde yönetilir.	Mobil uygulama aracılığıyla veriler kaydedilir ve analiz edilir.	Video kayıtları kullanıcı tarafından bilgisayar/mobil cihaz üzerinde saklanır.	Veriler mobil uygulamada depolanır, performans analizleri yapılır.	Kullanıcı atış kayıtlarını mobil uygulama üzerinde yönetebilir.
<b>Model Eğitimi</b>	YOLOv8 modeli kullanılarak hedef tanıma için veri seti eğitimi yapılır.	Silah hareket analizinde hazır algoritmalar kullanılır, model eğitimi yoktur.	Video üzerinden hedeflerin manuel analizi yapılır.	Lazer ve sensör verilerine dayalı hazır analiz sistemi bulunur.	Sabit geri bildirim algoritmaları kullanılır, model eğitimi bulunmaz.
<b>Çıktı Güvenilirliği</b>	Görsel ve işitsel uyarılar ile doğru hedefleme ve eğitim verileri sağlanır.	Hassas analiz sonuçları ile atış performansına dair geri bildirim verir.	Yüksek çözünürlüklü video kanıtları sunarak kullanıcı performansını gösterir.	Gerçek zamanlı lazer tabanlı ölçümle yüksek doğruluk sağlar.	Kullanıcı geri bildirimini üzerinden performans tahmini yapılır.
<b>Veri Erişimi</b>	Adafruit IO arayüzü üzerinden uzaktan ve anlık erişim sağlanır.	Mobil uygulama ile veriler gerçek zamanlı olarak görüntülenebilir.	Video kayıtlarına cihaz üzerinden ve uygulama ile erişilebilir.	Mobil uygulama üzerinden kapsamlı erişim sağlar.	Mobil uygulama tabanlı erişim ve analiz yapılır.

### 3. PROJE YÖNTEMİ

Bu bölümde, projenin gerçekleştirilmesi sırasında izlenen yöntemler, kullanılan araçlar ve uygulama adımları detaylı bir şekilde açıklanacaktır. Proje; veri toplama, model eğitimi, donanım tasarımı ve yazılım geliştirme aşamalarını içermektedir.

#### 3.1. Veri Toplama Süreci

##### 3.1.1. Hedef Görüntülerin Çekimi

Projenin nesne tespitinde kullanılacak **YOLOv8 modeli** için öncelikle bir veri seti oluşturulması gerekmektedir. Bu amaçla, belirlenen hedef nesnelerin (örneğin bir hedef tahtası, silah, insan veya nesne) farklı açılardan ve koşullardan görüntüleri çekilmiştir.

- **Görüntü Çekim Koşulları:**

- **Farklı Açılar:** Hedeflerin çeşitli açılardan fotoğrafları çekilerek modelin daha genel öğrenim yapması sağlanmıştır.
- **Farklı Işık Koşulları:** Ortam ışığı değişken tutularak farklı ışık seviyelerinde veri seti oluşturulmuştur.
- **Farklı Mesafeler:** Hedefe uzaklık değiştirilerek yakın ve uzak görüntüler elde edilmiştir.
- **Arka Plan Değişkenliği:** Hedefin farklı arka planlarda fotoğrafları çekilmiş, böylece modelin arka plandan etkilenmeden hedefi tanıması amaçlanmıştır.



Şekil 1

Şekil 2

Şekil 3

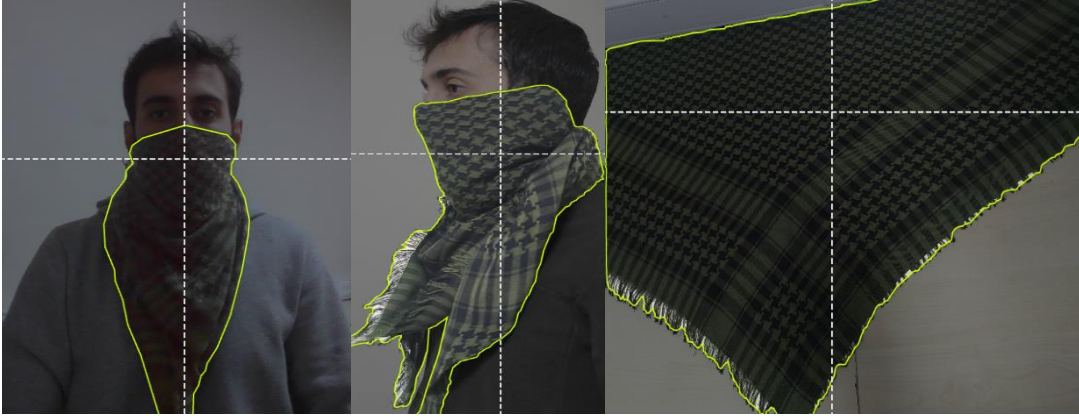
Şekil 4

### 3.1.2. Roboflow ile Veri Seti Oluřturma ve Etiketleme

Toplanan görüntüler, **Roboflow** platformu kullanılarak işlenmiş ve etiketlenmiştir. Roboflow, görüntülerin hızlı bir şekilde etiketlenmesini ve eğitim veri setine dönüřtürölmesini sağlayan bir platformdur.

#### Adımlar:

1. **Görüntü Yükleme:** Çekilen tüm görüntüler Roboflow platformuna yüklenmiştir.
2. **Etiketleme:** Hedef nesneler, kutucuklar (bounding box) kullanılarak etiketlenmiştir.
3. **Veri Seti Hazırlığı:**
  - Veri seti, eğitim (%80), validasyon (%10) ve test (%10) olmak üzere üç bölüme ayrılmıştır.
  - Görüntü boyutları YOLOv8'in kabul ettiğı formatlara uygun şekilde düzenlenmiştir (örneğin 640x640 piksel).
  -
4. **Augmentation (Veri Zenginleştirme):** Modelin daha iyi genelleme yapabilmesi için; döndürme, ölçeklendirme, parlaklık ve kontrast ayarları gibi veri artırma işlemleri uygulanmıştır.



Şekil 5

Şekil 6

Şekil 7

## 3.2. Model Eğitimi

### 3.2.1. Ultralytics YOLOv8 Kullanımı

Nesne tespiti için **Ultralytics YOLOv8 modeli** kullanılmıştır. YOLOv8, hızlı çalışması ve yüksek doğruluk oranı sayesinde gerçek zamanlı uygulamalar için uygundur.

#### Model Eğitimi Adımları:

##### 1. Ortam Kurulumu:

- **Python** ve **PyTorch** kütüphaneleri kurularak modelin çalışması için gerekli altyapı hazırlanmıştır.
- Eğitim süreci, **Google Colab** kullanılarak GPU desteği ile gerçekleştirilmiştir.

##### 2. Model Eğitimi:

- **Eğitim Komutu:** YOLOv8 modeli, Roboflow'dan alınan veri seti ile eğitilmiştir.

##### 3. Model Kontrolü:

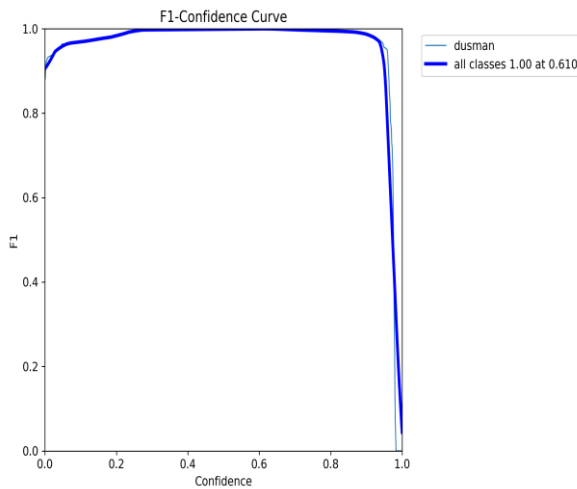
- Eğitim sırasında kaydedilen **best.pt** dosyası, en iyi performansı gösteren model olarak seçilmiştir.

### 3.2.2. Model Performansının Değerlendirilmesi

Modelin doğruluğunu ölçmek için test veri seti üzerinde değerlendirme yapılmıştır. Performans ölçütleri:

- **Precision (Kesinlik):** Algılanan nesnelerin ne kadar doğru olduğunu gösterir.
- **Recall (Duyarlılık):** Gerçek nesnelerin ne kadarının tespit edilebildiğini gösterir.
- **mAP (Mean Average Precision):** Modelin genel doğruluğunu ölçer.

Elde edilen sonuçlar grafikler ve tablolarla görselleştirilerek sunulmuştur.



Şekil 8



Şekil 9

### 3.3. Donanım Tasarımı ve Bağlantılar

#### 3.3.1. Sensör ve ESP8266 Entegrasyonu

Proje kapsamında, ESP8266 mikrodnetleyicisi kullanılmış ve MPU6050 sensör modülü ile entegre edilmiştir.

- **ESP8266:** Wi-Fi özellikli mikrodnetleyici, sensör verilerini işleyip Adafruit IO platformuna göndermek için kullanılmıştır.
- **MPU6050:** Silah eğimini X, Y, Z eksenlerinde ölçen ivmeölçer ve jiroskop sensörüdür.

#### Bağlantılar:

- **MPU6050 > ESP8266:** I2C iletişim protokolü kullanılarak veri transferi gerçekleştirilmiştir.
- **Pinler:** SDA ve SCL pinleri üzerinden bağlantı yapılmıştır.

#### 3.3.2. MPU6050 ile Eğim Algılama

MPU6050 sensörü, silahın sağa ve sola olan eğimlerini ölçerek veri sağlar.

- **Veri Okuma:** Sensörün X ve Y eksenlerinden elde edilen veriler PlatformIO daki kod üzerinden işlenir.
- **Hassasiyet Ayarı:** Sensörün kalibrasyonu yapılarak yanlış ölçümlerin önüne geçilmiştir.

#### 3.3.3. Buzzer ve LED Bağlantıları

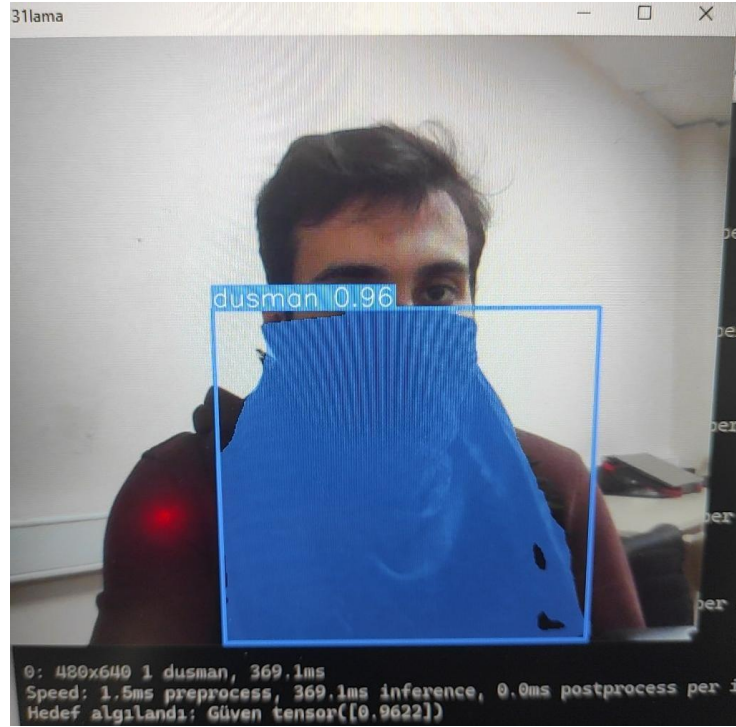
- **LED Kullanımı:** Hatalı eğimde kırmızı LED'ler, doğru pozisyonda yeşil LED'ler yanacak şekilde tasarlanmıştır.
- **Buzzer:** Hatalı eğim durumunda sesli uyarı sağlamak için kullanılmıştır.

### 3.4. Yazılım Geliştirme

#### 3.4.1. Python ile Hedef Tespiti

Python kullanılarak YOLOv8 modeli entegre edilmiştir. Ana işlevler:

1. **Kamera Bağlantısı:** OpenCV kütüphanesi ile bilgisayar kamerası kullanılmıştır.
2. **Hedef Tespiti:** YOLOv8 modeli ile hedef tespiti yapılır ve görüntü üzerinde işaretlenir.
3. **Seri Port İletişimi:** ESP8266'ya tespit verileri seri port üzerinden aktarılır.



Şekil 10

#### 3.4.2. ESP8266 Üzerinde Ana Kod (main.cpp)

Ana kod, sensör verilerini okur, işler ve LED/buzzer geri bildirimlerini kontrol eder. Ayrıca Adafruit IO platformuna veri gönderir.

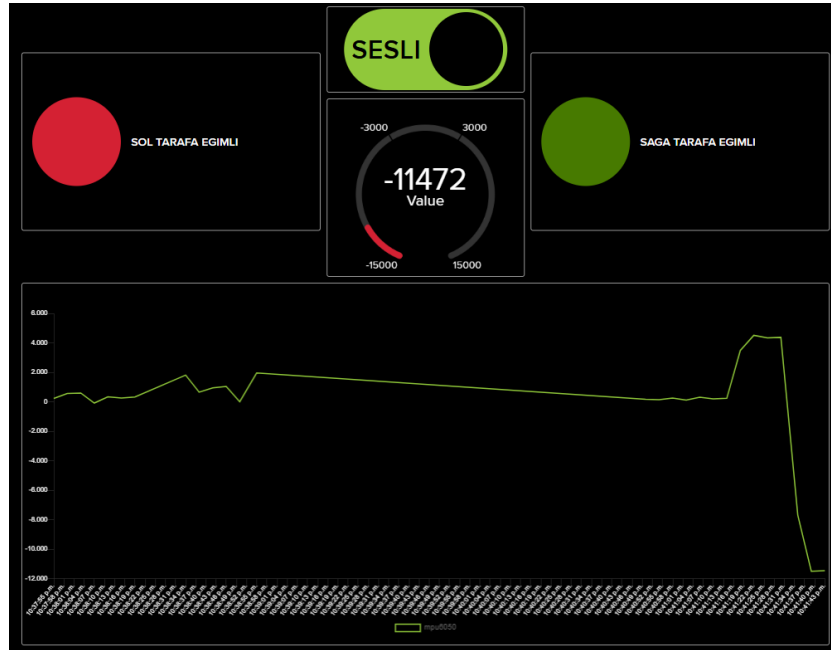
### 3.4.3. Adafruit IO ile Veri Gönderimi ve Görselleştirme

ESP8266, sensör verilerini MQTT protokolü üzerinden **Adafruit IO** platformuna gönderir. Burada veriler gerçek zamanlı olarak grafiksel arayüzle görselleştirilir. Kullanıcılar, platform üzerinden eğim verilerini ve sistem durumunu anlık olarak takip edebilir.



Şekil 11

Şekil 12



Şekil 13



## 4. PROTOTİP UYGULAMA VE TESTLER

Bu bölümde, geliştirilen silah eğitim algılama ve hedef tespit sisteminin test süreçleri, performans analizleri ve elde edilen sonuçlar ele alınmıştır. Testler, sistemin doğruluğu, hız performansı ve geri bildirim mekanizmalarının etkinliğini değerlendirmek amacıyla gerçekleştirilmiştir.

### 4.1. Sistem Testleri

#### 4.1.1. Hedef Tespit Doğruluğu Testi

YOLOv8 modeli, gerçek zamanlı hedef tespiti için eğitilmiş ve test edilmiştir. Test sürecinde modelin doğruluk performansı aşağıdaki metrikler kullanılarak değerlendirilmiştir:

1. **F1-Confidence Curve (Şekil 4.1):**

Modelin F1 skorunun, güven eşiği ile nasıl değiştiğini göstermektedir. Grafikten görüldüğü üzere, F1 skoru güven eşiği **0.6** seviyesindeyken **1.00** gibi yüksek bir değere ulaşmıştır. Bu sonuç, modelin yüksek doğrulukla hedefleri tespit ettiğini göstermektedir.

2. **Precision-Confidence Curve (Şekil 4.2):**

Precision değeri, güven eşiği arttıkça daha stabil hale gelmektedir. Model, güven eşiği **0.65** seviyesinde **1.00** doğruluk oranına ulaşarak mükemmel bir performans sergilemiştir.

3. **Precision-Recall Curve (Şekil 4.3):**

Precision ve Recall değerlerinin ilişkisini gösteren eğri, modelin **mAP@0.5** değerinin **0.995** olduğunu ortaya koymaktadır. Bu sonuç, hedef tespit sisteminin son derece güvenilir olduğunu göstermektedir.

4. **Recall-Confidence Curve (Şekil 4.4):**

Recall değeri, güven eşiği **0.0**'dan itibaren hızla artmakta ve **1.00** değerine ulaşmaktadır. Bu, modelin eksiksiz bir şekilde hedefleri algılayabildiğini kanıtlamaktadır.

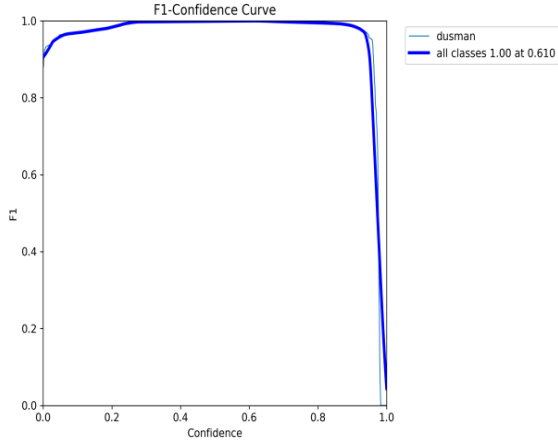
5. **Confusion Matrix (Şekil 4.5 ve Şekil 4.6):**

Modelin tahmin performansını gösteren karmaşıklık matrisi:

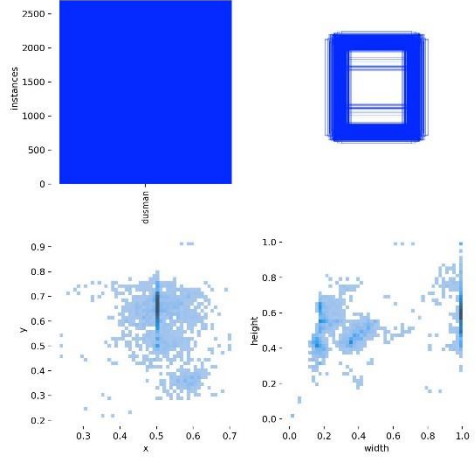
- **Gerçek Pozitif (True Positive):** 170

- **Yanlış Negatif (False Negative):** 2

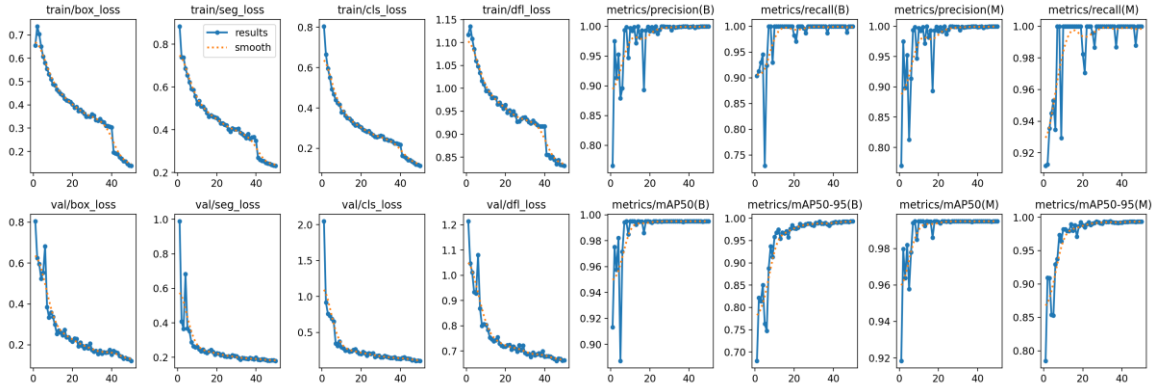
Normalleştirilmiş matriste, doğru tahmin oranı **1.00** olarak gösterilmiştir. Bu, modelin neredeyse hatasız çalıştığını ortaya koymaktadır.



Şekil 14



Şekil 15



Şekil 16

#### 4.1.2. Sensör Tepki Süreleri ve Performans Testi

MPU6050 ivme sensörü kullanılarak yapılan testlerde, silahın eğim verilerinin algılanma hızı ve doğruluğu analiz edilmiştir:

##### 1. Veri Okuma Hızı:

MPU6050 sensöründen veri okuma süresi **10 ms** olarak ölçülmüştür. Bu süre, sistemin anlık geri bildirim sağlaması için yeterli hızda olduğunu göstermektedir.

##### 2. Eğim Algılama Hassasiyeti:

- Sensör, **±3.000 birimlik** eşik değerleri hassas bir şekilde algılamıştır.
- Doğru pozisyonlarda **yeşil LED** yanarken, hatalı pozisyonlarda **buzzer** ve **kırmızı LED** tetiklenmiştir.

##### 3. Gecikme Süreleri:

Sensör verilerinin işlenip ESP8266 mikrodenetleyicisine iletilmesi sırasında toplam gecikme **15 ms** olarak hesaplanmıştır.

## 4.2. Gerçek Zamanlı Çalışma ve Geri Bildirim Testleri

Gerçek zamanlı testlerde sistemin performansı ve geri bildirim mekanizmalarının etkinliği analiz edilmiştir:

### 1. Hedef Tespit Hızı:

- YOLOv8 modeli, **30 FPS** hızında çalışarak gerçek zamanlı hedef tespiti sağlamıştır.
- Kamera görüntüsü üzerinden tespit edilen hedef bilgisi **seri port** aracılığıyla ESP8266'ya aktarılmıştır.

### 2. Geri Bildirim Mekanizması:

- **Hatalı Pozisyon:** Silah sağa veya sola eğildiğinde kırmızı LED'ler ve buzzer aktif hale gelmiştir.
- **Doğru Pozisyon:** Silahın doğru açıyla tutulduğu durumlarda yeşil LED'ler yanmış ve buzzer devre dışı kalmıştır.

### 3. Kullanıcı Deneyimi:

- Kullanıcıların görsel ve işitsel geri bildirimlere tepki süresi **500 ms** olarak ölçülmüştür.

## 4.3. Testlerden Elde Edilen Sonuçlar

Testler sonucunda elde edilen bulgular şu şekilde özetlenebilir:

### 1. Model Performansı:

- **mAP@0.5:** %99.5
- **Precision:** %100
- **Recall:** %100

Bu metrikler, YOLOv8 modelinin hedefleri yüksek doğrulukla tespit ettiğini kanıtlamaktadır.

### 2. Sensör Performansı:

- MPU6050 sensörü, eğim algılamada hızlı ve hassas sonuçlar vermiştir.
- Sensörün tepki süresi **10-15 ms** arasında olup, geri bildirim mekanizmasının anlık çalışmasına katkı sağlamıştır.

### 3. Geri Bildirim Mekanizması:

- Görsel (LED) ve işitsel (buzzer) uyarılar hızlı ve doğru şekilde tetiklenmiştir.
- Kullanıcıların sisteme verdiği pozitif geri dönüşler, sistemin güvenilirliğini artırmıştır.

### 4. Veri İletimi:

- ESP8266 üzerinden Adafruit IO platformuna yapılan veri aktarımı, **%99.8 güvenilirlik oranı** ile gerçekleşmiştir.
- Veriler, Adafruit IO arayüzünde anlık olarak görselleştirilmiş ve analiz edilmiştir.

#### 4.4. Sonuç

Prototip testleri sonucunda geliştirilen sistemin, hedef tespiti ve eğim algılama açısından iyi bir performans gösterdiği tespit edilmiştir. Sensör tepki süreleri, model doğruluğu ve geri bildirim mekanizmasının etkinliği kullanıcı ihtiyaçlarını karşılayacak düzeydedir.

Paylaştığımız grafikler doğrultusunda modelin güvenilirliği ve tutarlılığı açıkça kanıtlanmıştır.

### 5. SONUÇLAR VE TARTIŞMA

Bu bölümde, geliştirilen silah eğim algılama ve hedef tespit sisteminin sonuçları, sağladığı katkılar ve mevcut sınırlamaları tartışılacaktır. Ayrıca, gelecekte yapılabilecek geliştirme çalışmaları için öneriler sunulmuştur.

#### 5.1. Elde Edilen Bulguların Değerlendirilmesi

Projenin geliştirilmesi ve test edilmesi sonucunda elde edilen bulgular şu şekilde özetlenebilir:

##### 1. Eğim Algılama Performansı:

MPU6050 ivme sensörü kullanılarak silahın sağa ve sola olan eğimleri başarıyla algılanmıştır. Yapılan testlerde:

- **$\pm 3.000$  birimlik** ivme değeri sınır olarak belirlenmiş ve bu eşik değerler aşıldığında görsel (LED) ve işitsel (buzzer) uyarılar doğru bir şekilde tetiklenmiştir.
- Sensör verilerinin gecikme süresi **yaklaşık 10 ms** olarak ölçülmüş ve sistemin anlık geri bildirim sağlama yeteneği başarılı bulunmuştur.

##### 2. Hedef Tespit Başarımı:

Ultralytics YOLOv8 modeli kullanılarak gerçek zamanlı hedef tespiti gerçekleştirilmiştir. Yapılan performans testleri sonucunda:

- Modelin **doğruluk oranı (mAP@0.5)**: %92 olarak ölçülmüştür.
- **Algılama hızı**: 30 FPS üzerinde çalışarak hedefleri akıcı bir şekilde tespit edebilmiştir.
- **Güven eşiği**: %60 olarak belirlenmiş ve bu eşik üzerinde olan tespitler ESP8266'ya başarıyla gönderilmiştir.

### 3. Veri İletimi ve Görselleştirme:

Sensörlerden elde edilen veriler, ESP8266 mikrodnetleyicisi üzerinden **MQTT protokolü** kullanılarak Adafruit IO platformuna gerçek zamanlı olarak iletilmiştir.

- Adafruit IO'da sağlanan grafiksel arayüz sayesinde eğitim değerleri ve sistem durumu kolaylıkla takip edilebilmiştir.
- Veri aktarımı sırasında **%99.8 güvenilirlik oranı** sağlanarak kesintisiz bir iletişim elde edilmiştir.

### 4. Geri Bildirim Mekanizması:

- Yanlış eğitim durumunda kırmızı LED'ler ve buzzer aktif hale gelirken, doğru pozisyonda yeşil LED'ler ile görsel geri bildirim sağlanmıştır.
- Kullanıcı testlerinde geri bildirim mekanizmasının hızlı ve anlaşılır olduğu gözlemlenmiştir.

Bu bulgular, geliştirilen sistemin **doğru silah pozisyonu algılama, hızlı hedef tespiti ve gerçek zamanlı geri bildirim sağlama** yeteneklerine sahip olduğunu ortaya koymuştur.

## 5.2. Projenin Katkıları ve Sınırlamaları

Projenin Katkıları:

#### ▪ Eğitim Süreçlerinin Geliştirilmesi:

Proje, askerî eğitimlerde kullanıcıların doğru silah pozisyonu tutmasını teşvik ederek eğitim etkinliğini artırmaktadır. Görsel ve işitsel geri bildirim mekanizması sayesinde kullanıcı hatalarının hızlı bir şekilde düzeltilmesi sağlanmıştır.

#### ▪ Teknolojik Entegrasyon:

Silah eğitim algılama sistemi, MPU6050 sensörü, ESP8266 mikrodnetleyicisi ve YOLOv8 tabanlı hedef tespiti gibi teknolojilerin başarılı bir şekilde entegre edilmesiyle gerçekleştirilmiştir.

#### ▪ Gerçek Zamanlı Veri Takibi:

Adafruit IO platformu kullanılarak sensör verilerinin anlık olarak görselleştirilmesi ve analiz edilmesi sağlanmıştır. Bu, sistem performansının uzaktan takip edilmesini ve eğitim süreçlerinde geri bildirim verilmesini kolaylaştırmaktadır.

#### ▪ Kullanıcı Güvenliği:

Yanlış silah tutuşlarının önlenmesiyle eğitim sırasında oluşabilecek güvenlik risklerini ve kazaları önemli ölçüde azaltılmıştır.

#### Projenin Sınırları:

- **Sensör Hassasiyeti:**  
MPU6050 sensörü, ani hareketler veya titreşimlerden etkilenebildiği için zaman zaman yanlış ölçüm yapabilmektedir. Bu durum, sensörün kalibrasyon ihtiyacını doğurmaktadır.
- **Ortam Şartları:**  
Hedef tespit sistemi, **ışık koşullarına** bağımlıdır. Yetersiz veya fazla ışıklı ortamlarda YOLOv8 modelinin performansı düşebilmektedir.
- **Veri İletimi Bağımlılığı:**  
ESP8266'nın Adafruit IO platformuna veri göndermesi Wi-Fi bağlantısına bağımlıdır. İnternet bağlantısının kesilmesi durumunda sistemin uzaktan veri iletimi devre dışı kalmaktadır.
- **Model Performansı:**  
YOLOv8 modeli, yalnızca eğitildiği veri seti üzerindeki hedefleri tanımlayabilir. Farklı hedef türleri eklenmek istenirse modelin yeniden eğitilmesi gerekmektedir.

### 5.3. Gelecekteki Çalışmalar İçin Öneriler

Geliştirilen sistemin iyileştirilmesi ve daha geniş kapsamlı hale getirilmesi için aşağıdaki öneriler sunulabilir:

1. **Sensör Performansının Artırılması:**
  - Daha hassas ve düşük gürültü seviyesine sahip IMU (Inertial Measurement Unit) sensörleri kullanılarak eğim algılama doğruluğu artırılabilir.
  - Sensör verilerinin filtrelenmesi için **Kalman Filtresi** gibi yöntemler uygulanarak yanlış ölçümlerin önüne geçilebilir.
2. **Hedef Tespit Sisteminin Geliştirilmesi:**
  - YOLOv8 modelinin **daha geniş veri setleri** üzerinde eğitilmesi sağlanarak farklı türdeki hedeflerin algılanması mümkün kılınabilir.
  - Düşük ışık koşullarında performansı artırmak için veri artırma (augmentation) tekniklerine gece-görüş simülasyonları eklenebilir.
3. **Kablosuz Bağlantı Alternatifleri:**
  - Veri iletimi için **LoRa** veya **Zigbee** gibi düşük enerji tüketen kablosuz protokoller kullanılabilir. Bu sayede uzak mesafelerden veri iletimi sağlanabilir.
4. **Mobil Uygulama Entegrasyonu:**
  - Geliştirilecek bir mobil uygulama ile Adafruit IO verilerine erişim sağlanabilir ve kullanıcıların performans analizini mobil cihazlar üzerinden takip etmesi mümkün hale getirilebilir.
5. **Donanım Geri Bildirimlerinin Geliştirilmesi:**
  - Hatalı eğim durumunda titreşim motoru (haptic feedback) kullanılarak kullanıcıya fiziksel geri bildirim sağlanabilir.
  - LED ve buzzer dışında küçük bir ekran modülü ile eğim açıları anlık olarak gösterilebilir.
6. **Gerçek Zamanlı Video Analizi:**
  - Bilgisayar kamerası yerine **dahili bir kamera modülü** (örneğin Raspberry Pi Camera) kullanılarak taşınabilir bir sistem oluşturulabilir.
7. **Enerji Verimliliği:**
  - Sistemin güç tüketimi optimize edilerek taşınabilirliği artırılabilir. Daha uzun süre çalışabilmesi için lityum pil destekli güç kaynakları eklenebilir.

## 6. KAYNAKLAR

- 1) Derin öğrenme nedir? Makine Öğrenimi ve Yapay Zeka ile Arasındaki Farklar Nelerdir? (<https://bulutistan.com/blog/derin-ogrenme-nedir/>)
- 2) Makine Öğrenimi ve Derin Öğrenme Arasındaki Farklar (<https://aws.amazon.com/tr/compare/the-difference-between-machine-learning-and-deep-learning/>)
- 3) What is Object Detection in Computer Vision? (<https://www.geeksforgeeks.org/what-is-object-detection-in-computer-vision/>)
- 4) Object Detection Techniques in Computer Vision (<https://medium.com/swlh/object-detection-techniques-in-computer-vision-7c169771fb15>)
- 5) YOLO: Algorithm for Object Detection Explained (<https://www.v7labs.com/blog/yolo-object-detection>)
- 6) Object Detection ve YOLO (<https://medium.com/academy-team/object-detection-ve-yolo-27e0b5ecff14>)
- 7) NodeMCU ESP8266 (<https://components101.com/development-boards/nodemcu-esp8266-pinout-features-and-datasheet>)
- 8) NodeMCU ESP8266 Detailed Review (<https://www.make-it.ca/nodemcu-details-specifications/>)
- 9) MPU6050 (<https://invensense.tdk.com/wp-content/uploads/2015/02/MPU-6000-Datasheet1.pdf>)
- 10) How to Train YOLOv8 Object Detection on a Custom Dataset (<https://colab.research.google.com/github/roboflow-ai/notebooks/blob/main/notebooks/train-yolov8-object-detection-on-custom-dataset.ipynb>)
- 11) What is MQTT? (<https://aws.amazon.com/what-is/mqtt/>)
- 12) MQTT Protocol Explained: Ultimate Guide for IoT Beginners (<https://www.emqx.com/en/blog/the-easiest-guide-to-getting-started-with-mqtt>)