**TEKNOFEST**

**HAVACILIK, UZAY VE TEKNOLOJİ FESTİVALİ**

**TARIMSAL İNSANSIZ KARA ARAÇLARI YARIŞMASI**

**KRİTİK TASARIM RAPORU**

**TAKIM ADI**

**Cyberova TİKA**

**PROJE ADI**

**Cyberoot**

**BAŞVURU ID**

**354483**

**İçindekiler**

[1. Takım Organizasyonu 4](#_Toc105565933)

[2. Ön Tasarım Raporu Değerlendirmesi 4](#_Toc105565934)

[2.1. Mekanik Değerlendirme 5](#_Toc105565935)

[2.2. Elektrik Değerlendirme 5](#_Toc105565936)

[2.3. Yazılım Değerlendirme 5](#_Toc105565937)

[2.4. Malzemeler 6](#_Toc105565938)

[3. Araç Özellikleri 7](#_Toc105565939)

[3.1. Mekanik Özellikler 7](#_Toc105565940)

[3.1.1. Gövde 7](#_Toc105565941)

[3.1.2. Tekerlek-Süspansiyon 8](#_Toc105565942)

[3.1.3. Tekerlek 8](#_Toc105565943)

[3.1.4. Püskürtme Sistemi 9](#_Toc105565944)

[3.2. Elektronik Özellikler 9](#_Toc105565945)

[3.3. Motor ve Motor Sürücüler 9](#_Toc105565946)

[3.3.1. Motor Sürücü Devresi 10](#_Toc105565947)

[3.4. Haberleşme 11](#_Toc105565948)

[3.5. Pil ve Güç 12](#_Toc105565949)

[3.5.1. Voltaj Regülatörü Devresi 12](#_Toc105565950)

[3.6. Yazılımsal Özellikler 13](#_Toc105565951)

[3.6.1. Haritalama 15](#_Toc105565952)

[3.6.2. Görüntü İşleme ile Yabani Ot Tespiti 15](#_Toc105565953)

[4. Sensörler 17](#_Toc105565954)

[5. Araç Kontrol Ünitesi 18](#_Toc105565955)

[6. Otonom Sürüş Algoritmaları 19](#_Toc105565956)

[6.1.1. Genel Algoritma 19](#_Toc105565957)

[6.2. Sıra Tespit ve Takip Algoritması 20](#_Toc105565958)

[6.3. Yabani Ot Tanıma ve Müdahale Algoritması 21](#_Toc105565959)

[6.4. Engel ve Çukur Tespiti Algoritması 22](#_Toc105565960)

[8. Özgün Bileşenler 23](#_Toc105565961)

[8.1. Mekanik 23](#_Toc105565962)

[8.1.1. Tekerlek Tasarımı 23](#_Toc105565963)

[8.1.2. Elektrik Kabineti 23](#_Toc105565964)

[8.1.3. Rocker Süspansiyon Sistemi 24](#_Toc105565965)

[8.1.4. Sigma Profil 24](#_Toc105565966)

[8.1.5. Püskürtme sistemi 24](#_Toc105565967)

[8.2. Yazılım 25](#_Toc105565968)

[8.3. Elektrik 25](#_Toc105565969)

[9. Güvenlik Önlemleri 25](#_Toc105565970)

[10.1. Mekanik Testler 26](#_Toc105565971)

[10.1.1. Jant Analizi 26](#_Toc105565972)

[10.1.1.1. Toplam Deformasyon Analizi 26](#_Toc105565973)

[10.1.1.2. Stres Analizi 26](#_Toc105565974)

[10.1.2. Motor Yatağı Analizi 26](#_Toc105565975)

[10.1.2.1. Toplam Deformasyon Analizi 26](#_Toc105565976)

[10.1.2.2. Stres Analizi 27](#_Toc105565977)

[10.1.3. Jant Kolu Analizi 27](#_Toc105565978)

[10.1.3.1. Toplam Deformasyon Analizi 27](#_Toc105565979)

[10.1.3.2. Stres Analizi 27](#_Toc105565980)

[10.1.4. Bacak Silindir Profil 28](#_Toc105565981)

[10.1.4.1. Toplam Deformasyon Analizi 28](#_Toc105565982)

[10.1.4.2. Stres Analizi 28](#_Toc105565983)

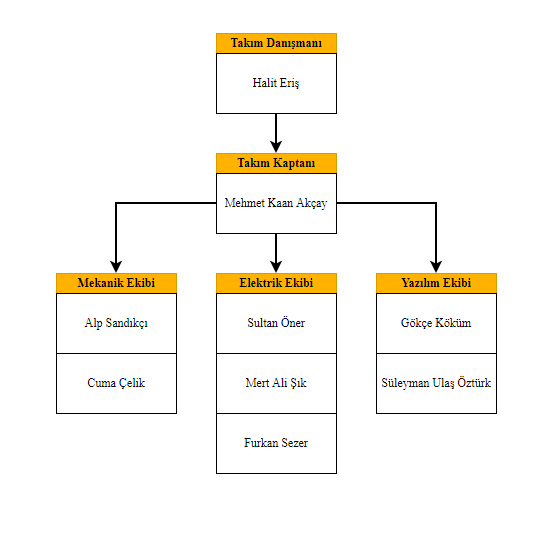
[10.2. Yazılım Testleri 28](#_Toc105565984)

[11. Referanslar (Kaynakça) 30](#_Toc105565985)

# Takım Organizasyonu

Cyberova TİKA Çukurova Üniversitesi’nin mühendislik fakültesinin çeşitli dallarında öğrenim görmekte olan 8 öğrenciden ve 1 Araştırma Görevlisinden oluşmaktadır.

Takım 5 ana gruba ayrılmaktadır, takım danışmanı takıma gerekli kaynakların sağlanması ve mentorluk yapmakla görevlidir. Takım kaptanı takım içindeki işleyişin planlanması ve takibi, takıma gerekli kaynakların sağlanması için gerekli mercilerle iletişime geçilmesinden sorumludur. Mekanik ekibi üyeleri aracın tasarımından, analizlerinin yapılması ve üretimden sorumludur. Elektrik ekibi üyeleri aracın güç dağıtımı, pil, motor seçimi, araç ile istasyon arasında haberleşmesi ve aracın güvenliğinden sorumludur. Yazılım ekibi haritalama ve otonom sürüş düğümleri, sensörlerden, görüntü işleme ve nesne tespitinden sorumludur.





İş Paketleri

# Ön Tasarım Raporu Değerlendirmesi

Aracımız üzerinde ön tasarım raporundan sonraki süreçte yaptığımız analiz ve testler sonucunda ulaştığımız verilere dayanarak bir takım iyileştirme ve geliştirme faaliyetlerinde bulunuldu. Bu faaliyetler sonucunda aracın maliyeti çok fazla değişim göstermese de mekaniksel parçaların mukavemetlerinde ve dayanımlarında büyük ölçüde artış gözlemlenmiştir.

## Mekanik Değerlendirme

İnsansız kara araçlarının tarım alanında kullanmak insana kıyasla günün her saati çalıştırılabilir olması, mahsullerin ve çiftlik hayvanlarının sağlığının izlenmesi, yabani otlarla mücadele edebilmesi gibi tarımda verimliliği artırma yönünde kullanılmaktadır. Ön tasarım raporumuzun tekrar üstünden geçip görevlerde sorun yaşayabileceğimiz ve güvenlik açısından bizi tehdit eden durumların değerlendirmesini yaptık. Bunun sonucunda bazı sistemler için özgün bir tasarımın yanında yeni parça seçimiyle kara aracımızın daha verimli olmasını sağladık. Bunlardan bir tanesi bacak silindir profilin çapını ve et kalınlığını hesaplanan ölçekte artırmak oldu. Diğer bir değişiklik ise bacak tutucu parçaların kalınlıklarının arttırılması ve değişen boru çapına uyarlanmasıdır. Araçta yaptığımız diğer büyük değişiklik ise ilaçlama sistemini daha kullanışlı ve verimli yapma amacıyla komple değiştirmek olmuştur. Son olarak da aracın sigma profil parçalarının gereksiz büyüklüğü ve ağırlığı nedeniyle daha küçük olanları ile değiştirilmesi olmuştur.

## Elektrik Değerlendirme

Ön tasarım raporumuzda motor sürücü olarak BTS7960B hazır kartını kullanmaya karar vermiştik. Fakat kritik tasarım raporumuzda motor sürücümüz konusunda yerli, milli, özgün ve daha verimli olması için kendi kart tasarımımızı yapmaya karar verdik. Bu kart tasarımını yaparken yüksek akım çeken motorlarımızı besleyebilecek, bizi kablo karmaşasından kurtaracak ve kullanım açısından basit bir yazılım ile kontrol edilebilecek bir kart tasarladık. Ayrıca verimlilik adına motorda dönme etkisinden oluşacak kapasitif etkiyi daha hızlı deşarj etmesi için base bacaklarında diyot kullandık. Kart seçimizdeki ana değişikler bunlar olup bizim bu kararı almamızı sağlamıştır. Kartımızda kullanılan ana prensip H-bridge mantığı olup gerekli detay raporumuzda motor sürücü kısmında belirtilmiştir. Ayrıca regülatör kartımız için hazır kart kullanma fikrimizi motor sürücüde olduğu gibi yerli, milli ve özgün olması adına kendi tasarımımız ile değiştirdik. Ana hedefimiz her yere uyum sağlayacak bir regülatör kartı olmasıdır. Bunu sağlamak adına çok türlü trimpot kullanmış olup verimliliğini ise Buck converter sistemi ile arttırmış bulunmaktayız.

Aracın manuel sürüşünde kullanılan, RF alıcı verici sistemi olaran Taranis X9D Plus With X8R Receiver kullanmaktan vazgeçip aracımızın manuel sürüş esnasında sensörlerden gelen veri bilgilerinin görülebilmesi için kontrol arayüzü oluşturuldu.

## Yazılım Değerlendirme

Ön tasarım raporunda bahsedilen derinlik algısı oluşturarak mesafe ölçümü için ZED stereo kameranın kullanımı iptal edilerek bunun yerine lidar kullanılmasına karar verilmiştir. Yabani otların tespit edilebilmesi için ayrı bir kamera kullanılmıştır. Bu durumun sebebi belirlenen stereo kameranın yüksek maliyetli oluşu ve ROS entegrasyonunun karmaşık olmasındandır. ROS uyumlu modellerde ise derinlik algısının oluşturulabilmesi için komplike matematik işlemleri içeren algoritmalar gerekmesi ve elde edilen verinin diğer sensörlere iletilmesi kısmında uyumsuzluklar yaşanabildiği için tercih edilmemiştir. Bunların yerine hem gelen verilerin diğer görevlerde de kullanılabilmesi hem de gelen veri ile direkt olarak sınır belirleme ve haritalama yapılabilmesi sebebiyle lidar tercih edilmiştir.

İlk planlama aşamasında araçtan gelen sensör verilerinin ve kamera görüntüsünün kontrol masasına Wi-Fi üzerinden iletilmesi planlanmıştı. Kritik tasarımda bu kısımda değişikliğe gidilerek aracın kamera görüntüsünün direkt olarak kontrol masasına aktarımı askıya alınarak Wi-Fi protokolü kullanılması yerine XBee üzerinden yalnızca telemetri verilerinin aktarılmasına karar verilmiştir. Geliştirme ve iyileştirmeler hakkındaki detaylar raporun ilgili kısımlarında açıklanmıştır.

## Malzemeler

tablo içeren bir resim

Açıklama otomatik olarak oluşturuldu

Değişen Malzemeler

tablo içeren bir resim

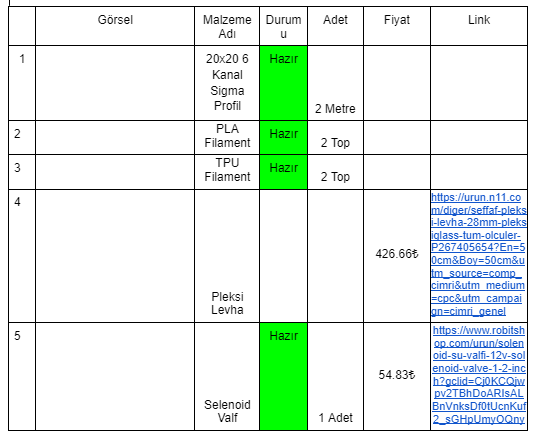
Açıklama otomatik olarak oluşturuldu

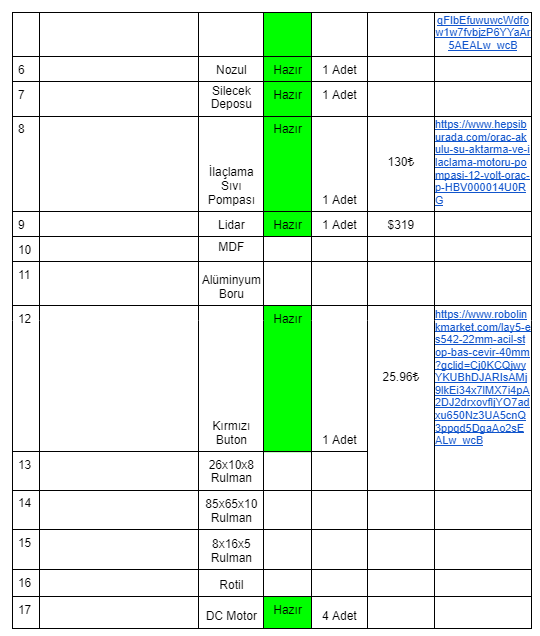
Malzeme Listesi

# Araç Özellikleri

## Mekanik Özellikler

Aracımız 985 mm uzunluğa, 694 mm genişliğe, 617 mm yüksekliğe 35-40 kg arası bir ağırlığa sahiptir ve mekanik sistemler olarak 3 ana başlık altında incelenebilir. Bunlar; gövde, tekerlek-süspansiyon ve püskürtme sistemidir.





### Gövde

Aracın gövdesi modifiye etmemizde ve tasarımda kritik değişiklere gitmemizde bize yeterli toleransı sağlaması açısından, ayrıca malzeme temini ve bağlantı parçalarının çeşitliliğinden dolayı 20x20 6 Kanal Sigma Profil kullanılmıştır. Kullanılacak olan diğer parçaların (elektronik kabini, sigma, bağlantı aparatları vs.) montajlanması için kendi tasarımımız olan parçaların 3D yazıcı ile basılması planlanmıştır.

el arabası içeren bir resim

Açıklama otomatik olarak oluşturuldu

### Tekerlek-Süspansiyon

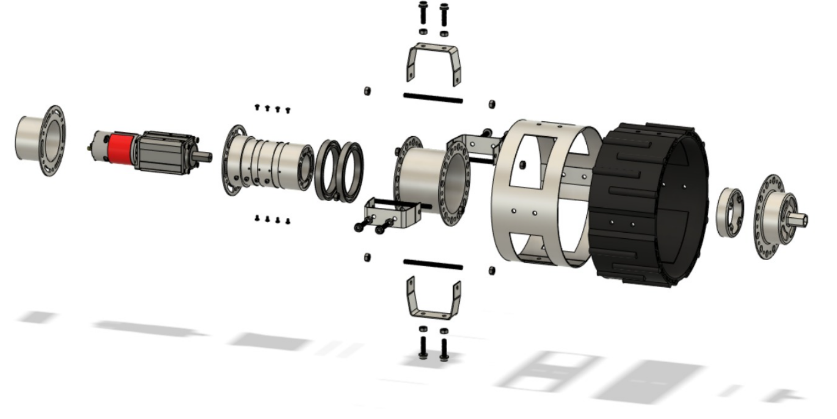
Tekerlekler araca diferansiyel bağlantısı kullanılarak bağlanmıştır. Süspansiyon sistemi ise kullanılan diferansiyel bağlantısının merkezine dik olan aksis etrafında serbest dönüşü ile Rocker-Suspension sistemi örnek alınarak tasarlanmıştır.

duvar içeren bir resim

Açıklama otomatik olarak oluşturuldu

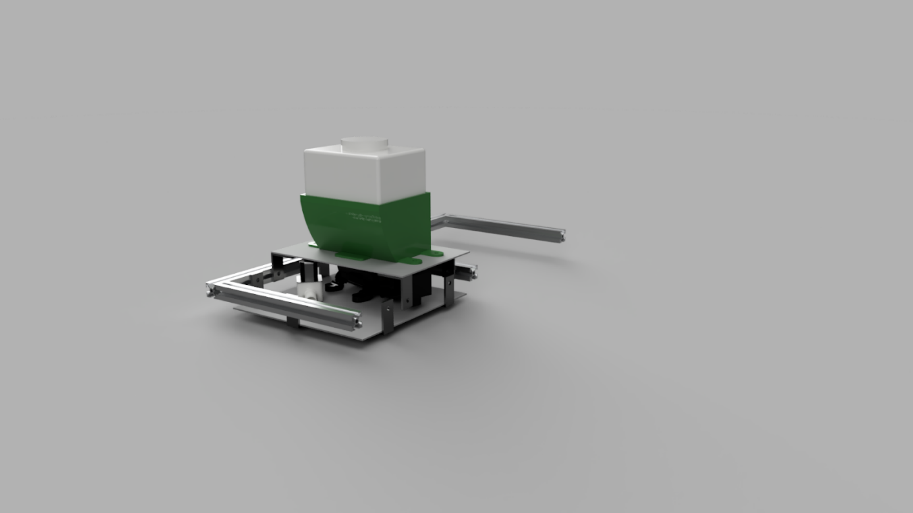
### Tekerlek

Kullandığımız tekerlekler ise motoru jantın içinde olan tamamıyla bize ait bir tasarıma sahiptir. Tekerlek çapı 200 mm dir. Tekerlek jantı ve taşıyıcı parçalarının 6063 Alüminyum sac kullanılarak yapılmıştır. Bu tasarım sayesinde aracın motoru dış etkenlerden korunur. Ayrıca tekerlek taşıyıcı parça üzerinde bulunan vida delikleri sayesinde motorların içeride fazla ısınmasına önlem olarak fan takılabilme opsiyonu bulunmaktadır. Amaca göre bu kısımlar su ve toz gibi etkenlere karşı sızdırmazlık için de kullanılabilmektedir.

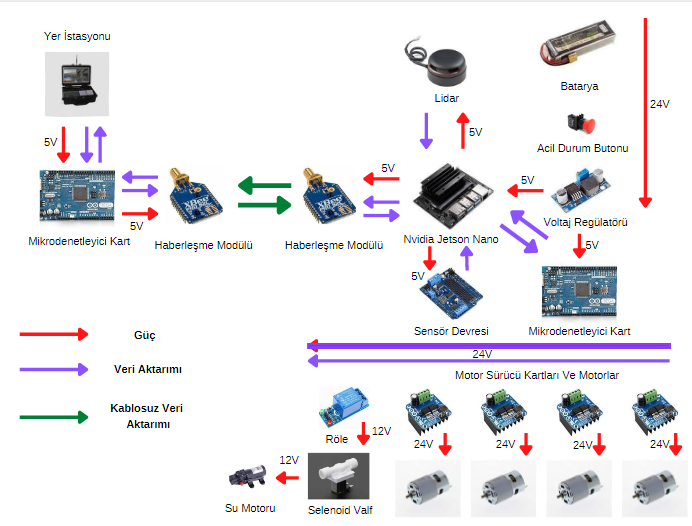


### Püskürtme Sistemi

Püskürtme sisteminde kullanılacak olan zirai ilaçlama sıvısının (herbisit) ilaçlanacak olan bitkinin yüzeyine olabildiğince temas etmesi için piyasadan kolayca temin edebileceğimiz, güvenilir ve uygun fiyatlı ürünleri tercih ettik. Bunlar 2 litre araç silecek suyu deposu, 12V-5 bar sensörlü sıvı pompası, 12V selenoid valf ve nozuldan oluşmaktadır.



## Elektronik Özellikler



## Motor ve Motor Sürücüler

Motor olarak “24V 1000 Rpm Magnum 775 serisi Forceup Planet Redüktörlü Yüksek Torklu DC Motor” ürününde karar kıldık. Çünkü güç tüketimi ve sağladığı tork miktarı göz önüne alınarak parkur alanında bize gerekli koşulları sağlayacağını hesapladık. Nominal torkumuz 23 Kg/cm’dir. Motorumuzun durdurma ise torku 46 Kg/cm’dir. Araç boştayken 24V 1000 Rpm değerleri ile yükün stabilizasyonunu hedef alarak yaptığımız hesaplamalar sonucunda hızımızı 1Km/h olarak ayarladık. Ayrıca gerekli olan tork ve kaldırılacak yük göz önüne alınarak 4 adet motor kullandık.

|  |  |
| --- | --- |
| Voltaj Değeri | 24V |
| Güç | 240W |
| Yüksüz Amper | 1.5A |
| Durdurma Amperi | 20A |
| RPM Değeri | 1000 |
| Motor Uzunluğu | 14cm |
| Motor Çapı | 42mm |

V= (Rpm)\*2\*pi\*R

M= m\*g

Mgx= M\*sin(θ) Mgy= M\*cos(θ)

T= f \* R

m\*a= f - Mgx

m=50kg, V=1 m/s, a=0.2m/s², R=0.02m, θ=12, g=9,81 m/s²

Rpm= (6000)/ 2\*pi\*10 → Rpm= 95.541

M= m\*g= 50x9,81 =490.5 N

Mgx= M\*sin(θ)= 490.5\*0,208=102.024 N Mgy= M\*cos(θ)= 490.5\*0,978=479.709N

T= f \* R

m\*a= f – Mgx m\*a=( T/R) – Mgx 50\*0,2= (T/0,1) – 102.24 N

T=35.224 N.m değerinde bulunur.

### Motor Sürücü Devresi

ÖTR raporumuzda belirttiğimiz motor sürücümüz olan BTS kartını kendi tasarımımız ile değiştirmeye karar verdik. Bunun sebepleri yarışmada önemli olan yerlilik, özgünlük ve verimlilik gibi alanlarda avantaj sağlamaktı. Kartımızı tasarlarken amacımız MOSFET’lerimizi H-bridge mantığını kullanarak sürmektir. Bu işlemde mosfet sürücü olarak IR2104 MOSFET sürücüsünü kullandık. IR2104 seçmemizin sağlayan etkenler çok karmaşık olmayan kullanım mantığı, PWM kontrolü sağlaması ve SD (shut down) pinine bağlayabileceğimiz akım sensörü ile güvenlik açısından gerekli ve önemli avantajlar sağlamasıdır.

metin, iç mekan, siyah, döşenmiş içeren bir resim

Açıklama otomatik olarak oluşturuldu

Şekil 1.1 Motor sürücü şematik gösterimi

Şekil 1.1’ de gözüktüğü gibi H-bridge oluşturulması, kullanılan devre elemanları ve devre elemanlarının değerleri gösterilmiştir.

metin, elektronik eşyalar, devre içeren bir resim

Açıklama otomatik olarak oluşturuldu

Şekil 1.2 Motor sürücü kartı ve parametreleri

Şekil 1.2’de kart tasarımımız verilmiştir. Motorlarımız 24V ve zorlanmaları durumunda yaklaşık 10A-20A akım çekebilmektedirler. Bu yüzden kartımızı tasarlarken bileşenlerin üzerinden geçecek akıma göre kablo kalınlıkları değiştirilmiştir. Kullanılan H-bridge mantığı ise basit bir şekilde anlatmak gerekirse anahtarlama mantığı ile anlatılabilir. Motorumuzu döndürmek istediğimiz yere göre MOSFET’lere high ve low değerlerini veriyoruz. Aynı yöndeki MOSFET’ler için biri high alıyorsa diğeri low almak zorunda oluyor aksi takdirde akımımız yükten yani motordan geçmeyip kısa devre olarak direkt toprağa bağlanıyor. Bu yüzden çapraz olacak şekilde high ve low veriyoruz. Ayrıca verimliliği artırmak için MOSFET’lerimizin base bacaklarına diyot yerleştirdik. Bu diyotun görevi motorda dönme etkisinden dolayı oluşacak kapasitif etkiyi daha hızlı bir şekilde deşarj etmektir.

### Haberleşme

Aracın sistem haberleşmesinde NVIDIA JETSON NANO (Ana bilgisayar) kullanıldı. Sensör devresi sensörlerin tek bir karta bağlı olduğu devredir, bu sayede tüm sensörlerden gelen veriler tek bir kart üzerinden Jetson’a gelecek. Mikrodenetleyici (Yardımcı kart) kart, sürücü kartlarına bağlıdır, Jetson’dan gelen komutlarla motor sürücü kartları motorları sürecektir. Yer istasyonuyla ana bilgisayar arasındaki haberleşmeyi XBee haberleşme modülü sağlayacaktır. Ayrıca XBee haberleşme modülü aracın manuel sürüş sırasında sensörlerden gelen verileri kontrol istasyonuna ulaştıracaktır.

### Pil ve Güç

Cyberova TİKA’nın güç kısmında 22.2V 6S 22000mAh 25C LiPo batarya kullanılması planlanmıştır. Kullanılması planlanan pil “kapasite” x “c değeri” formülünden yaklaşık olarak 550 Amper maksimum akım verebilmektedir. Batarya konusunda Li-Ion, LiFePo4, LTO ve LiPo bataryalar arasında kalınmış ancak Li-Ion bataryaların LiPo’ya denk hallerinin ağırlık ve hacim olarak LiPo’dan yüksek olması ile devreye verebilecekleri maksimum akım değerinin de düşük olmasından dolayı Li-Ion seçeneklerimiz arasından çıkarılmıştır. LiPo ve Li-Ion bataryaların dışında LiFePo4 piller de göz önünde bulundurulmuş ancak sağlayacağı faydaların yanında çok büyük dezavantajları vardı; düşük nominal voltaj, ağırlık, daha hassas olmaları ve pahalı olmaları bu dezavantajlarından bazılarıdır. LTO piller ise yüksek kapasitelerine nazaran düşük nominal voltaj (2.3V) ve boyut, ağırlık, fiyat konularındaki dezavantajlarından dolayı kullanılmamıştır. Tüm bu pil çeşitlerini karşılaştırdığımızda en mantıklı seçim olarak LiPo batarya kullanılmasına karar verilmiştir. Kullanacağımız LiPo pilin ağırlığı 2588 gram olup kapladığı alan ise 206mmx92mmx67mm’dir. LiPo pil yapısı gereği çok hassas bir pildir. Dış muhafazası keskin bir cisimle temas ederse patlama oluşup çevreye ve doğaya zarar vereceğinden aracın gerekli güvenlik önlemlerinin alındığı güvenli bir yere konumlandırılacaktır.

Güç dağıtım konusunda ise pilimiz nominal 22.2V-25.2V arasında çalışmaktadır. Kullandığımız motorlar 24V olsa da her bileşen farklı bir voltaj aralığında çalıştığı için 5V-9V-12V değerlerine ihtiyacımız oluyor. Burada sabit voltaj ve sabit akım kullanan ve voltaj farkının fazla olmadığı durumlarda doğrusal regülatör kullanılmıştır çünkü lineer regülasyonda gelen voltaj ile çıkan voltaj farkının arttığı durumlarda regülatör artan enerjiyi direnç gibi davranıp ısı yoluyla harcadığı için hem sistem gereksiz şekilde ısınmakta ve enerjiyi boşa harcamaktadır. Yüksek ve değişken akım isteyen bileşenlerde ise anahtarlamalı regülatörler kullanılmıştır çünkü anahtarlamalı regülatörler direnç gibi enerjiyi ısı yoluyla kaybetmemize sebep olan devre elemanları bulundurmadığı ve minimum düzeyde kayıp maksimum düzeyde verimlilik sağladığı, bunlardan dolayı çevreye ısı yayıp devreyi ısıtmadığı ve enerji kaybına sebep olmadığı için lineer regülatör kullanmadığımız yerlerde tercih edilmiştir.tablo içeren bir resim

Açıklama otomatik olarak oluşturuldu

Aracımızda kullanılan ve güç tüketiminde en büyük paya sahip olan bileşenler ve akım voltaj değerleri yukarıdaki tabloda verilmiştir. Aracımızda bulunan selenoid valf ve su motoru saatlik güç tüketimleri yüksek olsa bile kullanılacağı süre birkaç saniye olacağı için düşük güç tüketmektedirler. Bundan dolayı tabloya eklenmemiştir.

### Voltaj Regülatörü Devresi

Tarım aracımız için hazır regülatör kartı kullanmak yerine kendi tasarımımız olan kartımızı kullanmaya karar verdik. Bu kararı vermemize neden olan şey yarışmada özgünlük, yerlilik ve verimlilik gibi kavramların önemli olduğunu düşünmemizdir.

metin, skorbord içeren bir resim

Açıklama otomatik olarak oluşturuldu

Şekil 1.1 Regülatör devresi şematik gösterimi

metin, devre, elektronik eşyalar içeren bir resim

Açıklama otomatik olarak oluşturuldu  
Şekil 1.2 Devre kartı tasarımı ve parametreleri

Regülatör kartımızın ana mantığı Buck Converter’ a dayanmaktadır. LM2576 kullanarak oluşturduğumuz regülatör kartımızda en önemli etken girişe uyguladığımız gerilim çıkıştan büyük olmak zorundadır. Sadece direnç kullanarak yapabileceğimiz bu devreyi Buck Converter şekilde kullanmamızın amacı ise kayıpları azaltmaktır. LM2576’ yı kullanmamızdaki sebep ise çekilen akım arttığında oluşacak olan voltaj düşümüne karşı kendi içinde feedback içermesidir. Bu sayede oluşacak kayıplardan ve devremizin ihtiyaç duyacağı voltajı karşılayamama gibi durumlardan kaçınmış oluyoruz. Ayrıca çizdiğimiz devrede çok turlu trimpot kullanarak regülatör devremizi aracımızın herhangi bir yerinde kullanma esnekliğini sağlamış oluyoruz.

## Yazılımsal Özellikler

Aracın kontrolü için ana bilgisayar görevindeki Jetson Nano’ya açık kaynaklı yazılım olan ROS *(Robot Operating System)* yüklenecektir. ROS sürümü olarak Noetic, ROS’un birlikte çalıştırılacağı Linux dağıtımı olarak Ubuntu 20.04 kullanılmıştır. ROS eş zamanlı olarak birden fazla programlama dilini çalıştırabilmesi sebebiyle farklı sistem yazılımlarını bir arada çalıştırabilmektedir. Bu özelliği sayesinde, sensörleri yöneten bir düğümü C++ ile yazarken kameradan gelen görüntüyü Python kullanarak ayrı bir düğümde çalıştırabilmektedir. Ayrıca birden fazla düğümü aynı anda çalıştırma imkanı sunarak karmaşık yazılımları küçük parçalar halinde farklı programlama dilleri ile yönetebilmemizi sağlamaktadır. ROS’un çalışma mantığında farklı konularda *(topic)* oluşturulan mesajlar *(messages)* üzerinden veri iletimi yapılır. Yayıncı-abone *(publisher-subscriber)* mantığında çalışan düğümler *(nodes)* her sensör için ayrı ayrı oluşturulur. Yazılımın ana bir düğüm dışında parça parça düğümlere bölünmesi geliştirilebilirlik açısından kolaylık sağlamaktadır. Düğümlerin birbiriyle haberleşmesini sağlayan temel uygulama *ROS-Master* adını alır. Yalnızca bir istek olduğunda yanıt veren talep-cevap iletişim modelini kullanan yapılar ise servisler olarak adlandırılır. İsteğe yanıt verildikten sonra düğümler arasındaki bağlantı kesilir, bu sayede ağın yükü azalmış olur.

ROS’un ayrı bir avantajı olarak ROS’un kendi simülasyon ortamı olan Gazebo ile henüz üretim aşamasına geçilmemiş olan projelerde herhangi bir maliyet gerekmeksizin tasarlanan aracın hareketleri ve amacına uygun çalışıp çalışmadığı 3D olarak test edilebilmektedir. Haritalama işlemi sonrası çıkarılan sınır alanları gibi araçtan paketler halinde gelen veriler RViz görselleştiricileri sayesinde simülasyon ortamında gözlemlenebilmektedir. Bunlara ek olarak ROS’a uyumlu olarak oluşturulan birçok araç sayesinde gelen anlık verilerin zamanla değişimi eğim grafikleri üzerinden okunabilmektedir.

Projeye başlarken dosya dizininde ilk olarak ROS çalışma ortamımız olan *catkin\_ws* oluşturuldu. Bunun alt dizininde yer alan *CMakeLists.txt* metin editöründe eklenen her bir mesaj, servis, launch vb. dosyaları için düzenlemeye gidilmiştir. Yine aynı klasör içerisinde default olarak var olan *package.xml* dosyasında düzenleme yapılarak paketin içerisinde yer alan bilgiler özelleştirildi. Oluşturulan her bir düğümün kodları farklı dosya uzantısı ile scripts klasörü içerisinde yer almaktadır. Diğer bütün paket içerikleri servis, launch, mesaj vb. dosyaları *scripts* klasörü ile eş dizin içerisinde yer almaktadır. Terminale *roscore* komutunun girilmesi ile ROS çekirdeği çalıştırılarak sistem aktive edilmektedir. Bu adımdan sonra çalıştırılması gereken bütün düğümler sırayla ayrı bir terminal üzerinden *rosrun* komutu kullanılarak çalıştırılmaktadır. Bu şekilde *ROS-Master* bütün düğümler arasında iletişimi sağlayarak aracın otonom olarak ilerlemesi sağlanmaktadır.

Jetson Nano ve Arduino Mega arasında kablolu bağlantı ile sürekli iletişim sağlanacaktır. İkisi arasındaki veri iletimi UART protokolü kullanılarak yapılacaktır. Arduino ile Jetson görev dağılımları gelen verilerin işlenmesi konusunda ayrılmaktadır. Arduino’nun görev şeması motorların kontrol edilmesi, sensör verilerinin okunmasından; Jetson’ın görev şeması kamera görüntülerinin işlenmesi, nesne ve bitki tespitinin yapılması, haritalama işleminin gerçekleştirilmesinden oluşmaktadır. Arduino Mega’ya bağlı olan sensörlerden bir kısmında I2C protokolleri ile veri iletimi yapılmaktadır. Ayrıca motorların çalışması için gerekli olan komutlar da PWM sinyalleri ile iletilmektedir. Genel olarak açıklandığında sensörler, ek donanımlar ve ana bilgisayar arasında UART, I2C protokolleri ve PWM özellikleri kullanılarak veri akışı sağlanmaktadır. Kameradan ve diğer sensörlerden gelen veriler Jetson anabilgisayarına gönderilip ana işlemci üzerinden işlenmektedir.

Aracın hareketi sırasında hız kontrolünün sağlanabilmesi için IMU içerisinde yer alan hız ve ivme sensörleri kullanılacaktır. IMU doğrudan Arduino Mega ile bağlantılıdır, gelen veriler direkt olarak burada kullanılacaktır.

Algoritmaların geliştirilme aşamasında farklı donanımlar için birden fazla yazılım dili kullanılmıştır. Bunları kullanım amaçlarına göre sıralamak gerekirse haritalama ve görüntü işleme kısmında Python, motor kodlarında Arduino’nun özelleşmiş C++ temelli kendi dili kullanılmış olup Jetson ile Arduino arasında seri haberleşme sağlanması için özelleşmiş bir kütüphane olan *pySerial* kullanılmıştır.

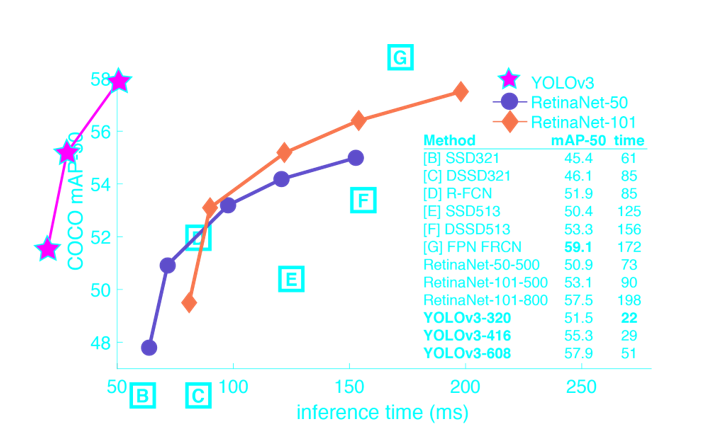
Araç dışarıdan bir müdahale olmadığı durumda kendisinden beklenilen görevleri tam otonom olarak gerçekleştirecektir. Eğer kullanıcı aracın kontrolünü almak isterse kullanıcı arayüzünde bulunan manuel moda geçiş butonu ile kontrolü kendisi sağlayabilmektedir. Aracı yönetmek için arayüzde yer alan ileri, geri vb. butonlar kullanılarak istenilen komutlar araca iletilir. Ayrıca olası acil durumlara karşı bir tanesi aracın üzerinde bir diğeri kullanıcı arayüzünde olan iki farklı acil durum butonu ile aracın hareketi anında sonlandırılarak araç bulunduğu yerde beklemeye alınabilmektedir.

### Haritalama

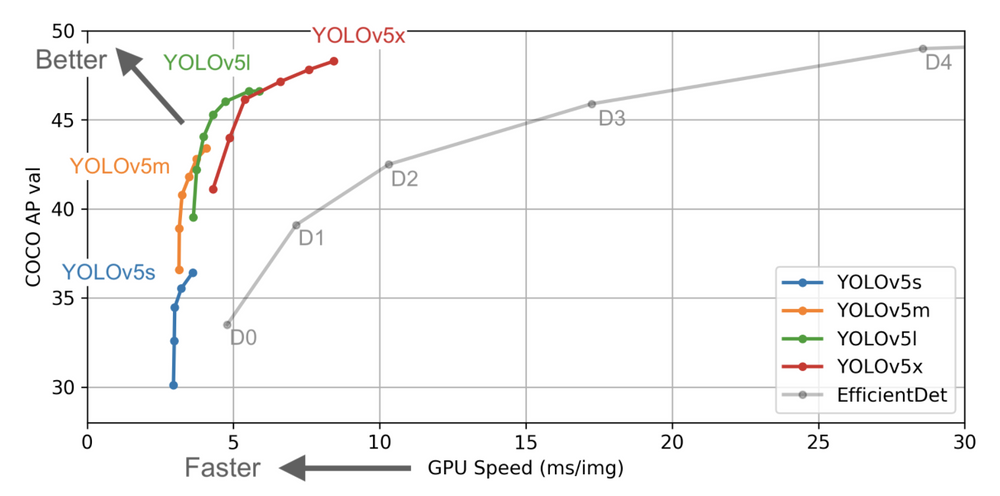
Aracın otonom bir şekilde ilerleyebilmesi için haritalama önemli bir yere sahiptir. Robotik alanında kullanılan haritalama iki önemli veriye göre şekillenir. Bunlar aracın pozisyonu gibi iç bilgileri destekleyen odometri verisi ve çevreye dayalı bilgiler için toplanan algı ve sensör verileridir. Projemizde ortam haritasının çıkarılması için daha etkili olduğunu düşündüğümüz lazer ile mesafe ölçümü yapan lidar sensörü kullanılmayı tercih ettik. Ek olarak sistem tasarımı kameradan toplanan veriler ile görüntü işleme tekniği kullanılarak arazi tanımlamasının iyileştirilmesi planlanmıştır. Aracın bulunduğu alanda lidar ile *LazerScan* taraması yapılarak parkur sınırları, parkur alanında yer alan nesnelerin araca göre konumları belirlenerek haritalama işlemi gerçekleştirilir.

### Görüntü İşleme ile Yabani Ot Tespiti

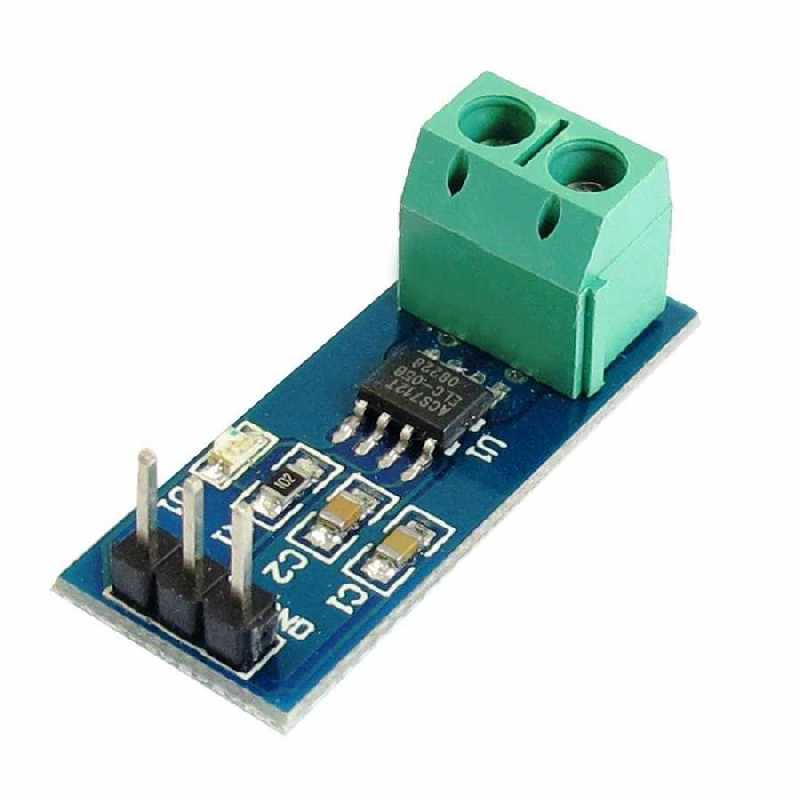
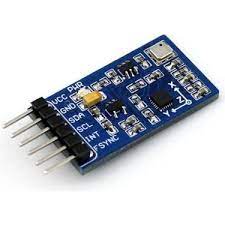
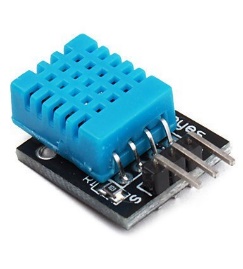
Yarışma şartnamesi gereğince verilen zararlı bitkiyi tanıyabilmek için bilgisayarlı görü çözümlerine ihtiyaç duymaktadır. Bu bağlamda bir çok derin öğrenme algoritmaları bulunmasına karşın bitkiyi en verimli şekilde tespit etmek için *CNN* *(Convolutional Neural Networks)* ağı tasarlanmıştır. İlk olarak R-CNN ve Fast R-CNN gibi çözümler denendiğinde anlık yayınlarda çok ciddi boyutlarda FPS sorunları yaşanmasıyla beraber bir başka ve yeni bir CNN algoritması olan *YOLO (you only look once)* algoritması kullanılmasına karar verilmiştir.



YOLO algoritması konvolüsyonel sinir ağlarını kullanarak verilen görüntüden nesne tespiti yapabilen açık kaynak kodlu bir algoritmadır. YOLO algoritmasının R-CNN ve Fast CNN algoritmalarından daha hızlı çalışmasının sebebi ise resmin tamamını tek seferde nöral sinir ağlarından geçiriyor olmasıdır.Öncelikle YOLO tespit ettiği nesnelerin etrafını bounding box adı verilen karesel şekiller içerisine sınırlar. Ve sonrasında elde ettiği görüntüyü 9x9, 17x17’lik ızgaralara böler. Ardından her ızgaranın içinde aranan nesne var mı diye düşünür ve bir kontrol yapar. Algoritmanın birden fazla ızgara içerisinde aynı görüntü olmasını düşünme ihtimaline karşın her *bounding box’*ın kendine özel güven skoru vardır. Algoritma bu bounding box’ların güven skorlarını kontrol eder ve en yüksek güven skoruna sahip bounding box’ı output olarak ekrana verir. YOLO algoritmasının YOLOv3, YOLOv4, YOLOv5 varyasyonlarından çözünürlüğü ve FPS değeri en yüksek olan algoritma YOLOv5 olduğu için bu algoritmanın kullanılmasına karar verilmiştir. YOLOv5 modelinin içinde YOLOv5s, YOLOv5m, YOLOv5l ve YOLOv5x varyasyonları bulunmaktadır. YOLOv5s varyasyonu az gecikme değeri (2.2ms) verir fakat buna karşın doğruluk oranı diğerlerine göre daha düşüktür. Bunun tam tersi de YOLOv5x için geçerlidir (6 ms). Aracımızda kullanacağımız gecikme değeri ile doğruluk oranı açısından en verimli olan YOLOv5l veya YOLOv5m modülleri üzerindeki testlerimiz halen devam etmekte olup aralarından en verimli olanını kullanmak yönünde kararımızı vermekteyiz.



# Sensörler



**Lidar sensörü:** Projede kullanılan lidar bir nesnenin veya bir yüzeyin uzaklığını algılamaya yarayan bir yöntemdir. Radardan farklı olarak ışık yani lazer darbelerini kullanır, radyo dalgaları yerine lazer ışınları tek bir çizgi üzerinde yayıldığı için sabit değil, sürekli hareket halinde çalışmaya ihtiyaç duyar. Uzaklığı ölçülecek nesne ya da yüzeye gönderilen lazer darbesinin gönderiliş zamanı ile nesneye çarpıp gelen yansımanın tekrar kaynağa ulaşma vakti arasındaki fark sayesinde uzaklık ölçülür. Işık hızında çalışması alanı çok hızlı bir şekilde, yüksek doğrulukla haritalandırması avantajı sunar. Lidar sensör 12 metre çapıyla ve 360 derece lazer tarama yaparak iki boyutlu haritalandırma yapabilecektir. Ayrıca bu ürünün programı ile yapılan SLAM işlemleri göstermektedir. Bağlantı olarak ana bilgisayar olan Jetson Nano’ya USB aracılığıyla bağlanmalıdır.

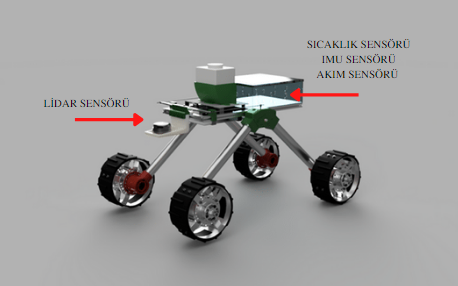
**Sıcaklık Sensörü:** Aracın genel sıcaklık durumunu ölçmek için SHT30 nem ve sıcaklık sensörü kullandık. SHT30 Sensör Modülü I2C haberleşme protokolü ile bağlanır ve yüzde 3 nem hassasiyetine,2.4V ve 5.5V geniş çalışma voltajı aralığına sahiptir.

**Inertial Measurement Unit (İnersiyal Ölçme Ünitesi):**

Ana işlemciye gönderilen açısal hız ve doğrusal ivme verisini tek bir modülde toplayan elektronik birime IMU denir. IMU sayesinde cihazın veya aracın hızını, dönme açısını ve dönüklük miktarlarını öğrenebiliyoruz. Sensörün çalışma gerilimi: 3.0 V – 5.0 V, haberleştiği protokol ise I2C’dir.

**Akım sensörü:**

ACS714 akım sensörünü kullanmayı tercih ettik. Bu sayede motorlardan gelen akımı gözlemleyebilmiş olduk. Bir iletken üzerinden geçen akımı ölçerek, amaca uygun bir sinyal oluşturan sensördür. Devrede korunum ve denetim için kullandık. 3V – 5.5 V voltaj aralığına sahip. IC sensör, aşırı akım durumuna beş sefere kadar dayanabilir.



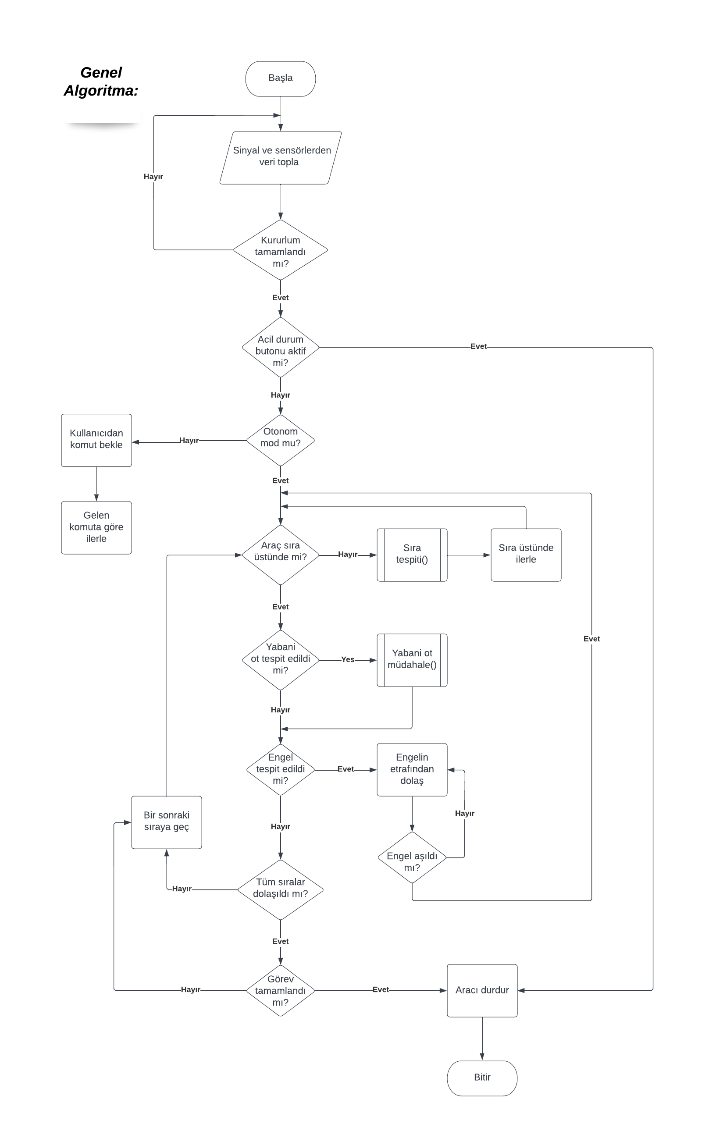
# Araç Kontrol Ünitesi

İnsansız kara aracımız otonom hareket edecek şekilde tasarlanmıştır. Aracın yardımcı bilgisayarı olarak Jetson Nano kullanılmıştır, yer istasyonuyla XBEE haberleşme modülü sayesinde haberleşecektir. Sensörlerden ve kameradan gelen veriler direkt olarak Jetson Nano’ya (yardımcı bilgisayar) gidecektir, ayrıca gerekli olan gücü de Jetson Nano’dan alacaktır. Jetson Nano’ya bağlı olan Atmega motorların harekete geçmesini sağlayacaktır ayrıca aracın uzaktan da kontrol edilebilir olması gerekmektedir. Kullanıcının araçtan gelen sensör verilerini takip edebilmesi ve gerektiğinde manuel bir şekilde kullanılabilmesi için gerekli olan kontrol arayüzünün *PyQt5* kullanılarak tasarlanmasına karar verildi. Temelde aracın haberleşme sistemi çift taraflı bir haberleşme ile çalışmakta olup arayüz sistemi ve araç üzerindeki *MasterNode* arasında çalışmaktadır. Arayüzde aracın kontrolü için otonom ve manuel olmak üzere iki farklı sürüş modu bulunmaktadır. Arayüzdeki her iki mod için aracın batarya seviyesi, ilaç sıvı seviyesi, alınan yol, konum, hız bilgisi, imha edilen yabani ot sayısı, araç içi sıcaklık derecesi gibi sensör verilerinin bilgisi aktarılacaktır. Acil bir durum olması halinde kullanıcıya hem mesaj hem de ikaz şeklinde uyarı verilecektir. Ayrıca tasarımda aracın anlık yaptığı bitki tespiti, ilaçlama, engelden kaçış gibi temel faaliyetlerine de yer verilecektir. Örneğin; engelle karşılaşıldığı senaryoda ‘Engel tespit edildi!’ ya da yabani ota müdahale sırasında ‘İlaç püskürtüldü.’ gibi kısa komutlarla durum açıklaması yapılacaktır. Arayüzün sağ üst köşesine yerleştirilen acil durum butonu sayesinde kullanıcı beklenmedik durumlarda motorlara giden gücü keserek aracı durdurabilecektir. İlave olarak kullanıcının manuel mod ile çalışmak istemesi halinde aracın ilerlemesi ve ilaç püskürtülmesi için ek butonlara yer verilmiştir. Ek olarak aracın dolaşımı sırasında karşılaştığı engellerin konumu, imha edilen bitki konumu, ilerlenen sıra düzlemi GPS verileri kaydedilerek arayüz uygulamasında bunların nokta gösterimi yapılacaktır. Tasarımın gerçekleştirilmesi için ilk olarak, ROS ile oluşturulan çalışma alanında bir CMakeList.txt dosyası oluşturulup dosya için gerekli tüm düğümler ve ROS ve *QtDesigner* içerisinde kullanılan kütüphaneler eklenmiştir. Bu sayede ROS tarafından algılanan pakete ikinci olarak gerekli arayüz kodları eklenmiştir. Daha sonra ise, arayüzün iletişimini sağlayacak düğümler oluşturulmuştur. Bu düğümler sayesinde, araçtan yayınlanan konulara abone olarak araç ve kontrol ünitesi arasında çift taraflı iletişim sağlanmıştır.

# Otonom Sürüş Algoritmaları

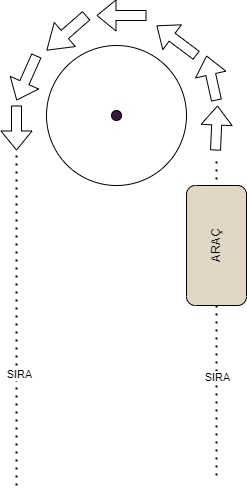
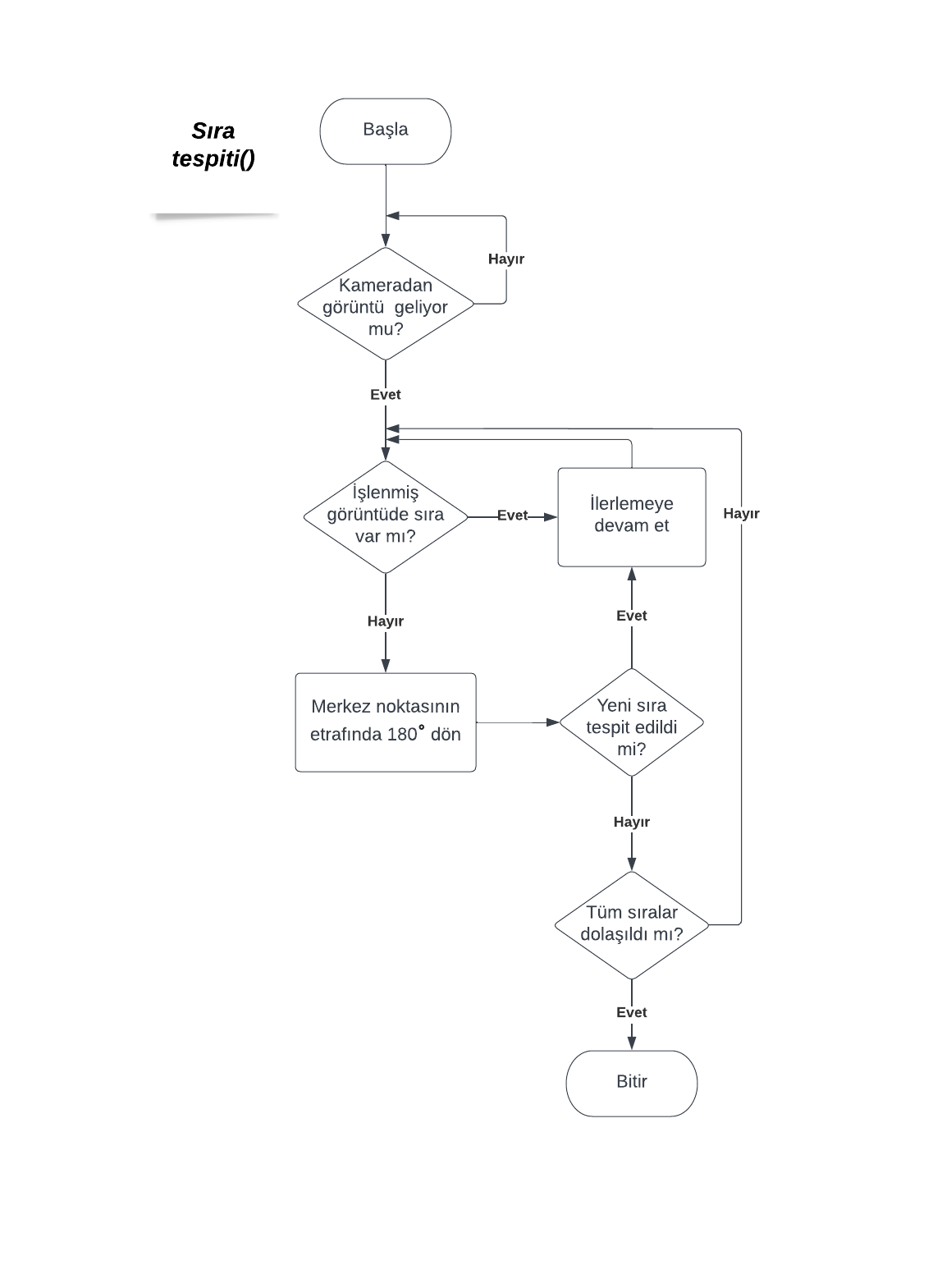
## Genel Algoritma

Yarışma alanına bırakılan aracın belirlenen görevleri dışarıdan müdahale olmadığı durumlarda otonom olarak ya da müdahale gerektiren durumlarda manuel olarak gerçekleştirmesi gerekmektedir. Araçtan beklenen ana görevler belli bir alanda sıralı olarak ekilen bitkiler arasındaki yabani otu belirleyerek yok etmesi, ilerlediği güzergah üzerinde karşılaştığı engelleri aşması, ilerleme sırasında sensörlerden gelen verileri yorumlayarak kullanıcıya bir arayüz aracılığı ile aktarmasıdır. Buna göre araç gerekli kurulumları tamamlanarak yarışma parkuruna bırakıldıktan sonra sensör ve kameradan gelen verileri toplayarak işlemlerine başlayacaktır. Bu sırada aracın ilk yapacağı kontroller parkur alanının sınırlarını belirlemek ve sıra tespit algoritmasını çalıştırmaktır. Bitki hizasında ilerleyen aracın belirlediği rotada herhangi bir engel ya da çukur ile karşılaşması durumunda 50 cm genişliğine kadar ulaşan engellerin ve çukurların üzerinden geçerek aşması, aşılamayan engellerin ise etrafından dolaşarak ilerleyişine devam etmesi gerekmektedir. Bu şekilde parkur boyunca herhangi bir çarpışma olmadan aracın güvenli sürüşünün sağlanması hedeflenmektedir. Aynı şekilde sıra üzerinde ekilmiş olan bitkilerin arasında araca daha önceden tanıtılmış olan yabani otların tespit edilmesi durumunda araç yavaşlatılarak yabani ota ilaç püskürtme yöntemi ile müdahale edilecek ve yok edilen her bir yabani ot için kayıt tutulacaktır. Bunlara ek olarak aracın sensörleriden bir hata alması durumunda acil durum planı devreye girecek ve aracın anlık durdurulması sağlanacaktır. Temel algoritmaların çalışma prensipleri aşağıda daha detaylı açıklanmıştır.



## Sıra Tespit ve Takip Algoritması

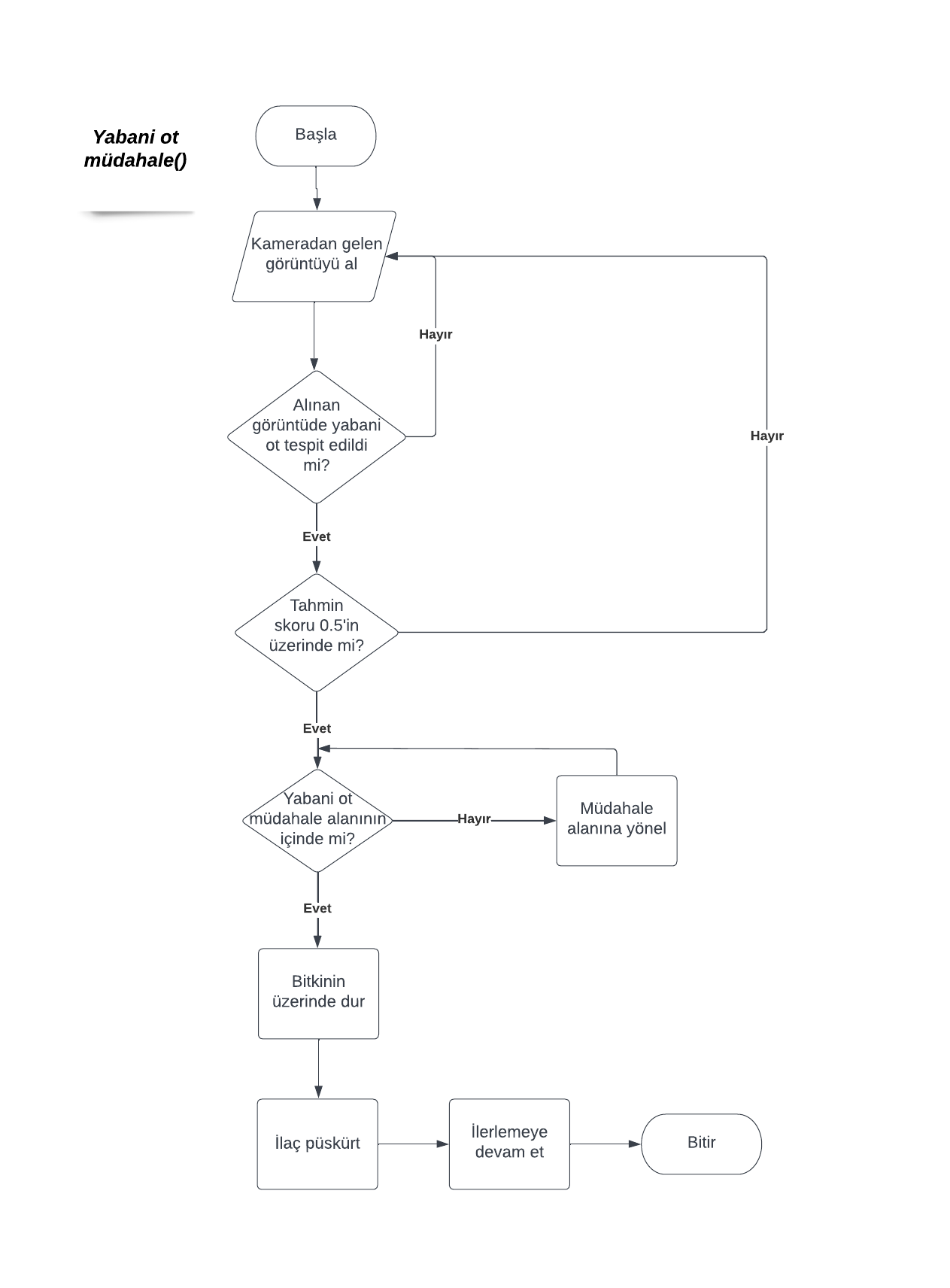
Sıra tespiti ve takibi aracın otonom sürüş algoritmasının temelini oluşturmaktadır. Sıra tespitinin yapılabilmesi için *OpenCV* kütüphanesi kullanılarak Python programlama dilinde yazılım geliştirilmiştir. Aracın ön tarafına yerleştirilecek olan kameradan alınan görüntüye *Hough Line Transform* uygulanır. Kameradan gelen veriye *gray scale* filtresi uygulanıp bir matrikse dönüştürülür. Oluşturulan bu matriksteki tüm değerlere sırasıyla 0 ile 180 dereceleri verilerek piksel yoğunluğu elde edilir, elde edilen piksel yoğunluklarına göre çizgiler tespit edilir. Kameradan gelen görüntüde herhangi bir çizgi tespit edilememişse sıranın sonuna gelinmiştir. Araç sıranın sonuna geldiğinde bir sonraki sıraya geçebilmek için yarıçapı aracın uzunluğunun yarısı kadar olacak şekilde bir hayali çember belirlenir. Daha sonra çemberin merkezi aracın dönme merkezi olarak belirlenip araç bu dönme merkezinin etrafından 180 derece dönerek tekrar sıra tespiti yapmaya başlayacaktır. Bu sayede araç hem 180 derece dönüş hem de çemberin merkezinden döndüğü için bir sonraki sıraya geçmiş olacaktır.



## Yabani Ot Tanıma ve Müdahale Algoritması

Aracın yabani otları algılayıp müdahale algoritmasını işleme alabilmesi için YOLOv5 algoritması üzerinden tanınan nesnelerin tahmin skoru puanı üzerinden değerlendirme yapılması gerekmektedir. Kameradan gelen görüntü üzerinde YOLOv5 algoritması çalıştırıldığında algoritma aldığı her görüntüde (saniyede yaklaşık 24-27 kare ) tanıması gereken yabani otun söz konusu karede olup olmadığını ve varsa kaç tane bulunduğunu belirten bir *backtrace* gönderir. Bu backtrace’lerin de YOLOv5 tarafından hesaplanan 0 ile 1 arasında değer alan bir tahmin skoru bulunmaktadır. Genellikle bu skorun 0.4-0.5 üzerinde olması bizim için daha güvenilir sonuçlar verir. Algoritma bu tahmin skorlarına bakarak söz konusu karede yabani ot bulunma durumunu hata payını en aza indirerek tahmin etmektedir.

Müdahale yöntemi olarak ilaç püskürtme kullanılacak olup kameradan gelen görüntüde herhangi bir yabani ot tespit edildiğinde burası müdahale alanı olarak belirlenecektir. Bu noktada yabani ot tespit edildiğine dair bilgi YOLO algoritmasındaki *detect* kısmından bir komut gönderilerek iletilecektir. Müdahale alanı belirlendiğinde araç ilerlediği sıra üzerinde ilerlemeyi durduracak, bu sayede yabani otun üzerinde durarak ilaç püskürtme mekanizması devreye girecektir. İmha edilen her yabani ot için kod içerisine yerleştirilen sayaç bir artırılarak alandaki yabani ot sayısı kaydedilecektir. Püskürtme işlemi gerçekleştikten sonra araç ilerlediği sıra üstünde hareketine devam edecektir.



## Engel ve Çukur Tespiti Algoritması

Engelden kaçış algoritması olası çarpışmaların önlenmesi ve aracın engelle karşılaştığı durumlarda alternatif bir rota belirleyerek yoluna devam edebilmesi için tasarlandı. Engel tespiti için lidardan gelen veri kullanılarak aracın yakınındaki nesneler belirlenecektir. Bu algoritmanın çalışma mantığında ilk olarak karşılaşılan cismin orta noktası hesaplanacak ve bu orta noktanın ilerlenen hayali sıra düzleminin ne tarafında olduğu belirlenecektir. Engelin orta noktası ilerlenen sıranın ne tarafında ise araç zıt taraftan dolaşacaktır, bu sayede en kısa yol kullanılarak harekete devam edilecektir. Engelin orta noktası ilerlenen hizanın tam üzerinde bulunuyorsa hangi taraftan dolaştığı bir fark oluşturmamaktadır. Aracın dönüş mesafesi ise yine engelin toplam genişliğine göre ayarlanacaktır. Karşılaşılan engel genişliği hayali bir çemberin çap ölçüsü olarak kullanılacak, engelin orta noktası da çemberin merkez noktası olarak varsayılacaktır. Buna göre gerekli hesaplamalar yapılarak aracın dönüş açısı belirlenecek ve hayali çemberin sınırlarından dolaşacaktır. Ek olarak engelin konumu GPS verisi kullanılarak kaydedilecek olup aynı güzergâh üzerinde tekrar dolaşılması gerektiği durumlar için saklanacaktır. Mekanik ekibimizin belirlediği esnek hareket kabiliyetindeki bacak yapısı şartnamede verilen büyüklükteki çukurları aşabildiği için çukurların aşılmasına yönelik ayrıca bir algoritma oluşturulmamıştır.

1. **Yabancı Otla Zirai Mücadele Yöntemleri**

Önceki tasarımımızda kullandığımız peristaltik pompanın basıncı düşük olduğunu ve etkin bölge alanı küçük olduğu için ilaçlama sırasında yapacağımız püskürtme verimliliği düşüktü. Öncelikle kullanacağımız püskürtme yöntemi geniş bir açı ve yüksek bir basınçla püskürtme yapacağından bitkiyi kaçırma ihtimali çok düşüktür. Yeni tasarımımızda bu verim düşüklülüğünü ortadan kaldırmak için 12V-5 bar ilaçlama pompası kullanmaya karar verdik. Kullanacağımız ilaçlama pompasının ise motorunda selenoid valf bağlı olmasının basınç kaybını ve nozuldan sıvı akıntısını önlemesi açısından daha iyi olacaktır. Depo olarak 2 litrelik bir silecek deposu edindik ayrıca kullanacağımız ilacı bir ilaç deposuna koymak yerine ilacın direkt olarak oturabileceği bir kucak yapmayı karararlaştırdık, böylece depodan depoya dökmeye gerek kalmayacaktır. Buradaki kullandığımız sistem suyu daha homojen olarak geniş bir bölgeye dağıtacak ve böylece zararlı bitkilere etkili bir yöntemle direkt yok etme sağlayacaktır.

# Özgün Bileşenler

Aracımızın bütün tasarımları kendimize aittir ve bazı özgün tasarımlarımız şu şekildedir:

## Mekanik

### Tekerlek Tasarımı

Tekerlek için hazırladığımız tasarım tamamen bize özgüdür. Bu tasarımda motorlar jantın içerisinde yer alır. Bu sayede motor dışarıdan gelebilecek darbelere kapalıdır. Tekerlek jantının bir diğer özelliği ise modüler bir yapıya sahip olmasıdır. Böylece aracın durumuna ve kullanılacak ortama göre istenilen şekilde modifiye edilebilir. Motor kabininin ön ve arka kısımlarında havalandırma delikleri bulunmaktadır fakat pasif havalandırmanın yetmediği durumlarda motoru soğutmak amacıyla motor tutucu başlığa ekstra fan takılabilmektedir. Modüler yapının bir diğer özelliği ise motor kısmının istenilen durumlarda contalı kapaklarla su ve toz geçirmez hale getirilebilmesidir.

### Elektrik Kabineti

Seçilen kabinet tasarımı sayesinde elektronik modüllerine kolayca ulaşabilme imkanının yanında içiriye aktif ve pasif yönden hava akışı sağlayabilme yeteneğine sahiptir. Özellikle pleksi levha kullanılması sayesinde ise kullanıcı tarafından araç daima kontrol altında tutulur ve bir arıza durumunda anında müdahale edilebilecek şekilde tasarlanmıştır.

### Rocker Süspansiyon Sistemi

Aracın arazi koşullarında hareket kabiliyetlerinden ödün vermemesi için aracı sarsıntılardan koruyabilecek ve 15 derece eğimde bile devrilmeden stabil bir hareket sağlayabilecek süspansiyon sistemine ihtiyacı vardı. Bu sebeple biz de rocker süspansiyon sistemini seçtik. Bu sistemin en güzel tarafı aracın sağ ve sol kısmı çukur veya tümsek ile karşılaşıldığında eşzamanlı olarak hareket ederek aracın gövdesini minimum düzeyde sallantıya maruz bırakmasıdır. Kullanılan bu sistem tasarladığımız bağlantı parçaları ve kullanılan malzemelere ulaşım kolaylığı açısından sağlam, güvenilir ve diğer süspansiyon sistemlerinden görece daha ucuza mal edilebilirdir. Ayrıca Rocker Süspansiyon Sistemi hali hazırda Mars’a gönderilen gezegen kaşifi araçlarda da benzeri kullanılan bir sistemdir. Tamamen mekanik bir sistem olduğu için sorun çıkarma ihtimali düşüktür, bakımı ve değişimi de kolaydır.

### Sigma Profil

Aracın gövde tasarımında 20x20 sigma profil kullanılmıştır. Bunun sebebi aracımızın gövde kısmında kullanılacak elektronik ve mekanik sistemlerin en uygun şekilde konumlandırılması açısından en uygun yapıya sahip olmasıdır. İçerisindeki geniş kanallar sayesinde ise kablolama sırasında görüntü kirliliğine sebep olmaması açısından sigma profillerin kabloların içerisine yerleştirilmesi planlanmıştır.

### Püskürtme sistemi

Araç yabani otlarla mücadelede herbisit adlı bir kimyasal madde kullanacaktır. Bu madde temas ettiği bitkiye zarar verir ve onu öldürür. Temasın maksimum seviyede tutulması ise herbisitin minimum sarfiyat ve maksimum etkisi için doğru ve usulünce yapılmalıdır. Piyasada pulunan ilaç pompaları bu amaç için özelleştirilmiş nozul sistemlerine sahiptirler. Kullanılan nozulun çapı ve kullanılması için gerekli basınç firmaları üreticiler tarafından çeşitli AR-GE (Araştırma ve Geliştirme) faaliyetleri ve yüksek maliyetli fonlamalar ile belirlenmiştir. Çünkü bu işlem doğru bir şekilde yapılamazsa işlem sırasında istemeden zirai bitkilere de zarar verilebilir. Biz de aracımızda bu beklentilere sahip olduğumuz için tasarımımızda piyasada bulunan standart bir nozulu ve elektronik ilaç pompalarında da kullanılan bir basınçlı sıvı pompasını kullandık. Kullandığımız sıvı pompası içerisindeki basınç sensörü yardımıyla sıvının çıkış basıncını daima 5 bar olacak şekilde ayarlamaktadır. Pompa ve nozul arasında ise bir adet selenoid valf kullanılarak sıvı giriş çıkışı kontrol edilmektedir. Uçtaki nozul ise püskürtme açısı ayarlanabildeceğinden dolayı istenilen açıda püskürtme yapılabilmektedir.

## Yazılım

Kontrol masasından kullanıcıya sunulacak olan arayüz sistemi tamamen ekibimizce tasarlanmıştır. Tasarım ve yazılımı QtDesigner aracı kullanılarak Python dilinde hazırlanmıştır. Kullanıcıya iletilecek olan veriler, arayüz tasarımının gereklilikleri ve kullanım kolaylıkları düşünülerek belirlenmiştir.

Aracın ilerlediği alanın bilgisini almak kullanıcının faydasına olacağı için haritalama işlemi özgün bir bileşen olarak kullanıcı arayüzünde yerini almıştır. Araç görevlerini yerine getirirken bu haritada GPS ve diğer sensörlerden gelen verilerin birleştirilmesi sonucu elde edilen aracın anlık konumu, imha edilen yabani otların konumu ve parkur alanının sınırları gösterilecektir. Son olarak aracın herhangi bir engel ile karşılaştığında ve ilaç püskürtme mekanizması devreye girdiğinde farklı ikaz sesleri çalması özgün tasarımımızın bir başka ögesidir.

## Elektrik

Motor sürücü kartını ve voltaj regülatörü kartımızı kendimiz tasarlayıp ürettik. Motor sürücü kartımızda MOSFET’lerimizi H-bridge mantığını kullanarak sürmektir. Bu işlemde mosfet sürücü olarak IR2104 MOSFET sürücüsünü kullandık. Voltaj regülatör kartımızda ana mantığı Buck Converter’ a dayanmaktadır. LM2576 kullanarak oluşturulmuştur. LM2576’ yı kullanmamızdaki sebep ise çekilen akım arttığında oluşacak olan voltaj düşümüne karşı kendi içinde feedback içermesidir. Ayrıca çizdiğimiz devrede çok turlu trimpot kullanarak regülatör devremizi aracımızın herhangi bir yerinde kullanma esnekliğini sağlamış oluyoruz.

# Güvenlik Önlemleri

Tekerlek yapısında gelen kum, toprak, taş ve ilacın motora girmesini engellemek adına ve motorun aşırı ısınması durumunda eklenebilecek mini fan için dış bölüme eklenebilecek bir kapak vardır. Su akış sensörü ile beraber aracın püskürtme alanında net göstermek amacıyla bir adet sensör eklenmiştir. Aracımızda bir acil durum stop butonu bulunmaktadır. Basıldığı takdirde aracın iletişimini kesmeyip sadece hareketini durdurmaktadır. Araç zararlı otu tespit ettiği zaman ilaçlama yapmadan önce çevredeki insanları uyarmak için ses çıkarmaktadır ve aynı zamanda da döner led yanmaktadır.

1. **Simülasyon ve Test**

## Mekanik Testler

### Jant Analizi

#### Toplam Deformasyon Analizi

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

#### Stres Analizi

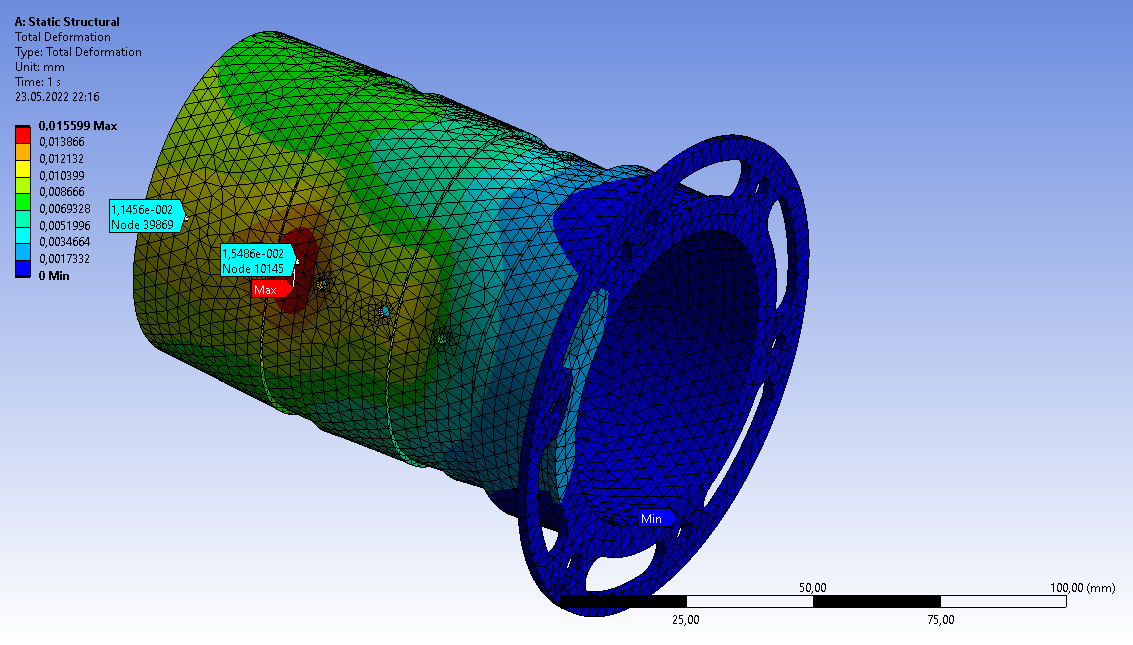
metin, zincir içeren bir resim

Açıklama otomatik olarak oluşturuldu

Aracımızın jant göbeğinin mevcut tasarımı ile yaptığımız analizlerde 15 üstü bir güvenlik katsayısı ile görevini sorunsuzca yerine getirebileceği ispatlanmıştır.

### Motor Yatağı Analizi

#### Toplam Deformasyon Analizi



#### Stres Analizi

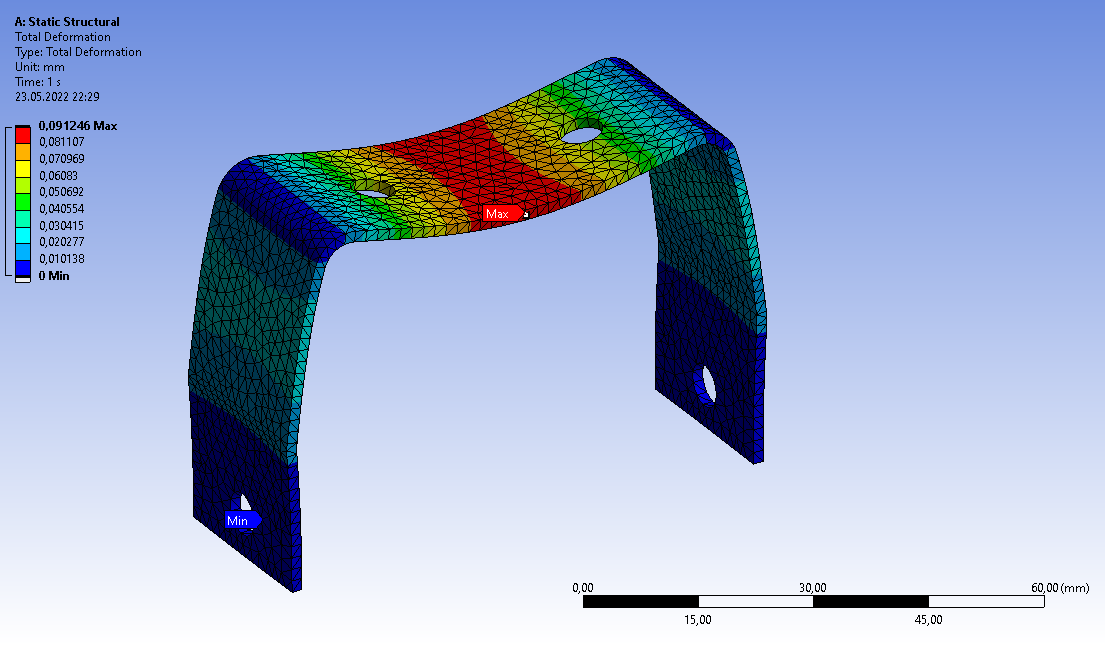
metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

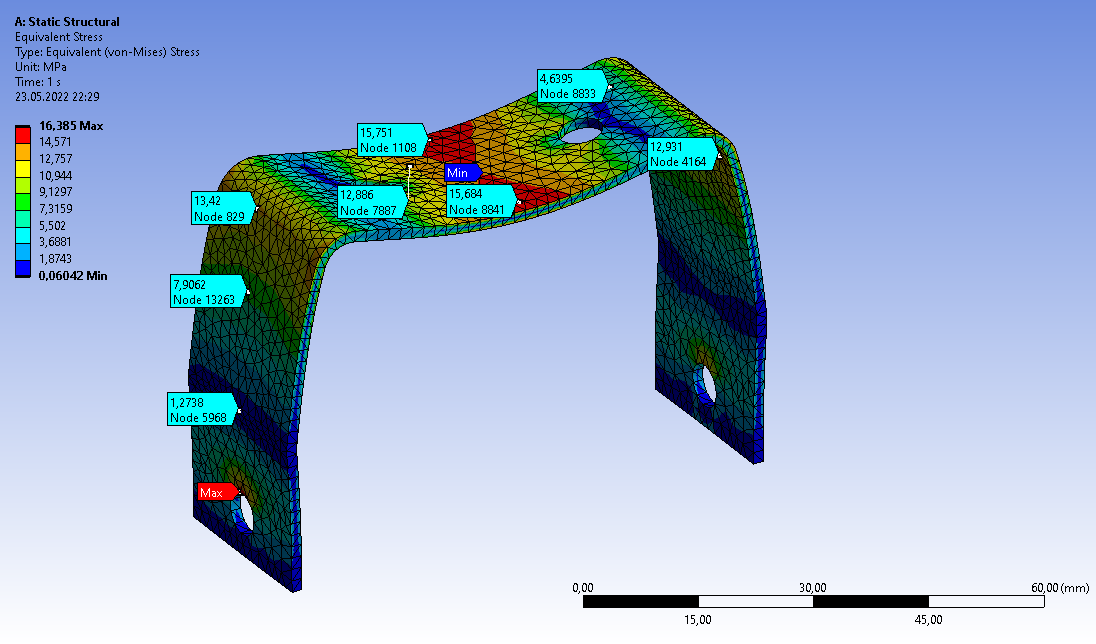
Aracımızın motor yatağının mevcut tasarımı ile yaptığımız analizlerde 11,947’lik bir güvenlik katsayısı ile görevini sorunsuzca yerine getirebileceği ispatlanmıştır.

### Jant Kolu Analizi

#### Toplam Deformasyon Analizi



#### Stres Analizi



    Aracımızın jant kolunun mevcut tasarımına yaptığımız analizlerde 15 üstü bir güvenlik katsayısı ile görevini sorunsuzca yerine getirebileceği ispatlanmıştır.

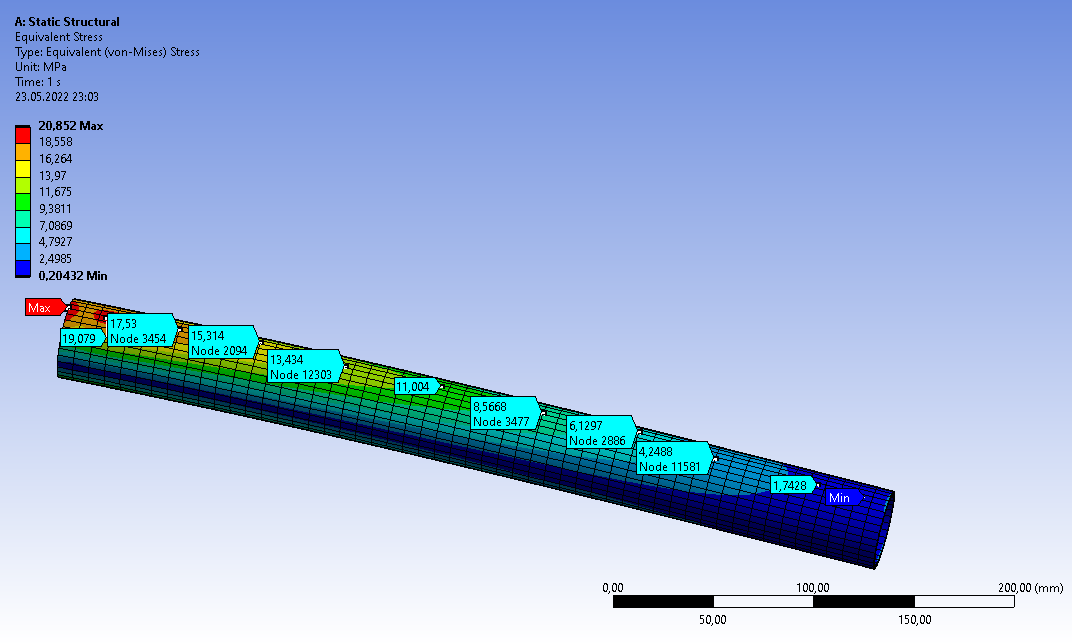
### Bacak Silindir Profil

#### Toplam Deformasyon Analizi

metin, yazı gereçleri içeren bir resim

Açıklama otomatik olarak oluşturuldu

#### Stres Analizi



 Aracımızın bacak silindir profilinin mevcut tasarımı ile yaptığımız analizlerde 13,428’lik bir güvenlik katsayısı ile görevini sorunsuzca yerine getirebileceği ispatlanmıştır.

## Yazılım Testleri

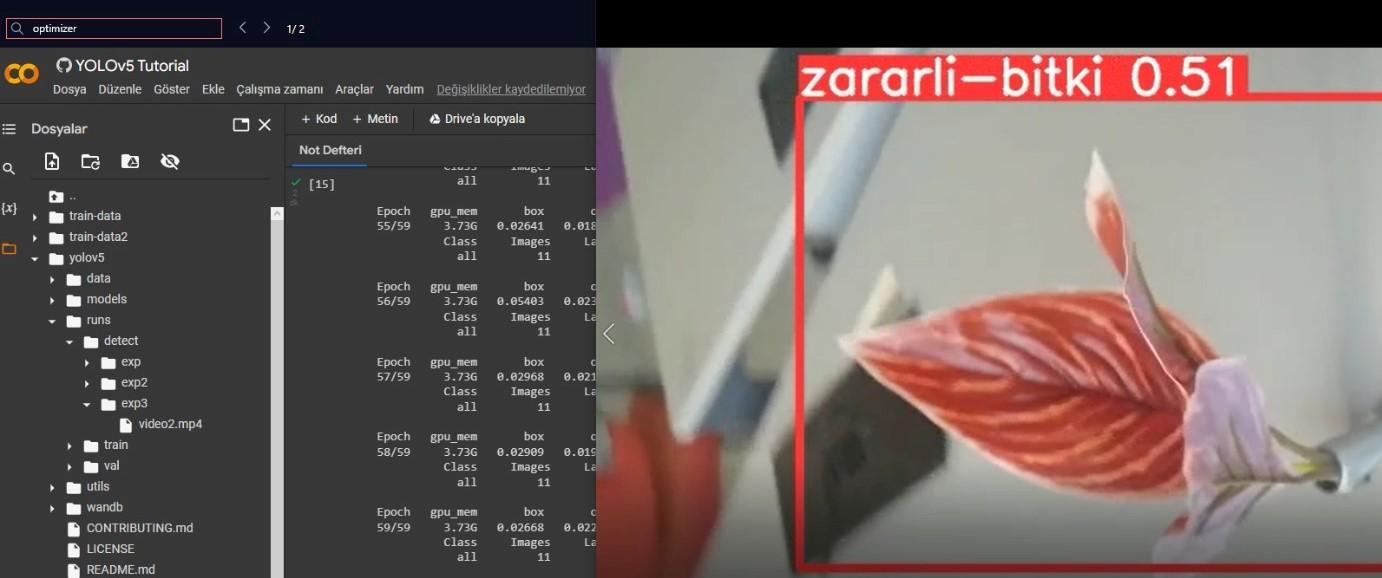
Projenin geliştirilme aşamasında ROS’un sunduğu Gazebo simülasyon ortamında üretim öncesi testler gerçekleştirilmiştir. Parkur şartları göz önünde bulundurularak hazırlanan ROS test ortamında *turtlebot3* simülasyon robotu kullanılarak aracın yapması beklenen hareketler sanal olarak gözlemlenmiştir. Gazebo’da birçok sensör hazır paketler olarak kullanıcıya sunulmaktadır. Testi yapılacak olan sensörün düğümünün çalıştırılması simülasyon ortamındaki robota o sensörün eklenmesi için yeterlidir. Örneğin; lidardan gelen veriler ile çıkarılan ortam haritası ROS görselleştirici aracı olan RViz üzerinden gözlemlenebilmektedir. Ek olarak kameradan elde edilen görüntü hızı ve kalitesi ayrı bir bilgisayar üzerinden test edilmiştir.

Basit formda geliştirilen bir arayüz uygulaması oluşturularak hem aracın durum bilgilerine dair anlık verinin çekilip çekilemediği hem de kullanılacak arayüzün testi gerçekleştirilmiştir.

Sıra tespiti için örnek görüntüler üzerinden testler yapılmıştır. Algoritmaya verilen görüntüde belirgin çizgiler halinde ekim yapılan arazide sıra tespiti gerçekleştirilmiştir. Ancak arası seyrek olarak ekim yapılan arazide sıra tespiti başarılı sonuç vermemiştir. Sıra tespit kesinlik oranını artırmak amacıyla halen geliştirmeler yapılmaktadır.



Görüntü işleme testlerine ilk olarak yabani otların tespitinde hangi yöntemin kullanılacağını belirlemek amacıyla farklı model eğitme algoritmaları denenerek başlanmıştır. Denemelerimiz sonucunda R-CNN ile %61 doğruluk oranı ve 6-13 arası FPS değeri, YOLOv3 ile %64 doğruluk oranı ve 8-15 arası FPS değeri, YOLOv4 ile % 73 doğruluk oranı ve 17-25 arası FPS değeri, YOLOv5 ile ise %81 doğruluk oranı ve 24-27 arası FPS değerleri elde edilmiştir. Bu nedenle en verimli ve en hızlı sonuç alabildiğimiz YOLOv5 algoritmasını kullanmakta karar kıldık. (Tüm algoritmalar objektif oranlar vermesi amacıyla aynı veri seti ile test edilmiştir.) YOLOv5 algoritması kullanırken görüntüyü canlı olarak kamera üzerinden almak için Python dilinde OpenCV kütüphanesini kullanıldı. YOLOv5 algoritmasını veriyle beslemek amacıyla bitkinin farklı açılardan 60’tan fazla fotoğrafı çekilerek model eğitiminde kullanıldı. Test görüntülerinin düzenlenmesi için ise *LabelImg* modül paketi kullanıldı. *Google Colab* üzerinden Python’da *YOLOv5-Master, PyTorch* modülleri ile modelin eğitimi tamamlandı.



Geliştirme sürecinde yapılan simülasyon testleri sırasında mevcut araç yazılımının yetersiz kaldığı tespit edilerek algoritmalarda iyileştirme ve geliştirmeye gidilmiştir. Ayrıca bazı sensör verilerinde hata payının yüksek olduğu tespit edilmiştir. Bu durum aracın simülasyon testlerinde yetersiz kalmasının önemli sebeplerinden biri olarak belirlenmiştir. Bu konuyla ilgili geliştirmeler devam etmektedir.

# Referanslar (Kaynakça)

* https://www.metalreyonu.com.tr/urun/aluminyum-boru-dis-cap-25-mm-et-kal-2-mm
* https://metalavm.com/aluminyum-boru-40-mm
* https://www.hepsiburada.com/orac-akulu-su-aktarma-ve-ilaclama-motoru-pompasi-12-volt-orac-p-HBV000014U0RG
* https://www.robitshop.com/urun/solenoid-su-valfi-12v-solenoid-valve-1-2-inch?gclid=Cj0KCQjwpv2TBhDoARIsALBnVnksDf0tUcnKuf2\_sGHpUmyOQnyqFIbEfuwuwcWdfow1w7fvbjzP6YYaAr5AEALw\_wcB
* <https://www.slamtec.com/en/Lidar/A2>
* http://wiki.ros.org/ROS/Tutorials
* https://husarion.com/tutorials/ros-tutorials/6-slam-navigation/
* https://acikerisim.uludag.edu.tr/bitstream/11452/11730/1/621649.pdf
* https://medium.com/deep-learning-turkiye/otonom-araclardaki-derin-ogrenme-mantigi-nasil-calisir-9f0fb59ba0a5
* https://www.youtube.com/watch?v=tFNJGim3FXw
* <https://www.smartera.com.tr/gercek-zamanli-nesne-takibireal-time-object-detection-w-yolo-python/>
* https://evrimagaci.org/lidar-teknolojisi-nedir-nasil-calisir-nerelerde-kullanilir-10214
* https://cbsakademi.ibb.istanbul/imu-nedir/
* https://droneyap.com/viewtopic.php?t=1775#:~:text=Lipo%20pilimiz%2040A%20verebilmesini%20%27%27C,5.2\*10%20%3D%2052A%20%C3%A7ekebiliriz.
* <http://www.robotiksistem.com/tork_nedir_tork_hesaplama.html>
* <https://www.elektrikrehberiniz.com/elektronik/regule-devresi-14633/#:~:text=Reg%C3%BCle%20devresi%3A%20Gerilimin%20belirli%20s%C4%B1n%C4%B1rlar,s%C4%B1n%C4%B1rlar%20i%C3%A7inde%2C%20otomatik%20olarak%20ayarlar>.
* <https://ozdenercin.com/2019/01/25/i2c-seri-haberlesme-protokolu/>
* <https://www.veribilimiokulu.com/nvidia-jetson-nano-1/#:~:text=NVIDIA%20Jetson%20Nano%20kart%C4%B1n%C4%B1n%20%C3%BCzerinde,toprak%20(GND)%20giri%C5%9Fi%20bulunmaktad%C4%B1r>.
* <https://arduinodestek.com/arduino-ile-acs712-akim-sensoru-kullanimi/>
* <https://www.f1depo.com/urun/22-2v-22000mah-25c-lipo-batarya-6s-jetfire-pil>
* https://www.arduinomedia.com/arduino-ile-dht11-sicaklik-ve-nem-sensoru-kullanimi/