TEKNOFEST

HAVACILIK, UZAY VE TEKNOLOJİ FESTİVALİ

TARIMSAL İNSANSIZ KARA ARAÇLARI

YARIŞMASI

KRİTİK TASARIM RAPORU ŞABLONU

TAKIM ADI

Creatiny Adana

PROJE ADI

Cewd-Ot

BAŞVURU ID

37022

İçindekiler

1. Takım Organizasyonu 3
1.1. Takım Üyeleri 3
2. Ön Tasarım Raporu Değerlendirmesi 4
2.1. Yapılan Değişiklikler 4
2.1.1. Yürür Sistemi 5
2.1.2. Denge Sistemi 5
2.1.3. İlaç Püskürtme Sistemi 6
2.1.4. Elektronik Sistemler 7
2.1.5. Görüntü İşleme Kartı 7
3. Araç Özellikleri 7
3.1. Mekanik Özellikler 7
3.2. Elektronik Özellikler 9
3.3. Yazılımsal Özellikler 12
4. Sensörler 12
5. Araç Kontrol Ünitesi 14
6. Otonom Sürüş Algoritmaları 16
6.1. Otonom Yol Takibi 16
6.2. Yabani Ot Tespiti 17
7. Özgün Bileşenler 22
7.1. PID Araç Dengeleme Sistemi 22
7.2. Raylı İlaç Püskürtme Sistemi 23
8. Güvenlik Önlemleri 24
9. Simülasyon ve Test 25
10. Referanslar 27

TEKNOFEST HAVACILIK, UZAY VE TEKNOLOJİ FESTİVALİ

1. Takım Organizasyonu

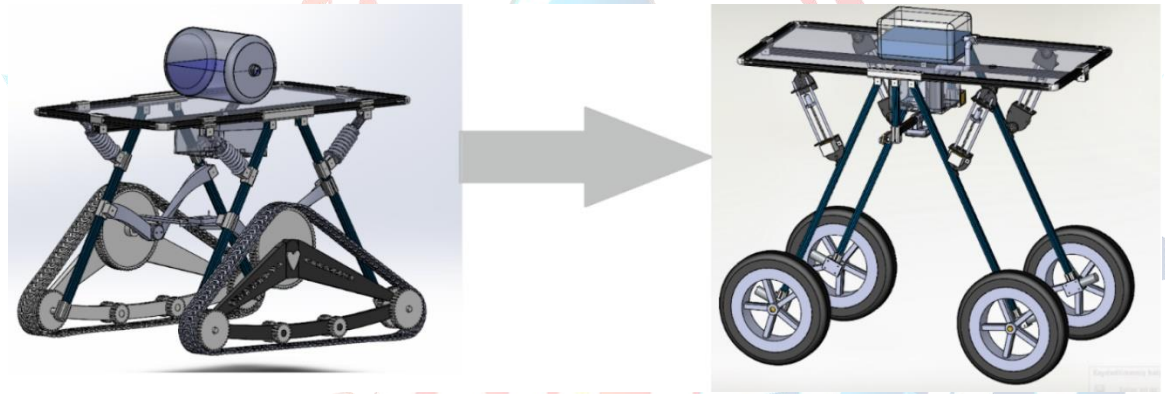
TEKNOFEST yarışmasında Çukurova Üniversitesi Tarım Makineleri Mühendisliği bölümünden bir danışman hoca ve Bilgisayar Mühendisliği ve Tarım Makineleri Mühendisliği bölümlerinde okuyan altı öğrenci ile lisans düzeyinde katılım sağlanacaktır.

2. Ön Tasarım Raporu Değerlendirmesi

Ön tasarım raporu aşamasından itibaren, daha önce belirlenen takvim çerçevesinde gerekli tasarım ve test adımları uygulanmıştır. Aracın kontrol sisteminde, mekaniğinde ve nesne tanıma sisteminde köklü değişikliklere gidilmiştir. Proje ayakları yere basan ve uygulanabilir hale gelmiş ve üretimden önceki son tasarıma karar verilmiştir. Bu aşamadan sonra yürütülecek sponsorluk çalışmaları ve Teknofest tarafından sağlanacak maddi destekler ile üretim aşamaları yürütülecektir.

Ön tasarım aşamasında belirlenen tahmini bütçe 3200 TL'dir. Kritik tasarım aşamasına gelindiğinde yapılan değişikliklerle beraber eklenen motorlar, Jetson Nano ve diğer küçük sistemler göze alındığında tahmini bütçe 8000 – 8200 TL arasındadır. Ön tasarım raporu hazırlanırken belirlenen bütçenin gerçekçi olmadığı anlaşılmıştır.

2.1. Yapılan Değişiklikler



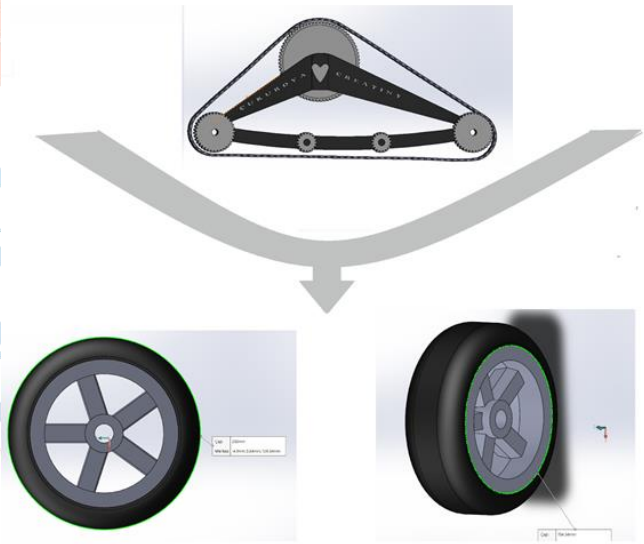
Şekil 1 Tarımsal IKA ÖTR Aşamasından KTR Aşamasına Değişimler

TEKNOFEST HAVACILIK, UZAY VE TEKNOLOJİ FESTİVALİ

2.1.1. Yürür Sistemi

Ön tasarım aşamasında aracın yürürünün paletli bir sistemle sağlanması kararlaştırılmıştı ve bu doğrultuda hesaplama ve tasarımlar yapıldı. Yapılan tasarımın üretim aşamaları düşünüldü ve yapılan plan neticesinde üretimin zorluğu sebebiyle alternatif yollar arandı. Palet sisteminin üretiminde yaşanabilecek teknik sıkıntılar sebebiyle oluşabilecek risklerin, palet sisteminin yol kavrayışında ve torkta sağlayacağı faydalardan daha fazla olduğuna karar verildi.

Üretim kolaylığı, yol tutuşu ve sağlayacağı tork göz önüne alınarak yapılan yeni tasarımda aracın yürürü birbirinden bağımsız 4 adet redüktörlü DC motor sağlanmaktadır. Motorların bağlı olduğu ayaklar oynar şekilde yerleştirilmiştir ve tasarlanan lineer aktüatör sistemi ile tekerler arası mesafe ve eğim değiştirilebilmektedir. Bu sayede tablanın her zaman düz tutulması sağlanarak görüntüde oluşabilecek titreşim önlenmiştir.



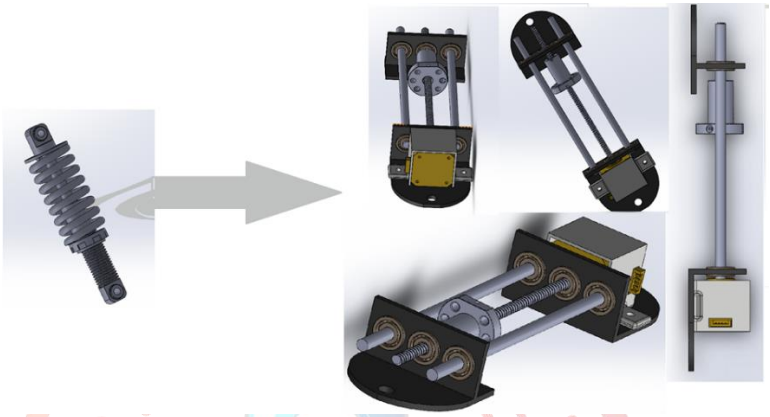
Şekil 2 Yürür Sisteminin Değişimi

2.1.2. Denge Sistemi

Ön tasarım aşamasında aracın dengesinin yerleştirilecek 4 adet amortisör ile sağlanması kararlaştırılmıştı. Bu amaçla aracın dengesini hızlı tepkilerle sağlayabilecek amortisör sistemleri araştırıldı ancak yapılan araştırmalar neticesinde aracın ağırlığı ve gideceği hız göze alındığında amortisör sisteminin yeterli faydayı sağlamayacağı kanısına varıldı. Denge sisteminde asıl hedeflenen fayda, kamera titreşimini azaltarak stabil bir görüntü elde ederek etkin bir görüntü işleme sistemi oluşturmaktır. Bu sebeple, hızlı tepki verebilen ve elektronik olarak kontrol edilebilecek bir denge sistemine ihtiyaç doğmaktadır.

Araştırmalarımız sonucunda PID fonksiyonu kullanarak otomatik ve hızlı dengeleme sağlayacak bir sistem tasarlanmıştır. Sistem mekanik olarak, takımımız tarafından tasarlanan 4 adet lineer aktüatör ile çalışmaktadır. Lineer aktüatörler 4 bacağa eğimli olarak bağlanmıştır ve gerektiğinde PID fonksiyonları doğrultusunda üst tablanın bacağa olan uzaklığını

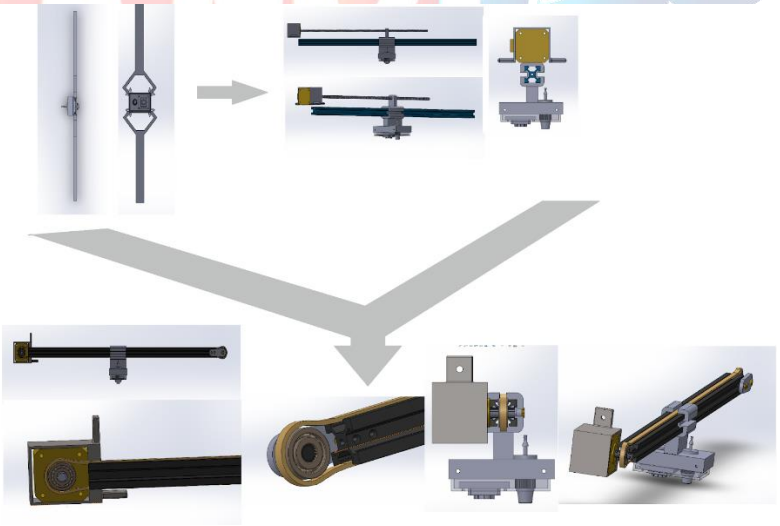
değiştirerek tablanın zemine paralel kalabilmesini sağlar. Lineer aktüatörler step motorlarla çalışır. Gerekli PID algoritmalarının koşturulması ve motorların sürülmesi işi Arduino Uno geliştirme kartı ile yapılmaktadır.



Şekil 3 Lineer Aktüatör Tasarımı

2.1.3. İlaç Püskürtme Sistemi

Ön tasarım aşamasında belirlenen püskürtme sistemi, ilacın yer çekimi kuvveti ile meme ucuna aktarılması fikrine dayanarak çalışmaktaydı ancak yapılan hesaplamalar bu yöntemin ilacın istendiği hız ve miktarda akması için yetersiz olduğunu gösterdi. Bu sebeple püskürtme sistemi için aktif olarak basıncı kontrol edilebilecek bir sistem tasarlandı. Tasarlanan sistem, ilaç haznesinin basıncını kullanılan mini kompresör ile belli bir seviyede tutmaktadır. Basınçlı ilacın meme ucuna yollanması için ise ilaç haznesinin çıkışına bağlanan bir solenoid valf kullanılmaktadır.



Şekil 4 İlaçlama ve Ray Sistemi

2.1.4. Elektronik Sistemler

Ön tasarım raporu aşamasında aracın yabani ot tanıma fonksiyonu hariç tüm fonksiyonları STM32F103 mikroişlemci üzerinde koşmaktaydı. Araca yeni eklenen denge sistemi ile karmaşıklaşan sistemde, zamanlama kontrolünün ve hata ayıklamanın zorlaşması sebebi ile STM yetersiz kaldı. Bu sebeple Sistemden bağımsız çalışacak işler bağımsız işlemcilere bölündü. Görüntü işleminin Jetson Nano ile, Denge kontrolü fonksiyonlarının Arduino Uno ile, Jetson Nano üzerinden UART ile gelen komutlara göre püskürtme ucunun hareketinin sağlanmasının Arduino Nano ile ve Otonom hareketin sağlanmasının STM32F103 ile koşturulmasına karar verildi.

2.1.5. Görüntü İşleme Kartı

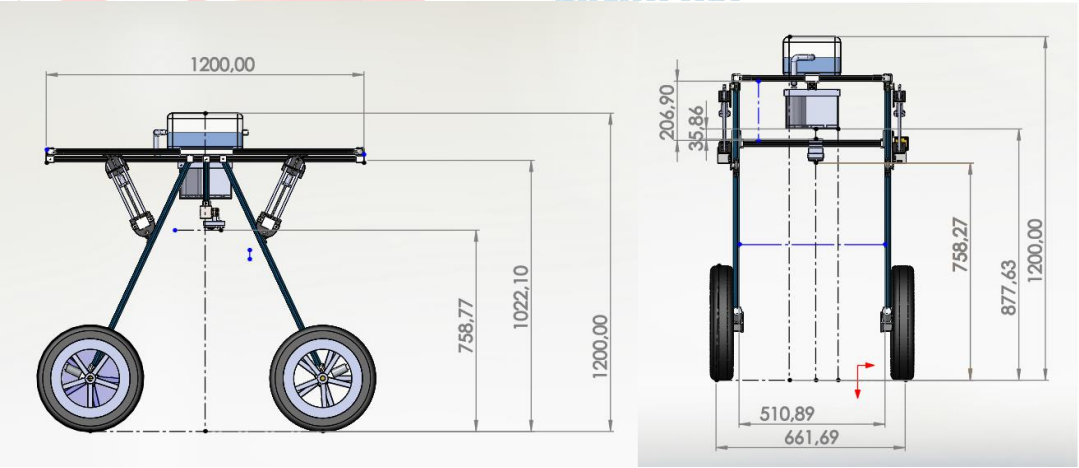
Aracın yabani otları tanıyabilmesi için gereken işlem gücünün Raspberry Pi 3b+ geliştirme kartı ile sağlanması planlanmaktaydı. Raspberry Pi 3B+ ile yapılan testler sonucunda alınan FPS değerleri beklenenin çok altında geldi. Aracın gerçek zamanlı görüntü işlemesini engel olabileceği için farklı bir çözüm arayışına gidildi ve görüntü işleme kartı olarak Jetson Nano kullanılmasına karar verildi. Jetson Nano, yapay zekâ işlemlerini hızlandıran özel çekirdekler barındırması sayesinde çok daha yüksek FPS değerlerinde görüntü işleyebilmektedir.

3. Araç Özellikleri

3.1. Mekanik Özellikler

Tarımsal insansız kara aracı, tarım alanlarında engebeli arazi şartlarında, toprak yollarda çalışacaktır. Bu sebeple aracın tür zor koşullara uygun malzemelerden üretilmesi ve toprak yolda gidebilecek güçte olması gereklidir.

Aracı sağlam, hafif, güçlü ve düşük maliyetli tasarlamaya çalışmak tasarım aşamasında zorluklar yarattı ve bu sorunlara çeşitli çözümler üretildi. Ön tasarım raporu aşamasında yapılan tasarımda belirlenen sorunları çözebilmek için araç yeniden tasarlandı ve üretime uygun hale getirildi.



Şekil 5 Araç boyutları ve Genel Görünüşü

Aracın sağlam, hafif olması ve kolay imal edilmesi amacıyla genel aksamı Alüminyum Sigma profil ile tasarlanmıştır. Aracı tasarımı mümkün olduğu kadar küçük boyutlu olarak yapılmaya çalışıldı. Aracın son tasarımında belirlenen genişlik, yükseklik ve uzunluğu; 661.69 x 1200 x 1200 olarak belirlendi. Aracın ölçüleri, kültür bitkilerine zarar vermemek, kameranın görüş açısını genişletmek ve püskürtme için gerekli optimum mesafeyi sağlamak için kültür bitkisi maksimum uzunluğu, kamera lensinin görüş açısı ve ilaç memesinin basıncı göz önünde bulundurularak belirlendi.

Araç, dört adet redüktörlü motor ve birbirinden bağımsız dört adet 40 cm çap, 7 cm genişlikte tekerlek ile hareket etmektedir. Aracın bağımsız tekerleklerle tasarlanmasının sebebi çeşitli arazi koşullarında tekerler arası mesafeyi ayarlayarak aracı stabil tutmaktır.

Aracın ayak kısmında, birbirinden bağımsız olan dört farklı linear aktüatör sistemi sayesinde aracın üst tablasının her zaman zemine paralel tutulması hedeflendi. Bu sayede aracın süspansiyon sistemi mekanik olarak kontrol edilecek bununla beraber, araçta bulunan her bir ayak için ayrı şekilde uzatılıp kısaltarak denge en hassas biçimde sağlanacaktır.

Yabani otlarla mücadele için aracın alt kısmında, yerden 758mm uzakta bulunan bir ray sistemine sabit kamera ve püskürtme memesi sistemi bulunmaktadır. Konum seçimi sayesinde, kamera açısı olabildiğince geniş tutulmuş ve ilaç püskürtme sisteminin doğrusal bir püskürtme sağlaması sağlanmıştır. İlaç memesi, kamera ile hareket ettiği için görüntü işleme aşamasında bitkinin konumunun tespiti işi kolaylaşmıştır. Kamera her zaman itkiyi merkezine almaya çalışacaktır.

Elektronik devre ve aksesuarları, aracın alt kısmında bulunan ve yerden yüksekliği 877 mm olan kapalı bir kutu içerisinde konumlandırılacaktır. Bu sayede elektronik aksesuarlara zarar verebilecek darbe, ısı, sıvı sıçrama gibi tehlikeler önlenmiş olacaktır.

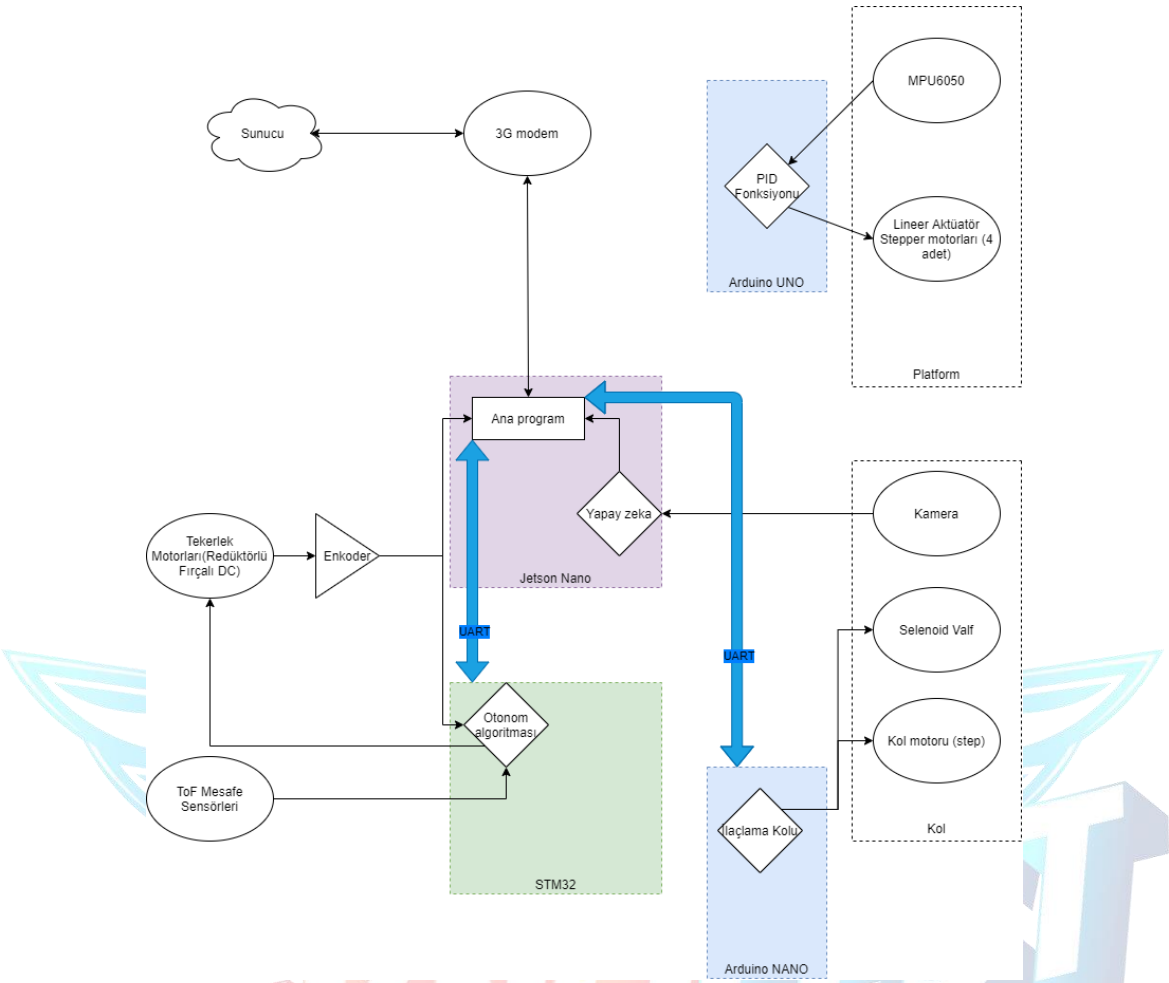
İlaç, aracın üst kısmına yerleştirilen depo bölümünde saklanmaktadır. Deponun aracın üst kısmında merkezde yer almasının sebebi, aracın dengesinin sağlanmasıdır. Depodaki ilacı püskürtmek için gerekli basıncı sağlamak için mini kompresör kullanılmıştır. Kompresör ilaç deposunun basıncını sabit tutmaya çalışır, Solenoid valf ve hortum ile meme ucuna bağlanan ilaç deposundaki ilaç istenildiğinde elektronik olarak valfin açılması ile püskürtülür.

Şekil 6 Aracın Çeşitli Açılardan Görünümü

3.2. Elektronik Özellikler

Tasarlanan araç 800 metrekarelik bir alanı 30 dakika altında tarayabilecek hız ve pil özellikleriyle donatılmıştır. Görev süresi içinde alanı tamamen tarayarak, yapay zekâ destekli nesne tanıma algoritması ile zararlı otları tarayabilmekte ve ilaçlama yoluyla yok edebilmektedir. 4S4P konfigürasyonunda bağlı 18650 pillerden güç alan sistem 13.8Ah kapasiteye sahiptir.

TEKNOFEST
HAVACILIK, UZAY VE TEKNOLOJİ FESTİVALİ

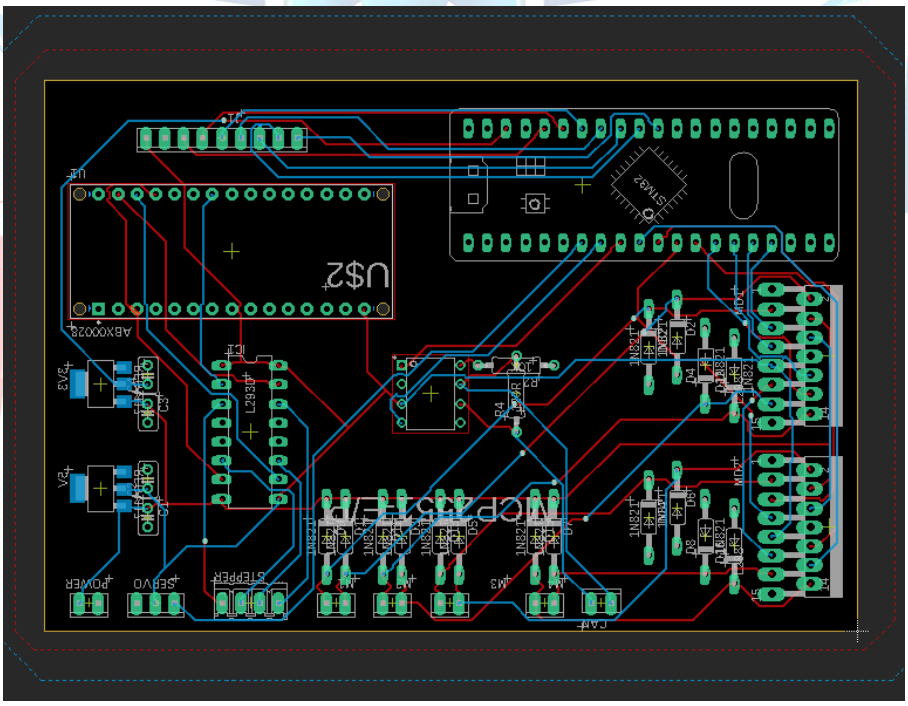


Şekil 7 Elektronik Sistem Akış Diyagramı

Aracın haberleşme ve kontrol ihtiyaçları için STM32F103 mikroişlemci tabanlı bir devrenin yanında bir adet Arduino Nano ve Arduino UNO bulunmaktadır. STM32F103 aracın otonom algoritmasının koşturulmasından, seri haberleşmenin çalışmasından, sensörlerin ve redüktörlü motorların kontrolünden sorumludur. Aracın hareketi yüksek tork sağlayan redüktörlü motorlarla sağlanmıştır. Bu motorlardan araçta her tekerlek için bir tane olmak üzere toplam dört tane bulunmaktadır. Ayrıca her motor için bir tane olmak üzere toplam dört adet encoder de mevcuttur. Encoderler, bağlı oldukları motorların şaftları bir devir tamamladığında sinyal olarak bir (1) yollamaktadırlar. STM32, bu devir sayısı ve şaftların çevre uzunluğu ile aracın hızını hesaplamaktadır. Araçta üçü önde, ikisi yanlarda olmak üzere beş adet ToF uzaklık sensörü mevcuttur. Bu sensörler STM32'ye etrafta bir engel varsa aracın bu engele mesafesini gönderir. STM32 bu mesafeler sayesinde aracın önüne bir engel çıktığında bunu anlayıp algoritmasının engelin etrafından dolaşmak için tasarlanmış kısmını

çalıştırıyor. Yine sensörlerin gönderdiği verilere göre de aracın taradığı arkın bitip bitmediğini anlıyor ve buna göre algoritmanın diğer arka geçmek için olan kısmını çalıştırıyor. Tasarlanan kontrol kolunu ise Arduino Nano, servo motorlar aracılığıyla kontrol etmektedir. Arduino Nano, kamerayı yabani otlara ortalaması için gereken komutları Jetson Nano'dan UART aracılığıyla almaktadır. Sistemin kalanından ayrık olarak çalışan Arduino Uno ise aracın sensörlerinin ve kamerasının bağlı olduğu platformun eğimini ve yüksekliğini kontrol etmektedir. Platforma bağlı olan MPU6050 sensörü bize platformun eğimini vermektedir. Bu veri Arduino Uno'da çalışan Platform Dengeleme Sistemi'ne gönderilip burada PID fonksiyonunda kullanılmaktadır. Tüm elektronik sistemin acil durumlarda durdurulmasını sağlayan bir adet acil durum butonu mevcuttur.

Sistemin STM32F103 ve Arduino Nano içeren bölümü ile ilaçlama kolu ve redüktörlü motorları kontrol eden kısımları için bir devre kartı geliştirilmiştir. Bu sayede çok daha küçük alanda daha güvenilir bir tasarım yapılması sağlanmıştır. Şekil 8'te görüldüğü gibi, tasarlanan devre 5- 19 Volt arası besleme girişleri ile çalışabilmektedir. Devreye gelen ana besleme 5 Volt ve 3.3 Volt voltaj regülatörleri ile regüle edildikten sonra STM32 ve Arduino kartlarını Besler. Motorlar ise doğrudan gelen giriş voltajı ile beslenir.



Şekil 8 Kontrol Devresi Çizimi

3.3. Yazılımsal Özellikler

Aracın üzerinde dört ayrı yazılım koşmaktadır. Jetson Nano üzerinde koşan Python kodu, önceden eğitilmiş yapay zekâ modelini kullanarak bitkinin konumunu tespitle sorumludur. STM32 üzerinde aracın otonom hareket algoritması koşar. Bu algoritma C++ ile yazılmıştır. Arduino Nano, sadece Jetson'dan UART protokolü ile gelen komutlara göre ilaçlama kolunun step motorlarını hareket ettirmekle görevlidir. Üzerinde C++ kodu koşmaktadır. Arduino UNO ise tüm sistemden bağımsız olarak MPU6050 sensöründen gelen verileri işleyerek PID algoritmaları ile aracın üst plakasını sabit tutmaya çalışmaktadır. 4 adet step motor sürer, içinde C++ kodu koşmaktadır.

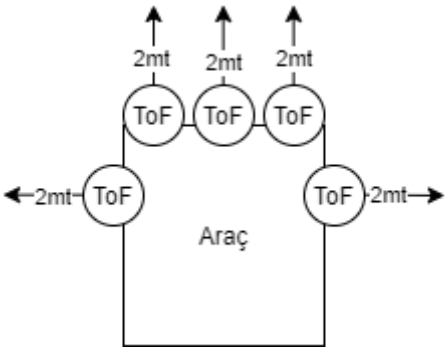
Aracın uzaktan kontrol arayüzü için web tabanlı bir sistem geliştirilmiştir. Veriler 3G modem aracılığı ile Jetson Nano'dan Firebase sunucuya yazılır. Ayrıca Jetson Nano içerisinde bir IPCamera sunucusu barındırır. Angular ile kodlanan web arayüzü ise bu verileri görselleştirerek kullanıcıya sunar.

4. Sensörler

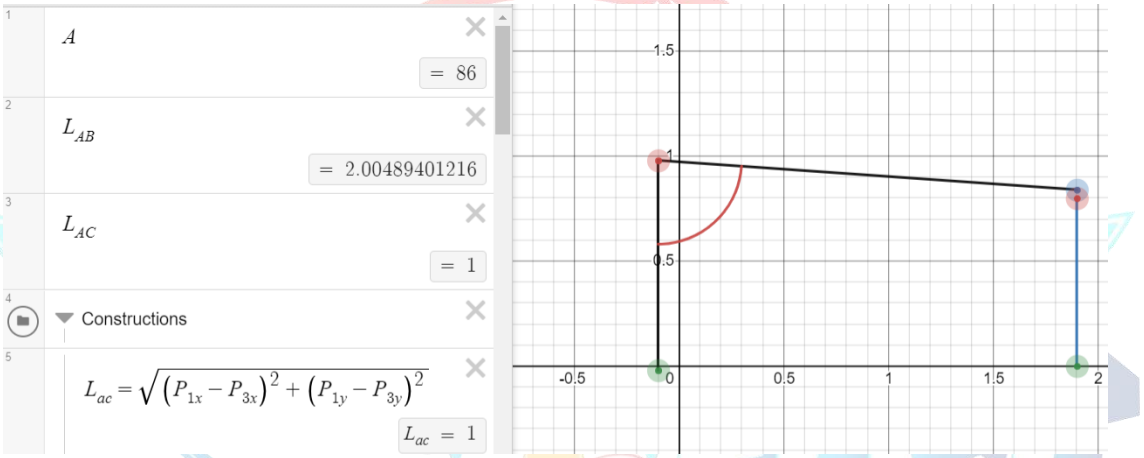
Görev sırasında aracın çevresini algılayabilmesi ve uygun kararlar alabilmesi için çeşitli sensörlere ihtiyaç duyulmaktadır. Sensörlere otonom sürüş ve yabani ot tanıma sistemlerinde ihtiyaç duyulmaktadır.

Yabani ot tanıma sistemi görüntü işleme prensipleri ile çalışmaktadır. Bu sebeple görüntü işleme algoritmasının koşacağı Jetson Nano cihazına bir adet 720P HD kamera takılmıştır. Yüksek FPS ihtiyacından dolayı düşük çözünürlüklü, yüksek renk doğruluğuna sahip bir kamera tercih edilmiştir. Gerektiğinde görüntü işleme esnasında çözünürlük düşürülerek FPS arttırılabilecektir.

Otonom sürüş için ise aracın çevresindeki nesneleri algılaması gerekmektedir. Bu sebeple Şekil 9'da da göründüğü gibi, üçü aracın önünde, kalan ikisi de aracın iki yanında olmak üzere toplam beş adet VL53L0X ToF (Time of Flight) sensör kullanılmıştır. Bu sensörün tespit edebileceği maksimum mesafe 2 metredir. Bu sensörlerin seçilme sebebi hızlı tepki veren, yüksek doğrulukla çalışan ve kolay işlenebilen sensörler olmalarıdır. Sensörler aracın tasarımı gereği 1 metre yükseklikte konumlandırılmışlardır. Aracın alçak araçlarının zarar görmemesi için 80 cm üzerindeki nesnelerin tespiti gereklidir. Bunun için gerekli hesaplamalar yapılarak sensörler Şekil 10'da görüldüğü gibi 86 derecelik açı ile yerleştirilmiştir. Böylece araç tam 1,9 metre mesafeden, 0,85 cm ve üzeri cisimleri tespit edebilmektedir.



Şekil 9 Sensör Konumlandırması



Şekil 10 Sensör Açı Gösterimi

Otonom sürüş için aracın konumunun doğru bir şekilde hesaplanması gereklidir. Bu sebeple aracın sağ ve sol ön tekerlerine birer adet encoder yerleştirilmiştir. Encoderler tekerin dönüş mesafesini hesaplayarak aracın ne kadar yol aldığını bulur ve bu sayede dönüşler doğru şekilde yapılır.

Aracın denge sisteminin çalışması için araçta bir adet MPU6050 6 eksen ivme sensörü bulunmaktadır. Böylece aracın dengesi doğru bir şekilde sağlanabilecektir.

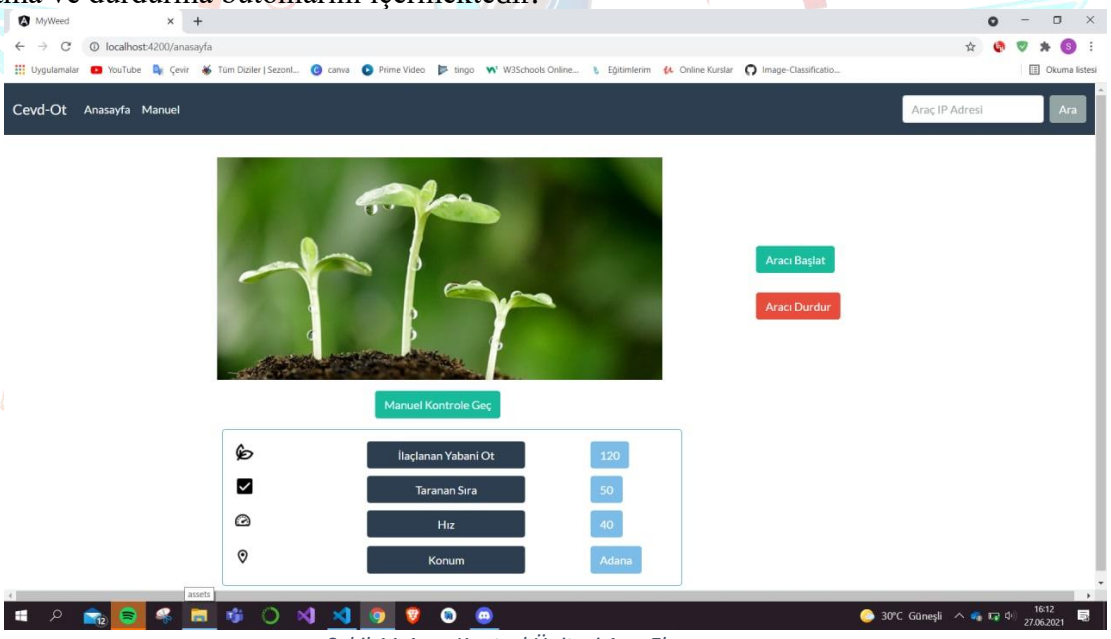
HAVACILIK, UZAY VE TEKNOLOJİ FESTİVALİ

5. Araç Kontrol Ünitesi

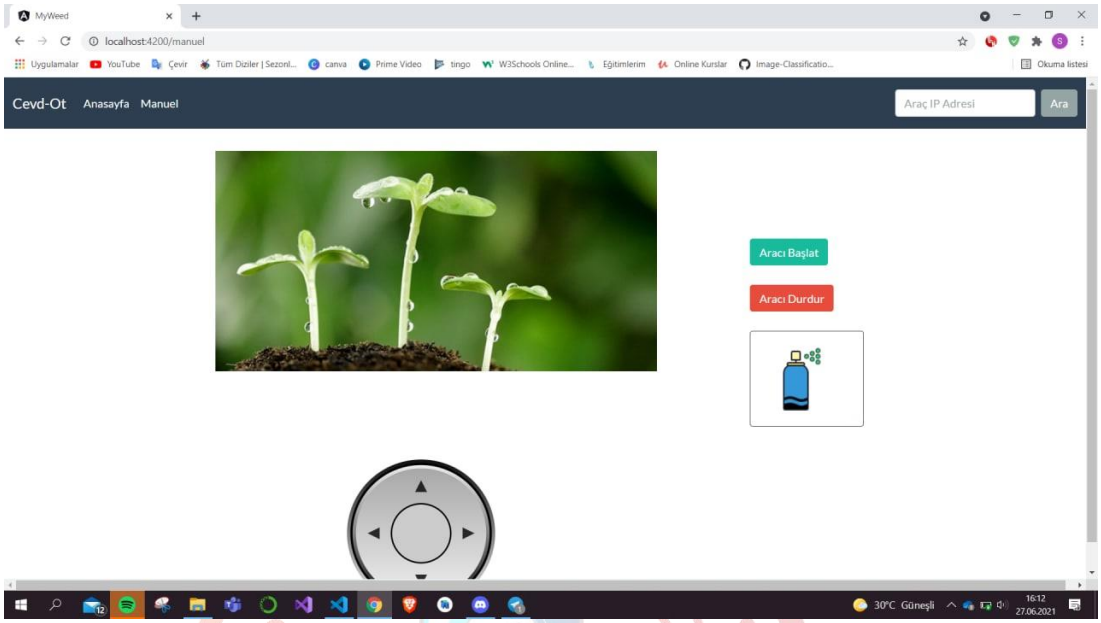
Tarımsal insansız kara aracının uzaktan kontrol edilebilmesi için web tabanlı bir arayüz oluşturulmuştur. Araç tasarımında Jetson Nano kullanılmış olup veriler bir 3G modem aracılığı ile minimum güç harcayarak oldukça hızlı bir şekilde Firabase'e aktarılmaktadır. 3G modem kullanılmasının sebebi tasarımda sağladığı kolaylık ve ortaya çıkabilecek sorunların azlığıdır. Aracın taradığı ekin sırası, tespit ettiği ve ilaçladığı yabani ot miktarı, aracın hızı konumu ve pil gücü, Araçta bulunan sensör füzyonundan elde edilip arayüzde gösterilmektedir.

Jetson Nano'da çalışan IPCamera sunucusu sayesinde elde edilen görüntüler Angular ile tasarlanan arayüzde anlık olarak gösterilmektedir. Web arayüzü yerel bir NodeJS sunucuda koşmaktadır. Bu sayede arayüz kullanıcısı aracın önündeki engelleri, ekinleri ve oluşabilecek acil durumları anlık olarak takip edebilecektir.

Acil durumları önlemek ve araç güvenilirliğini artırmak için manuel olarak aracı durdurma ve başlatma butonları bulunmaktadır. Araç istenildiği an durdurulabilecek ve acil durum giderildiğinde kaldığı yerden tekrardan başlatılabilecektir. Ayrıca oluşturulan arayüz manuel kumanda da içermektedir. Kullanıcı istediği an manuel kontrole geç butonu sayesinde aracı dijital joystick sayesinde kendisi kontrol edebilecektir. Manuel kumanda sayfası da aracı başlatma ve durdurma butonlarını içermektedir.



Şekil 11 Araç Kontrol Ünitesi Ana Ekran

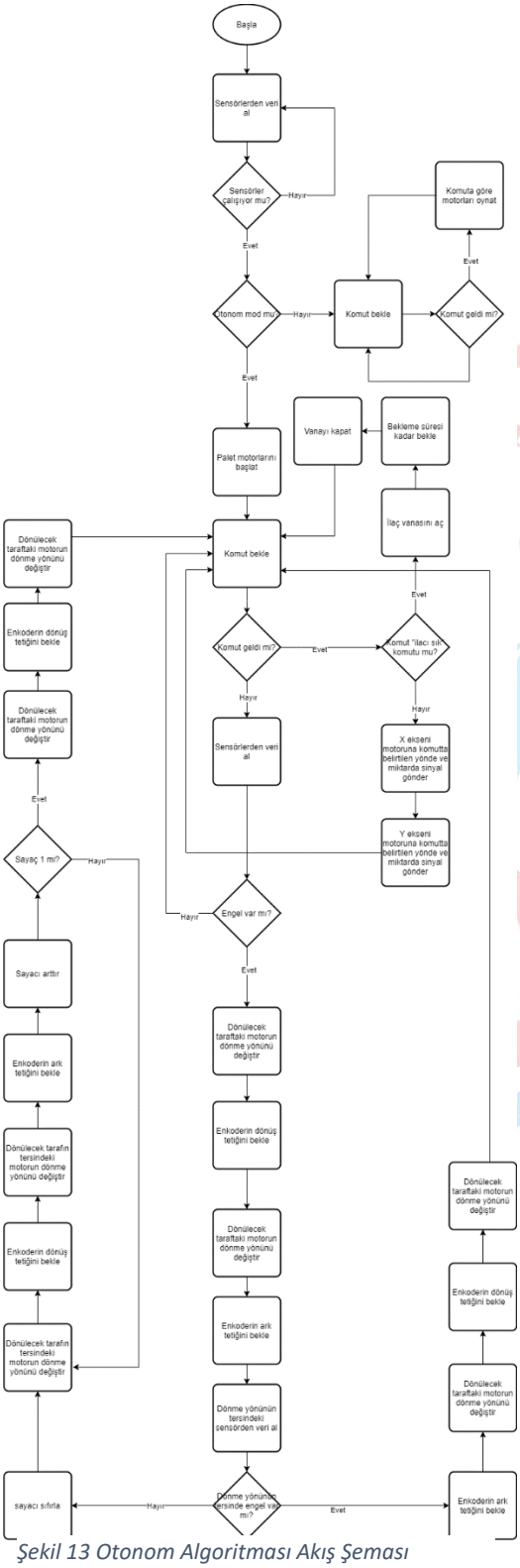


Şekil 12 Araç Kontrol Ünitesi Uzaktan Kontrol Arayüzü

TEKNOFEST
HAVACILIK, UZAY VE TEKNOLOJİ FESTİVALİ

6. Otonom Sürüş Algoritmaları

6.1. Otonom Yol Takibi



Şekil 13 Otonom Algoritması Akış Şeması

Aracın otonom olarak tüm alanı ekinleri tarayarak geçme görevi geleneksel algoritmalarla uyumsuz kalmıştır. Genelde Roomba (ev temizlik robotu) için kullanılan depth-first search türevi algoritmalar bu yarışmanın koşulları için yetersizdir. Bunun için bir çeşit labirent çözme algoritması uygulanmıştır. Ekin arkları labirentin koridorları gibi düşünülüp bir sonraki arka geçiş ancak mevcut arkın sonundaymış gibi düşünüldü. Aracın etrafındaki üç sensör demetinin birinci amacı koridor boyunca aracın karşısına bir engel çıktığında bunu tespit edip çevresinden dolaşmasını sağlamak. İkinci amacı ise alan sınırını tespit edip taranacak bir sonraki Ekin arkına (koridora) geçmek. Önce ön sensör engeli tespit eder, araç kendi etrafında 90° döner ve 1 ark mesafesi kadar ilerler. Burada iki olasılık var:

a) sol veya sağ sensörler bir engel tespit ederse, engel enine devam ediyor demektir. Yani çittir. Bu durumda 1 ark mesafesi kadar daha ilerlenir ve aynı yöne doğru araç kendi etrafında 90° daha döner ve tekrar ilerlemeye devam eder.

b) sağ veya sol sensörler bir şey tespit etmezse engel çevresinden dolaşılabilir demektir. Araç önce kendi etrafında bir öncekine göre ters yönde 90° döner, 1 ark mesafesi kadar ilerler, tekrar 90° ters döner, 1 ark mesafesi kadar ilerler ve ilk dönüşündeki yönde 90° kendi etrafında döndükten sonra ark üzerinde yoluna devam eder.

Engelin solundan mı sağından mı dolaşılacağı aracın alanın neresinden başladığına bağlıdır. Eğer alanın üstünden başlanırsa; ilk arkta bir engel çıkma olasılığına karşı, alan sınırı aracın solunda kalacağı hesaba katılarak çevresinden dolaşılırken alan sınırından çıkmamak için engelin sağından dolaşılacaktır. Sensörlerin ikinci amacı da ark sonunu, yani alanın sonunu belirten çiti engelden ayırt edip buna göre hareket etmesini sağlıyor.

Algoritmanın çalışma prensibi Şekil 13'te bulunan akış şemasında detaylı bir biçimde açıklanmıştır.

6.2. Yabani Ot Tespiti

Aracın yabani ot tanıma algoritması, kameradan gelen görüntüleri işleyerek görev için gerekli algoritmaları uygulamaktan sorumludur. Yabani otu tespit etmek için çeşitli algoritmalar mevcuttur. Bunlardan bazıları; R-CNN, FASTER R-CNN, MASK R-CNN, SSD, YOLOV3, YOLOV4, YOLOV5… Bu algoritmalar karşılaştırılmış ve en iyi sonucu elde etmek için gereken algoritmaya karar verilmiştir.

R-CNN, gelen görüntüyü 2000 bölge tavsiyesine böler, ardından her bölge için sırasıyla CNN (ConvNet) uygulanır. Bölgelerin boyutu belirlenir ve yapay sinir ağına doğru bölge yerleştirilir. Bu yöntem zaman açısından oldukça maliyetlidir. Resimdeki her bir bölge ayrı olarak CNN uyguladığından, eğitim süresi yaklaşık 84 saat ve tahmin süresi yaklaşık 47 saniyedir.

Fast R-CNN, R-CNN algoritmasının daha hızlandırılmış halidir. R-CNN oldukça yavaş bir modeldir ve gerçek zamanlı olarak nesne tanımak pek mümkün değildir. Fast R-CNN, R-CNN'in aksine resimlere bölge önerisi yapmak yerine görüntüleri doğrudan konvansiyonel sinir ağından geçirir ve orijinal resme uyan bir özellik haritası çıkartır. Fast R-CNN ile sınıflandırma yapmak için softmax layer kullanılmaktadır. Sınıflandırma yapabilmesi için yeni bir model oluşturmak yerine sinir ağını genişletip sinir ağı içerisinde sınıflandırma yapılmaktadır. Fakat Fast R-CNN de bir zayıf yönü vardır. Zamanın çoğunu test aşamasında bölge önerisi yapmak için harcamaktadır. Bölge önerisi ağının iki tane görevi vardır; her bir öneri için nesne var mı yok mu karar verilmesi ve önerilerin pencere büyüklüğünün belirlenmesi. Daha sonra asıl sinir ağı sınıflandırma işlemi gerçekleştirilerek bakılan bölge içerisinde nesne var mı yok mu tespit edilecektir. Bu algoritma denenmiş olup oldukça hızlı olmasına rağmen başarısız sonuçlar elde edilmiştir.

HAVACILIK, UZAY VE TEKNOLOJİ FESTİVALİ

Şekil 14 Fast RCNN

SSD (Single Shot MultiBox Detector), faster R-CNN'den daha hızlı bir modeldir. RCNN ile bölge önerisini iki farklı aşamada yapılmakta iken ilk önce resimde nesne olması beklenen bölgeler belirlenmektedir. Daha sonra tam bağlı katman ile sınıflandırma yapılmaktadır. SSD bu ikisini tek seferde gerçekleştirmektedir. Diğer modellerde olduğu gibi her zamanki gibi ilk öncelikle girdi olarak bir resim alınmaktadır.

Resim konvansiyonel sinir ağından geçirilmektedir. Böylece farklı boyutlarda özellik haritaları çıkmaktadır. Tüm özellik haritalarında 3x3 konvansiyonel filtre yardımıyla az miktarda sınırlayıcı dikdörtgenler elde edilmektedir. Her dikdörtgen için aynı zamanda hem sınırlar hem sınıflandırmalar belirlenmektedir. Bu dikdörtgenler her aktivasyon haritasında olduğu için hem küçük nesneleri hem büyük nesneleri tanıyabilmektedir. SSD algoritması da denenmiştir ve onunla da çok başarılı sonuçlara ulaşılamamıştır.

TEKNOFEST
HAVACILIK, UZAY VE TEKNOLOJİ FESTİVALİ

Şekil 15 SSD (Single Shot MultiBox Detector)

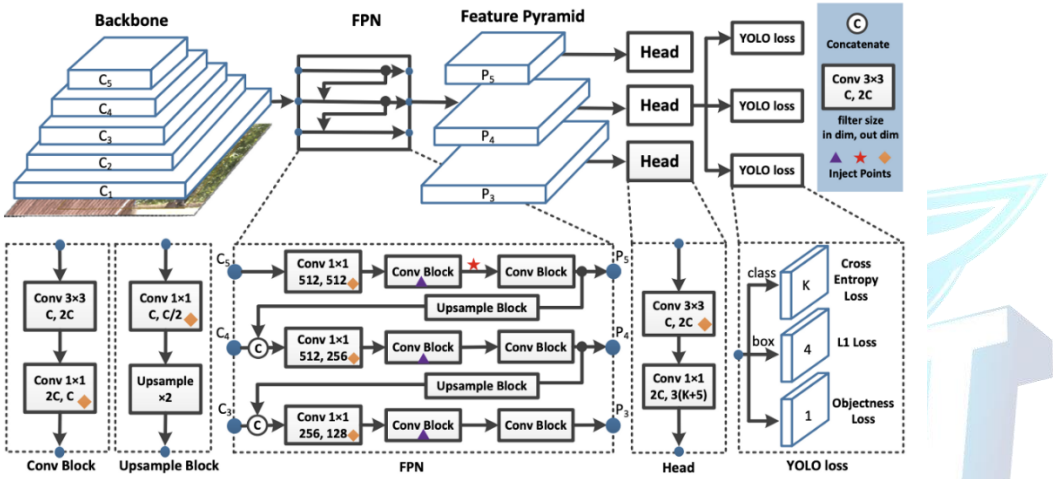
YOLO algoritması ile konvolüsyonel sinir ağları (CNN) kullanılarak yabani bitkiler tespit edilecektir. Algoritma nesne takibini tek seferde yapabilecek kadar hızlıdır çünkü algoritma resmin tamamını tek seferde nöral bir ağdan geçirme üzerine kurulmuştur ve bu bize performans olarak geri dönecektir. Bu nöral ağın (CNN) [7] ilk katmanı "convolutional layer" resmin özelliklerini algılamaktan sorumludur. Bu katman, görüntüdeki düşük ve yüksek seviyeli özellikleri çıkarmak için resme bazı filtreler uygular ve filtreyi görüntü üzerinden hareket ettirerek ve basit matris çarpımını kullanarak, özellikleri tespit eder. İlk filtre uygulandığında, bir Feature Map oluşturulur ve bir özellik türünü tespit eder. Ardından, ikinci bir filtre kullanıp başka bir özellik türünü algılayan ikinci bir Feature Map oluşturulur.

Tüm Convolutional katmanlarından sonra Non-Linearity (doğrusal olmayan) katmanı gelir. Görüntüdeki doğrusallık bir problemdir çünkü tüm katmanlar doğrusal bir fonksiyon olabildiğinden dolayı Sinir Ağı tek bir perception gibi davranır, yani sonuç, çıktıların linear kombinasyonu olarak hesaplanabilir. Bu katman aktivasyon katmanı (Activation Layer) olarak adlandırılır çünkü aktivasyon fonksiyonlarından [9] birini kullanılır. Sinir Ağı eğitiminin hızı konusunda en iyi sonucu Rectifier (ReLu) fonksiyonu verdiği için artık bu fonksiyon kullanılmaya başlanmıştır.

Konvolüsyonel katmanların ardından Toplu Normalleştirme (Batch Normalization) kullanılarak nöronların stabilizesi arttırılacaktır. [8] Bu katmandan sonra pooling katmanı gelir. Bu katmanın görevi, gösterimin kayma boyutunu ve ağ içindeki parametreleri ve hesaplama sayısını azaltmak içindir. Bu sayede ağdaki uyumsuzluk kontrol edilmiş olur. Daha sonra flatting katmanı gelir. Bu katmanın görevi basitçe, son ve en önemli katman olan Fully Connected Layer'ın girişindeki verileri hazırlamaktır. Sinir ağları, giriş verilerini tek boyutlu

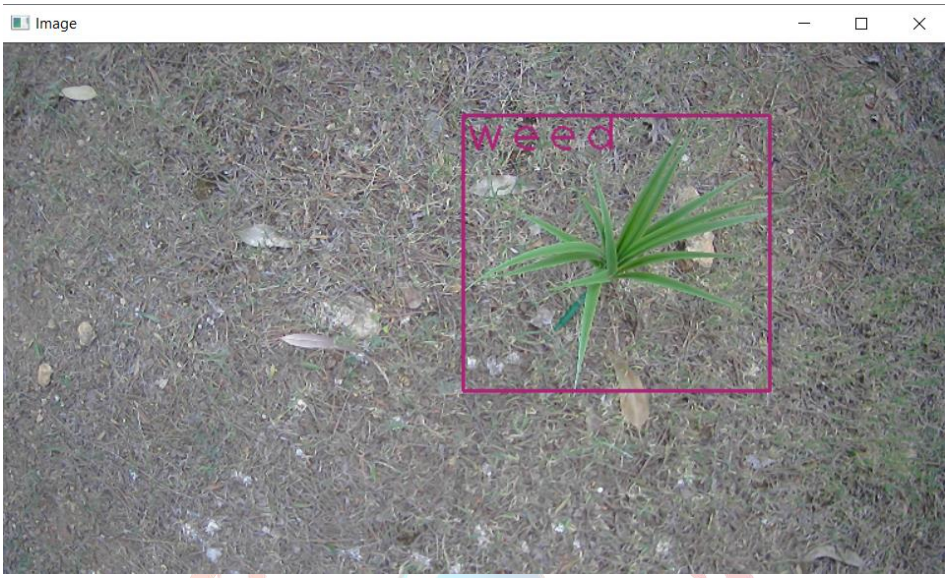
bir diziden alır. Bu sinir ağındaki veriler ise Convolutional ve Pooling katmanından gelen matrixlerin tek boyutlu diziye çevrilmiş halidir. Son olarak fully-connected layer katmanı gelir. Bu katman ConvNet'in son ve en önemli katmanıdır. Verileri Flattening işleminden alır ve Sinir ağı yoluyla öğrenme işlemini geçekleştirir.

Algoritma çalışmaya başladığında görüntülerdeki yabani bitkiler ve bitkilerin konum aynı anda tespit edilecektir. YOLO algoritması, öncelikle görüntüyü bölgelere ayıracak. Daha sonra her bir bölgedeki nesneleri çevreleyen kutuları (bounding box) çizecek ve her bir bölgede yabani bitki bulunma olasılığını hesaplayacaktır. Her bir bounding box için bir güven skoru hesaplayacak ve skor o nesnenin yüzde kaç olasılıkla tahmin edilen nesne olduğunu söyleyecektir.

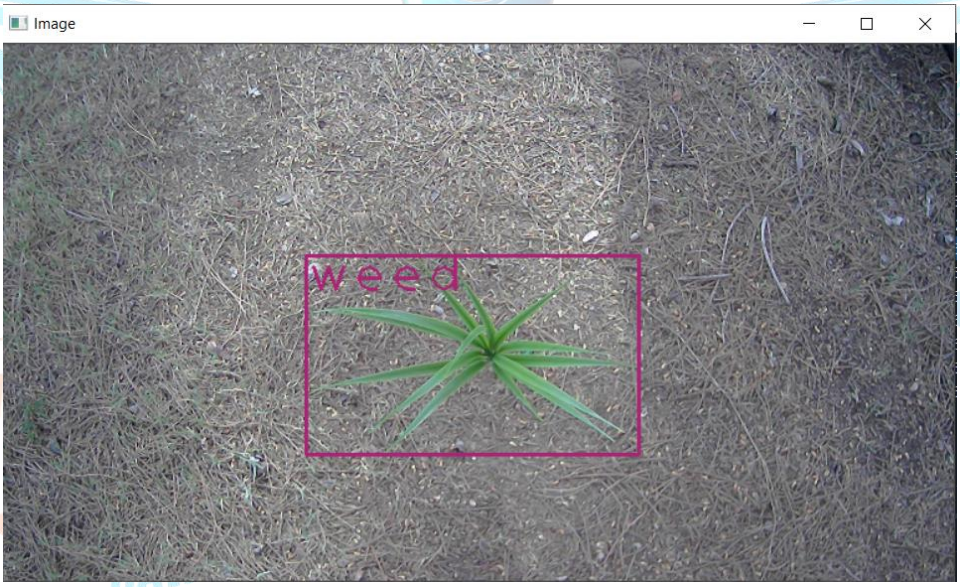


Şekil 16 Yolo Mimarisi

Bütün nesne tespiti algoritmaları karşılaştırılmış olup Tarımsal İnsansız Kara Aracı projesindeki yabani ot tanıma algoritması için YOLOV4 kullanılmaya karar verilmiştir. İlk önce yabani otun farklı ışık ortamlarında, farklı açılardan fotoğrafları çekilerek 500 verili bir veri seti oluşturulmuştur. Veri seti YOLOV4 algoritmasında kullanılmak üzere tek tek etiketlendirilmiştir. Etiketlenen veriler YOLOV4 algoritmasında eğitilmiş ve her 1000 iteration'da 1 tane olmak üzere 3 farklı ağırlık değeri elde edilmiştir. Son ağırlık değeri de elde edildiğinde test süreci başlamıştır. Testler oldukça başarılı olmakla birlikte yabani otun konumu tam olarak tespit edilebilmektedir.



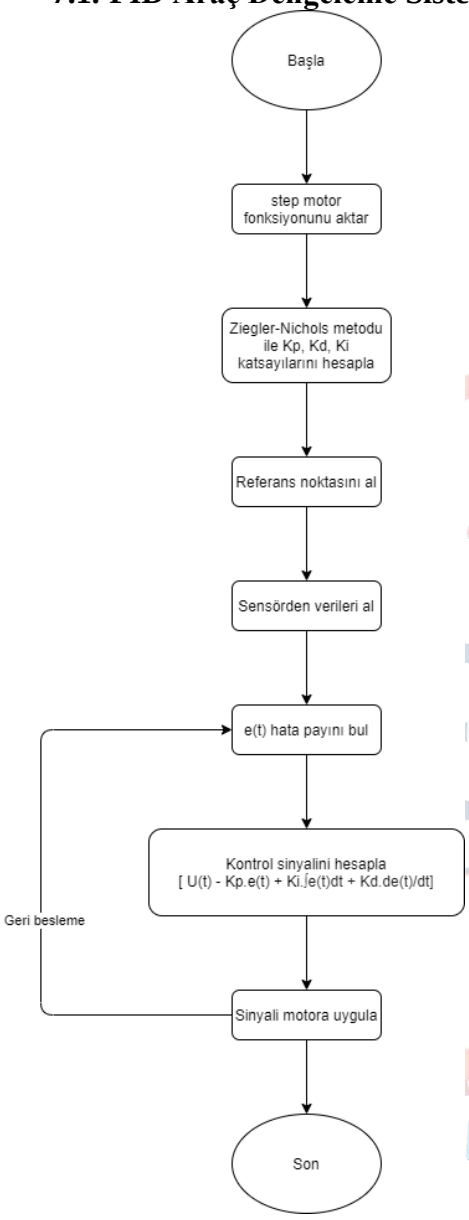
Şekil 17 Yolo Test Sonucu



Şekil 18 Yolo Test Sonucu

7. Özgün Bileşenler

7.1. PID Araç Dengeleme Sistemi



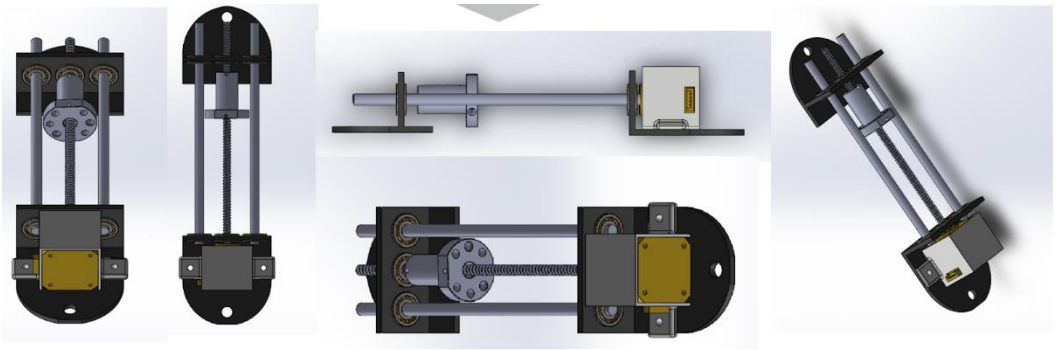
Şekil 19 PID Algoritması Akış Diyagramı

Aracın görüntü işleme başarısı çevresel faktörler ile doğrudan ilişkilidir. Bu çevresel faktörlerden birisi de titreşim ve sarsıntılardır. Engebeli arazilerde aracın ekseninin sabit tutulması zordur. Aracın gövdesi, altındaki toprağın eğimine göre eğildiği için sensörlerin ve kameranın hatalı veri üretmesine sebep olmaktadır. Bunun önüne geçmek için araç, gövdesini sabit tutacak bir sistem geliştirilmiştir. Sistemin çalışması Şekil 19'da anlatılmıştır. Kameradan alınan görüntüye göre engebenin eğimi ve bu eğimin yönü hesaplanıp Arduino Nano'ya gönderilecektir. Arduino Nano burada PID [2] fonksiyonu sayesinde gövdeyi sabit tutmak için ne tarafa eğmesi gerektiğini hesaplayıp buna göre kendisine bağlı olan dört step motoru çalıştıracaktır. PID aslında geri bildirimli bir kontrol mekanizmasıdır. PID aslında fonksiyonda kullandığımız terimlerin baş harflerinin birleşimidir. Açılımı "Oransal-Integral-Türev" (Proportional-Integral-Derivative) demek. Bu mekanizmada girdiğimiz bir fonksiyonun, belli bir zaman aralığında istediğimiz değerini ve o anki değerini alırız. Bu değerlerin arasındaki fark fonksiyonun istediğimiz değerden uzaklığını temsil eder. Buna hata payı (e(t)) denmektedir [1]. Hesaplama Denklem 1'deki gibi yapılmaktadır.

$$[\,U(t) - Kp.e(t) + Ki.\int e(t)dt + Kd.de(t)/dt]$$

Denklem 1 PID Fonksiyonu

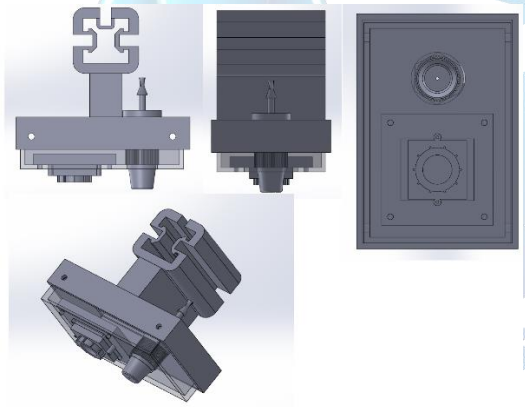
Tasarlanan sistemin çalışabilmesi için elektronik olarak üst tabla açısının değiştirilebileceği bir mekanik sistem geliştirildi. Bu sistem, step motor ile sürülecek şekilde tasarlanan lineer aktüatörler ile çalışmaktadır. Dört adet kullanılan lineer aktüatörler, dört adet bacak ile üst tablanın bağlantısını yapacak şekilde yerleştirilmiştir. Lineer aktüatörlerin tasarımı Şekil 20'de gösterilmiştir.



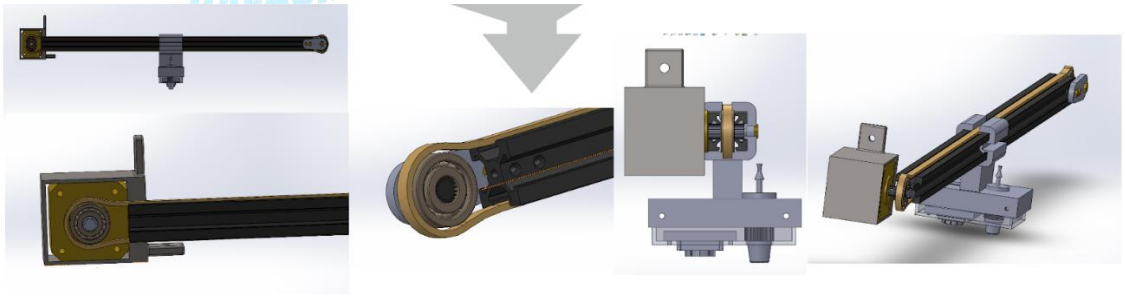
Şekil 20 Lineer Aktüatör Tasarımı

7.2. Raylı İlaç Püskürtme Sistemi

Aracın ilaç püskürtme sistemi tek eksende hareket eden bir meme ucu ve Şekil 21'de göründüğü gibi, ona bağlı kamera sistemi ile tasarlanmıştır. Araç ilerlerken, kamera ilerde bir nesne tespit ettiği takdirde raylı sistem püskürtme ucunu yabani ot ile aynı hizaya getirmeye çalışır. Aynı hizaya geldiği takdirde ilaç püskürtülür ve ilaçlama sağlanır. Bu sistem için araçta kayışlı bir ray sistemi tasarlanmıştır. Sistem bir step motor ile çalışmaktadır. Bu sayede hassas bir şekilde istenen konuma ilaç püskürtülebilmektedir. Sistemin nihai tasarımı Şekil 22'de gösterilmiştir.



Şekil 21 Kamera, Meme Ucu Paketi



Şekil 22 Raylı İlaç Püskürtme Sistemi

8. Güvenlik Önlemleri

Tarımsal insansız kara aracında yararlı bitkilere, hayvanlara, insanlara ve alandaki araçlara gelebilecek zararlara karşı önlemler alınmıştır:

Tasarlanan İKA, web tabanlı bir arayüz üzerinden kontrol edilebilmektedir. Arayüz, aracı başlatma ve durdurma butonlarının yanında, sistemin gücünü kesecek bir acil durum butonunu da içermektedir. Ayrıca aracın üzerinde bulunan devre kutusunun erişilebilir yüzünde bir adet acil stop butonu bulunmaktadır. Bu sayede araç, acil durum anlarında hemen durdurulabilecektir ve olası zararlar engellenebilecektir. Ayrıca acil durum giderildiğinde araç aynı yerden başlatılabilecektir.

Tasarlanan aracın iskeletinde alüminyum sigma profiller kullanılacaktır. Sigma profiller oldukça sağlam olmasının yanında hafif malzemelerdir. Bu da aracın güvenirliğini artırmaktadır. Sigma profillerin bağlantı noktalarında kullanılacak olan radiuslu köşe bağlayıcıları estetik görünümünün yanında keskin kenarları yok ederek yaralanmaları önler ve sağlam bir birleşim sağlar.

Aracın yararlı ekinlere zarar vermemesi, yalnızca yabani bitkileri yok etmesi için oldukça başarılı bir model eğitilmiştir. Model eğitilirken YOLOV4 algoritması kullanılmış olup başarı oranı %98'dir. Bu sayede ekinlere verilen zarar minimuma indirilmiştir.

%100 başarılı bir yapay zekâ modeli elde etmek çok zordur bu yüzden yabani bitkileri yok etme metodu olarak ilaçlama yöntemi tercih edilmiştir. Eğer ilaçlama yöntemi değil de yerinden sökme metodu tercih edilseydi yapay zekâ modelinin %2'lik hata ihtimali ekinlere geri döndürülemez bir zarar verebilecekti. Tercih edilen ilaca bağlı olmakla birlikte; bu yöntemde yararlı bitkilere zarar verme ihtimali düşürülmüştür.

Aracın dengesi PID fonksiyonları ile sürülen lineer aktüatör sistemi ile sağlanmaktadır.

Aracın elektronik aksamları sıcaklık, darbe gibi olası dış etkenlerden korumak adına aracın alt kısmına yerleştirilmiş bir kutu içerisinde hassas bir şekilde konumlandırılmış ve aracın alt kısmında bulunan ekstra destek kolonuna bağlanılmıştır. Böylece devreler araca sağlam bir şekilde yerleştirilmiştir.

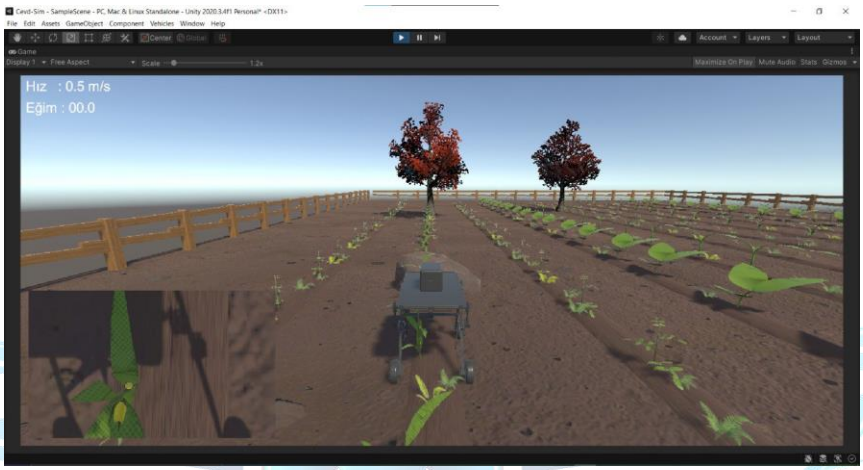
Araçta kullanılan tüm motorlara darbelere ve sarsılmalara dayanıklı bir şekilde korunaklı yuvalar yapılmıştır.

Aracın ilaç deposu üst kısma yerleştirilmiş, bu sayede sıvıların elektronik aksama sıçrama yapması önlenmiştir.

Aracın otonom algoritma test süreçleri hazırlanan simülasyon yazılımı ile yapılacaktır. Test sonucunda algoritmada oluşabilecek hatalar belirlenip hatalar soncu oluşabilecek hasarlar önlenecektir.

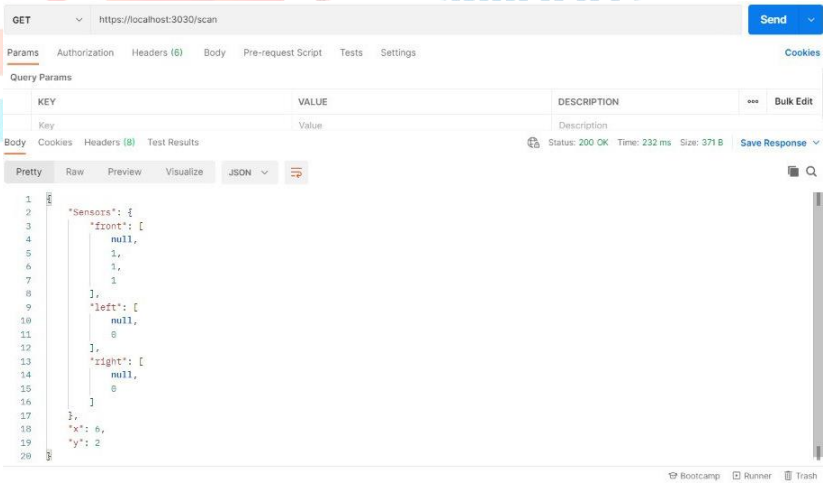
9. Simülasyon ve Test

Araç için geliştirilen otonom algoritmasının denenmesi için Unity üzerinde bir simülasyon ortamı oluşturulmuştur. Oluşturulan simülasyon ortamında ölçüler, gerçek araç ve alan ölçülerine göre belirlenmiştir. Araç, Unity'ye dahili olarak gelen Vehicles modülü üzerinde oluşturulmuştur. Aracın ağırlık, tork ve malzeme sertlikleri gerçeğe yakın şekilde belirlenmiş ve bu sayede fizik simülasyonu gerçekçi olmuştur. Şekil 23'te gözüktüğü gibi simülasyon ortamının arayüzü kullanıcıya hız ve eğim verilerini göstermektedir. Ayrıca araç kamerasının görüntüsü sol altta gösterilmektedir.



Şekil 23 Simülasyon Ortamı

Simülasyon ortamı yazılan algoritma kodu ile bir http rest API ile haberleşmektedir. Python üzerinde modellenen otonom algoritması simülasyon ortamına Şekil 24'te görüldüğü gibi gönderdiği GET Request ile sensör verilerini ve araç konumunu almaktadır. Daha sonra bu veriler ışığında aracın hızını ve yönünü belirler. Araç için tasarlanan otonom algoritması simülasyon üzerinde farklı engel koşullarına ve başlangıç konumlarına göre denenmiş olup algoritmada bir hata olmadığı onaylanmıştır.



Şekil 24 Rest API

Aracın yabani ot tespit etme algoritması araçta kullanılacak olan kameradan aracın tasarlanan yüksekliğinden gerçek zamanlı olarak test edilmiştir. Yapılan testler farklı ışık ortamlarında, bitkinin sağı, solu, üstü gibi farklı açılarından, farklı hızlarla gerçekleştirilmiştir. Testler sonucunda YOLO V4 ile yabani ot tespit etme modelimiz %89,5 başarı göstermiştir. Başarıyı etkileyen en önemli unsurun kameranın (aracın) hızı olduğuna karar verilmiştir ve elde edilebilecek maksimum başarı için gerekli hız belirlenmiştir. Model başarısını etkileyen bir diğer husus ise veri setinin çeşitliliğidir. Başlangıçta az fotoğraf ile eğitilen ve başarısı düşük olan model daha sonra veri setinin farklı ışık ortamlarında ve farklı açılarda çekilen fotoğraflarla çeşitlendirilmesi sonucunda oldukça yüksek bir başarıya ulaşmıştır. Yapay zekâ modelini veri setini daha da çeşitlendirmek, denenmeyen nesne tespiti algoritmalarını denemek, farklı arka planlarda veri setleri oluşturmak vb. suretiyle başarısının %100'e yaklaştırılması hedeflenmiştir. Eğitilen Modellerin bilgisayar üzerinde örnek bir video ile yapılan test sonuçları Tablo 1'de görülmektedir.

ALGORİTMA KARŞILAŞTIRMASI

	Başarı (%)	FPS
Faster R-CNN	75.4	5-17
SSD	72.4	22-59
YOLO V2	69	40
YOLO V4	89.5	91
Tensorflow	76	60

Tablo 1 Algoritma Karşılaştırması

10. Referanslar

[1] Radke, F., & Isermann, R. (1987). A parameter-adaptive PID-controller with stepwise parameter optimization. Automatica, 23(4), 449–457. doi:10.1016/0005-1098(87)90074-4
[2] Aslam, Sohaib & Hannan, Sundas & Sajjad, Muhammad & Zafar, Waheed. (2016). Implementation of pid on pic24f series microcontroller for speed control of a dc motor using mplab and proteus. Advances in Science and Technology Research Journal. 10. 40-50. 10.12913/22998624/64060.
[3] Watts, S. E. (1998). Short-Term Influence of Tank Tracks on Vegetation and Microphytic Crusts in Shrubsteppe Habitat. Environmental Management, 22(4), 611–616. doi:10.1007/s002679900132
[4] Adarsh, P., Rathi, P., & Kumar, M. (2020). YOLO v3-Tiny: Object Detection and Recognition using one stage improved model. 2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS). doi:10.1109/icaccs48705.2020.9074315
[5] Sammons, Phil & Furukawa, Tomonari & Bulgin, Andrew. (2005). Autonomous Pesticide Spraying Robot for use in a Greenhouse.
[6] BUDYNAS, Richard G.; NISBETT, J. Keith; TANGCHAICHIT, Kiatfa. Shigley's mechanical engineering design. New York: McGraw Hill, 2005.
[7] H. Shin et al., "Deep Convolutional Neural Networks for Computer-Aided Detection: CNN Architectures, Dataset Characteristics and Transfer Learning," in IEEE Transactions on Medical Imaging, vol. 35, no. 5, pp. 1285-1298, May 2016, doi: 10.1109/TMI.2016.2528162.
[8] R. Chauhan, K. K. Ghanshala and R. C. Joshi, "Convolutional Neural Network (CNN) for Image Detection and Recognition," 2018 First International Conference on Secure Cyber Computing and Communication (ICSCCC), 2018, pp. 278-282, doi: 10.1109/ICSCCC.2018.8703316.

TEKNOFEST
HAVACILIK, UZAY VE TEKNOLOJİ FESTİVALİ