TEKNOFEST

HAVACILIK, UZAY VE TEKNOLOJİ FESTİVALİ

TARIMSAL İNSANSIZ KARA ARACI YARIŞMASI KRİTİK TASARIM RAPORU

PROJE ADI : Leto
TAKIM ADI : Spektrum TİKA
TAKIM ID : 463321

Akademik Danışman : Arş. Gör. Kadir Tuğrul BOZKURT
Takım Seviyesi : Üniversite-Mezun
Takım Lideri : Ali YILDIRIM

TAKIM ÜYELERİ

Halil İbrahim SABUR
Ali Sefa AKBULUT
Durdu Onur Can YERLİ
Yakup Abdullah DOĞAN
Onur SEZGİN
Nesibe TURAN
Obeid ABDALLA

SPEKTRUM

İÇİNDEKİLER

1. Takım Organizasyonu ---------- 3

2. Ön Tasarım Raporu Değerlendirmesi ---------- 3

3. Araç Özellikleri ---------- 5

3.1.Mekanik Özellikler ----------5

3.2. Elektronik Özellikler ----------7

3.3.Yazılımsal Özellikler ----------8

4. Sensörler ---------- 9

4.1. Kullanılan Sensörler ----------9

4.2. Sensör Konumları ----------11

5. Araç Kontrol Ünitesi ---------- 11

5.1. Haberleşme----------12

5.2. Yazılım----------13

6. Otonom Sürüş Algoritmaları ---------- 13

6.1. Yabani Ot Algılama ----------13

6.2. Yol Takibi Algoritması----------18

7. Yabancı Otla Zirai Mücadele Yöntemleri ---------- 19

7.1. Yazılımsal----------19

7.2. Donanımsal----------19

8. Özgün Bileşenler ---------- 20

9. Güvenlik Önlemleri ---------- 20

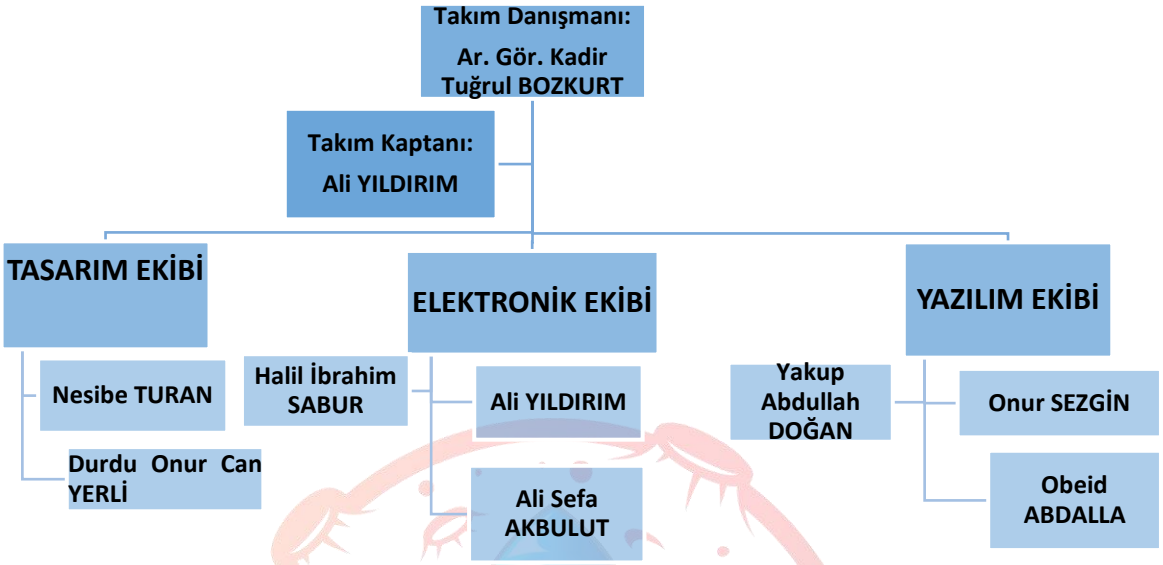
10. Simülasyon ve Test ---------- 21

10.1. Simülasyon----------21

10.2. Test----------24

11. Referanslar ----------25

1. Takım Organizasyonu



Spektrum Tika Takımı, başta döner kanatlı İHA olmak üzere otonom sistemler üzerine çalışmalarını sürdürmektedir. Takımımız 1 yüksek lisans ve 7 lisans öğrencisinden oluşmaktadır. Ağırlıklı olarak elektrik elektronik mühendisliği öğrencilerinden kurulu takımımızda makine ve enerji sistemleri mühendisliğinden de öğrenci bulunduran disiplinler arası bir takımdır. Takımımız Otonom kontrol sistemleri hakkında deneyimli olması sebebi ile tarımsal insansız kara aracında da kestirimlerde bulunabilmektedir. Takımımızda tasarım kısmı ile ilgilenen tasarımcı arkadaşlarımız SolidWorks Programı üzerinden etkin şekilde çizim ve analiz yapma, keskin Mekanik tespitlerde bulunabilme özelliklerine sahiptirler. Elektronik ekibimiz ise Altium, Proteus, elektronik bileşenler, elektronik bileşenlerin yazılım ve mekanik kısımlarına uygunlukları konusunda deneyimlidir. Yazılım ekibimiz ise Python, C++ yazılım dillerine hakimdir. Python ile nesne tabanlı programlama ve görüntü işleme konusunda süregelen tecrübeleri bulunmaktadır.

2. Ön Tasarım Raporu Değerlendirmesi

Ön tasarım raporu, yarışma hakemleri tarafından geri bildirimler ile tekrar takım tarafından incelenmiş ve gerekli düzenlemeler, eksikler, iyileştirmeler kritik tasarım raporunda daha açıklayıcı olarak üzerinde durulmuştur. Tarımsal insansız kara aracında kullanılan mekanik ve elektronik bileşenler ön tasarım raporundaki genel anlatım sebebiyle anlaşılır kalmamasından dolayı kritik tasarım raporunda daha net ve anlaşılır olunmasına dikkat edilmiştir. Aracımızda önemli bir değişiklik ilaçlama sisteminde olmuştur. Ön tasarım yerinde belirtilen ve aracın üst kısmında yer alan tamamen hareketli ilaçlama kolu sadece ilk hareket, yapıp kamera konumuna yönelen sistemine geçilmişi uygun görülmüştür. Detaylar ilgili bölümde anlatılacaktır.

No	Parça Adı	Miktarı	Fiyatı(TL)	Durumu
1	Pixhawk Cube 2.1	1	6646,31	Alım Yapılacak
2	Pixhawk güç modülü	1	219,05	Alım Yapılacak
3	Jetson Nano	1	3850,30	Alım Yapılacak
4	Jetson Nano Soğutucu	1	346,70	Alım Yapılacak
5	Pi Kamera v1	2	1550,88	Alım Yapılacak
6	Redüktörlü Motor	4	924,24	Alım yapıldı
7	Flysky FS-I6S Kumanda Alıcı verici	1	1559,37	Alım yapıldı
8	Motor Sürücüsü	4	159,08	Alım yapıldı
9	Acces Point	1	144,80	Alım yapıldı
10	GPS Modülü	1	92,2	Alım yapıldı
11	Li-Po Batarya 4S 6200mAh	1	1874,97	Alım yapıldı
12	Voltaj Regülatörü	2	53,24	Alım yapıldı
13	Röle	1	12,45	Alım yapıldı
14	Servo Motor	1	100,89	Alım yapıldı
15	Emergency Button	1	45,17	Alım Yapılacak
16	XBee Pro 2.4 GHz 60 mW RF kablosuz haberleşme modülü	1	1011,64	Alım yapıldı
17	Lidar Sensörü	1	909,13	Alım yapıldı
18	Sigorta	1	13,50	Alım Yapılacak
19	12V su Pompası	1	99,42	Alım Yapılacak
20	İkaz Lambası	1	265,42	Alım Yapılacak
21	IMU sensör	1	41,30	Alım Yapılacak

Tablo 2.1: Malzeme Listesi

Ön Tasarım Raporu(ÖTR)	Kritik Tasarım Raporu(KTR)	Açıklama
Tasarım sürecinde aracın tepesinde hareketli bir ilaç kolu düşünülmüştür.	Tamamı Hareketli ilaçlama kolu yerine sadece ilk başta kamera konumuna yönelme yapan bir kol eklenmiştir	Hareketli bir ilaçlama kolu daha fazla malzeme ve uğraş gerektireceğinden bu şekilde uygun görülmüştür.
IMU sensörü düşünülmemiştir	IMU sensörü eklemiştir.	Yol eğimi, çukur, engebe gibi arazi şartlarını görmek için kullanılmıştır
19.212,78 TL	**19.919,95 TL**	

Tablo 2.2: Ötr-Ktr Karşılaştırılması

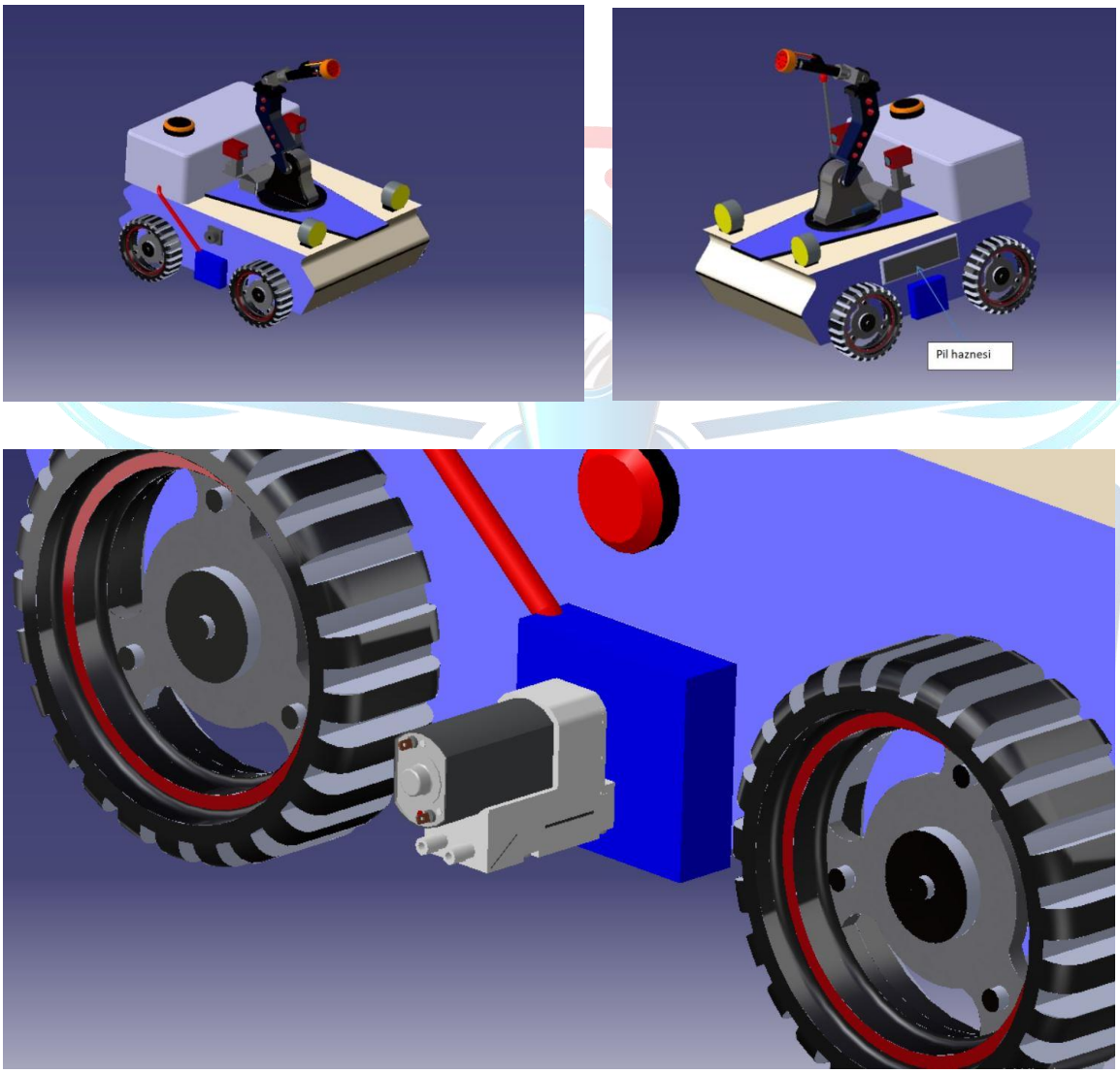
3. Araç Özellikleri

Tarımsal insansız kara aracının özellikleri mekanik, elektronik ve yazılımsal olmak üzere üç başlıkta incelenmiştir.

3.1 Mekanik Özellikler

Araç yarışma isterilerini gerçekleştirecek şekilde tasarlanmıştır. Yapılan ilk tasarımda araç paletli bir yapı olarak planlanmıştır. İlgili tasarım, hareket mekanizması bakımından 4*4 (4 çeker) yapıdadır. Aracın manevra mekanizması, tamamen hareket organları ile sağlanmaktadır. Aracın dönme yönüne göre tekerleklerin dönme yönü değiştirilerek araca sağa ve sola manevra yeteneği kazandırılmıştır. Yabani ot aracın sol kısmına yerleştirilmiş kamera ile algılanacaktır. Kameradan alınan fotoğraflar, görüntü işleme metotları ile işlenecektir. Araç mekanik olarak tarımsal arazide rahatlıkla dolaşabilecek ve istenilen görevleri başarıyla yerine getirebilecek şekilde tasarlanmıştır. Boyutları 110cm x 60cm x 30cm'dir. Aracın üretiminde şase ve PLA kullanılacaktır. Aracın geri kalan parçaları catia modelleme programı üzerinden modellenmiştir. Tasarım 3 boyutlu baskı makinelerinde PLA kullanılarak basılacaktır. Araç estetiğini arttırma, kolay montajlanabilme kabiliyeti ve düşük maliyeti sebebiyle 3 boyutlu baskı makineleriyle basılması uygun görülmüştür. 3 boyutlu baskıda PLA kullanılmasındaki amaç, PLA'nın bitki bazlı ve biyolojik olarak parçalanabilen bir plastik olmasıdır. Mekanik kısımların 3 boyutlu yazıcıda oluşturulmasının ardından motor sürücü katı planlanmış ve 4 adet

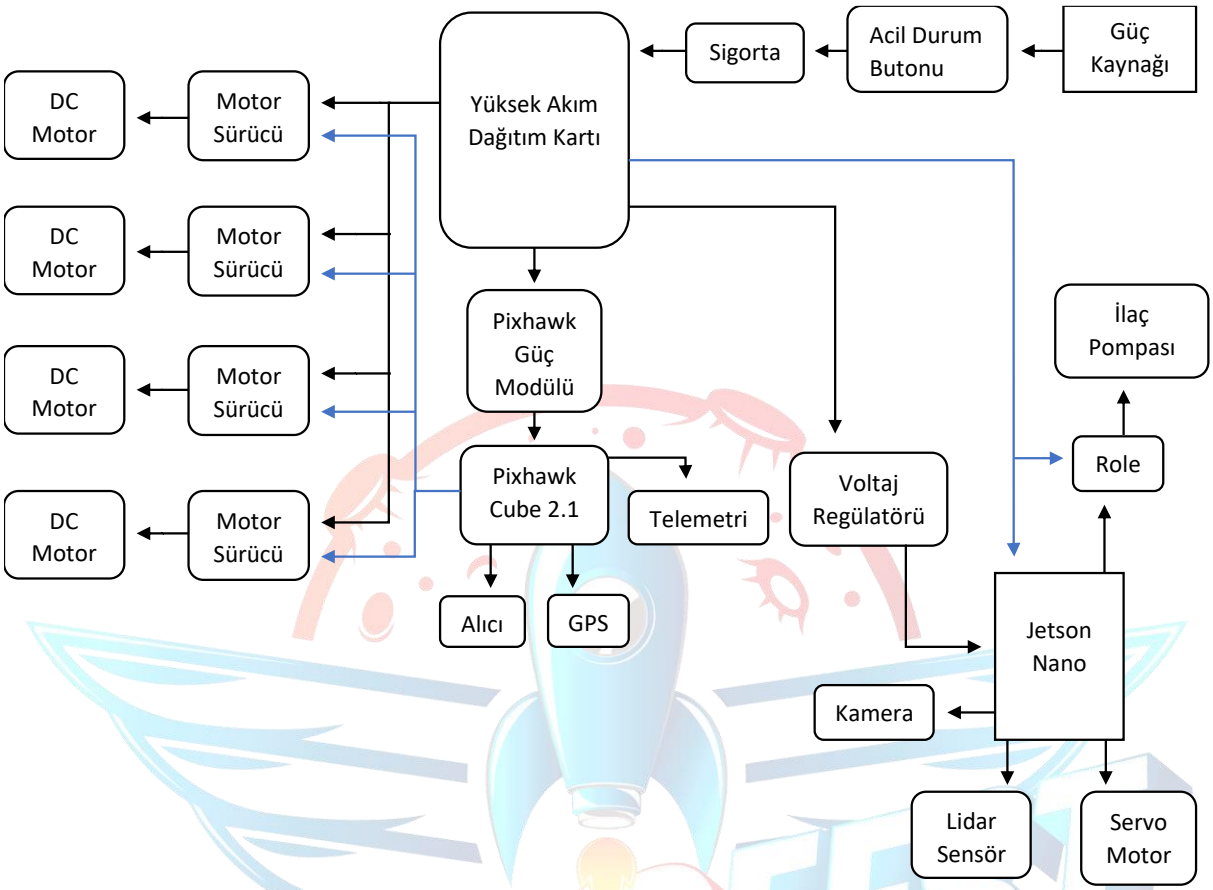
motor sürücü entegresi, şase içinde belirtilen motor sürücü katına yerleştirilecektir. Motor sürücü katı üzerine güç dağıtım katı eklenmiş ve batarya, regülatörler ve acil durum güç kesici sistem yerleşimi yapılacaktır. Fiziksel kombinasyonun yapılmasının ardından gerekli kablolamalar oluşturulacaktır. Güç dağıtım katının üzerine aracın beyni olan kontrol katı eklemesi yapılacaktır. Kontrol katında pixhawk ve Jetson Nano bulunmaktadır. Aracın üst kısmında ise verilen görevi gerçekleştirirken gerekli püskürtmeyi yapacağı ilaç kolu bulunmaktadır.



Şekil 3.1: Aracın Farklı Açılardan Tasarım Görselleri

Mekanik tasarım için büyük ölçüde ön tasarım raporuna sadık kalınmış eksik olan ilaç deposu ve pompası eklenmiştir.

3.2 Elektronik Özellikler



Şekil 3.2: Elektronik Konfigürasyon

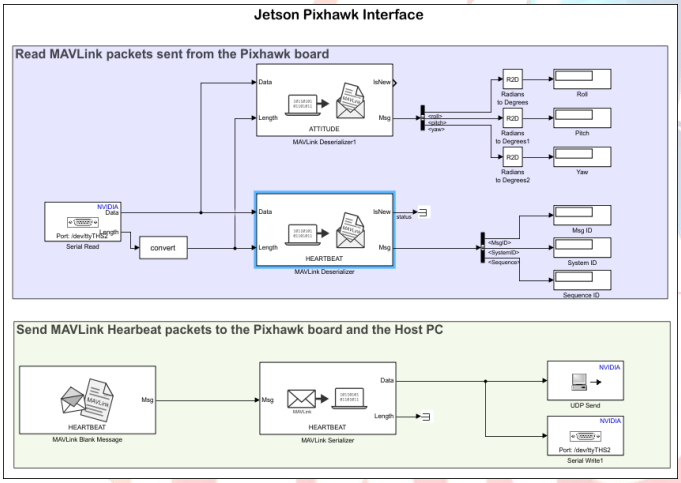
Aracımızda ana bilgisayar olarak Jetson Nano modülünü kullanacağız. Sensör verilerini yerel ağ bağlantısı üzerinden kontrol istasyonuna aktaracağız ve otonom görevleri, gerçekleştireceğiz. Motor sürücü kartlarımız ise bağlı olduğu motorun, hız, tork, yön gibi gerekli ayarların verisi pixhawk üzerinden alacak olup, motorlar ve kendisinin çalışabilmesi için gerekli gücü 12V regülatöre bağlı güç kaynağından alacaktır.

Görüntü işleme bilgisayarı ve nesne tespiti bilgisayarı olan Jetson nano yapay zeka için geliştirilmiş bir tek kartlı bilgisayardır. Bu bilgisayar ile TİKA'nın yabani ot tespiti ve yabani otun konumunun robot kola gönderilmesi sağlanacaktır. Jatson nanoya bağlanan yüksek çözünürlüklü kamera sayesinde görüntü işleme yapılacaktır. Ayrıca Jetson Nano, Pixhawk kontrol kartı ile Mavlink kütüphanesi üzerinden haberleşerek tüm temel hareketleri otonom olarak yönetecektir.

3.3 Yazılımsal Özellikler

Tarımsal insansız kara aracımız yarışma kapsamında yapacağı görevleri gerçekleştirirken nesne tespiti için Jetson Nano Geliştirme kartı üzerinden, Python dilini kullanmaktadır. Bu Python diliyle yazılan kod yapay zeka kullanarak bitkinin konumunu ve şeklini tanıyarak görevine uygun şekilde hareket etmesini sağlamaktadır. İlk olarak görüntü işleme ve nesne tespiti yapabilmesi için Respberry Pi kullanılması düşünülmüş, fakat performans ve yapay zeka kullanımında GPU farkından ötürü Jetson Nano kullanımı daha cazip görülmüştür.

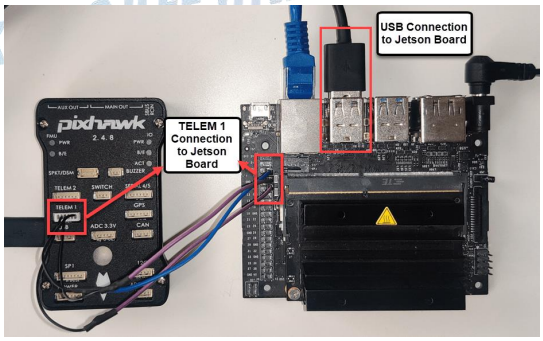
Mevcut tarım problemini detaylı bir şekilde araştırarak bu problemi çözmek için çeşitli metotlar geliştirdik. Tarladaki yabani otları algılamak için kullanılabilecek en verimli yöntemin yapay zekâ teknolojisi olduğuna karar verdik. Bu işlemi gerçekleştirmek için çeşitli yapay zekâ kütüphaneleri araştırılıp hız, fps çözünürlük ve doğruluk oranları karşılaştırıldı ve mevcut problemin çözümü için en faydalı modelin YoloV4 olduğuna karar verildi. Görüntü işleme için python üzerinden OpenCV kütüphanesinden faydalanılacaktır.



Şekil 3.3: Jetson Nano-Pixhawk Haberleşme Sistemi

Göreve başladıktan sonra Jetson Nano'ya bağlı olan kamera sayesinde sürekli görüntü alınarak daha önce eğitilen yapay zekâ algoritmasına gönderilir. Gönderilen bu veriler değerlendirilerek yabani ot tespit edilmeye çalışılır. Eğer sistem herhangi bir yabani ot algılarsa araç otu karşısına alacak şekilde manevra yapar. Ot ortalanınca bir servo motora bağlı olan ilaçlama mekanizması çalıştırılır ve ilaçlama işlemi gerçekleştirilmiş olur.

Bahsi geçen servo motor Jetson Nano üzerindeki GPIO pinlerinden faydalanılarak sürülecektir. İlaçlama işleminin doğruluğunu kontrol etmek amacıyla, ilaç içine eklenecek olan kimyasallar sayesinde görüntü işleme yapılacaktır. Bu kimyasalları algılamak için termal kamera ya da benzer bir teknoloji kullanılması hedeflenmektedir. Burada bahsedilen çalışma sistemi şekil 3.3'te basitçe anlatılmıştır.



Şekil 3.4: Jetson Nano-Pixhawk Bağlantısı

4. Sensörler

4.1 Kullanılan Sensörler

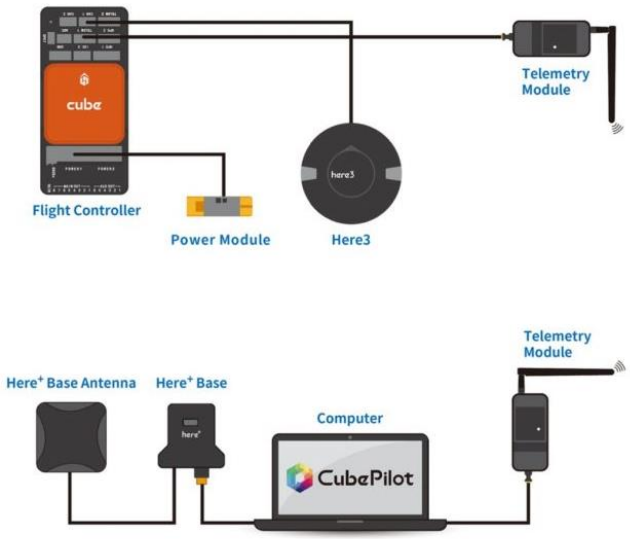
Kara aracımızda 2 adet raspberry pi v2 kamera,1 adet lidar sensör, 1 adet GPS, 1 adet IMU sensör yer almaktadır.

GPS sensörü

Aracımızda GPS sensörü olarak Here3 RTK GPS modülünü tercih ettik. 5 metrelik hassasiyete sahip bu modül konum kontrolü ve takibi yapmak için kullanılır. Mükemmel çekiyor hacı valla süpe

Özellikleri:

- Alıcı Tipi: u-blox M8 yüksek hassasiyetli GNSS modülleri (M8P)
- Uydu Takımyıldızı: GPS L1C / A, GLONASS L1OF, BeiDou B1I
- Konumlandırma doğruluğu: 3D FIX: 2,5 m / RTK: 0,025 m
- İşlemci: STM32F302
- IMU sensörü: ICM20948
- Navigasyon Güncelleme Hızı: Maks: 8Hz
- İletişim Protokolü: CAN
- Çalışma Sıcaklığı: -40 °C ila 85 °C
- Boyut: 76mm x 76mm x 16.6mm
- Ağırlık: 48,8 g



Şekil 4.1: RTK GPS

Lidar sensörü

LIDAR (*Laser Imaging Detection and Ranging*), lazer darbeleri kullanılarak bir nesne veya bir yüzeyin uzaklığını anlamaya yarayan teknolojidir. Radar teknolojisiyle benzerdir. Radarda kullanılan radyo dalgaları yerine ışık, yani lazer darbeleri kullanılır. 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.

Şekil 4.2: Lidar

Aracımızda nesne,engel tespiti için TF Mini LiDAR (ToF) Lazer Menzil sensörünü tercih ettik. Radyo dalgaları kullanılan diğer mesafe sensörlerine göre daha hassas ve hızlı kullanıldığı için tercih edilmiştir.TF Mini'nin maksimum algılama mesafesi 12 metredir, 6 metrede 100Hz örnekleme çözünürlüğünü destekler. Sinyal, parazitlerden etkilenmez ve dış ortamlarda kullanım için uygundur.TF Mini LiDAR, UART (TTL) haberleşme ara yüzünü kullanır, standart 5V çalışır ve ortalama güç tüketimi 0,6w'dir.

Kamera Sensörü

Aracımızın ön tarafında yol kamerası ve sol tarafında ise yabani ot tespit kamerası olmak üzere 2 adet raspberry pi v2 kullandık. Küçük ,hafif hd görüntü sunması, jetson nano ile uyumlu olması bu kamerayı tercih sebebimizdir.

Şekil 4.3: V2 Pi Kamera

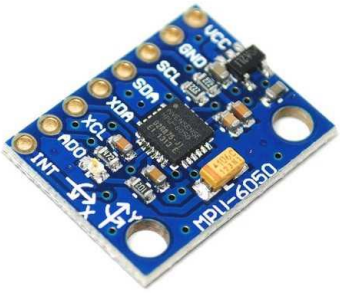
Teknik Özellikleri:

- Yüksek kaliteli görüntü algılama
- Büyük veri işleme kapasitesi
- 8 megapiksel sabit odak noktalı
- 1080p, 720p60 ve VGA90 destekli
- Sony IMX219PQ CMOS görüntü algılayıcı
- 15-pin şerit kablo

IMU sensör

Yolun eğimi ile ilgili bilgileri almak için MPU6050 ivme ve gyro sensör kartını kullandık. Ucuz ve kullanım kolaylığı sebebiyle bu sensörü tercih ettik. Üzerinde 3 eksenli bir Gyro ve 3 eksenli bir açısal ivme ölçer bulunduran 6 eksenli bir IMU kartıdır.

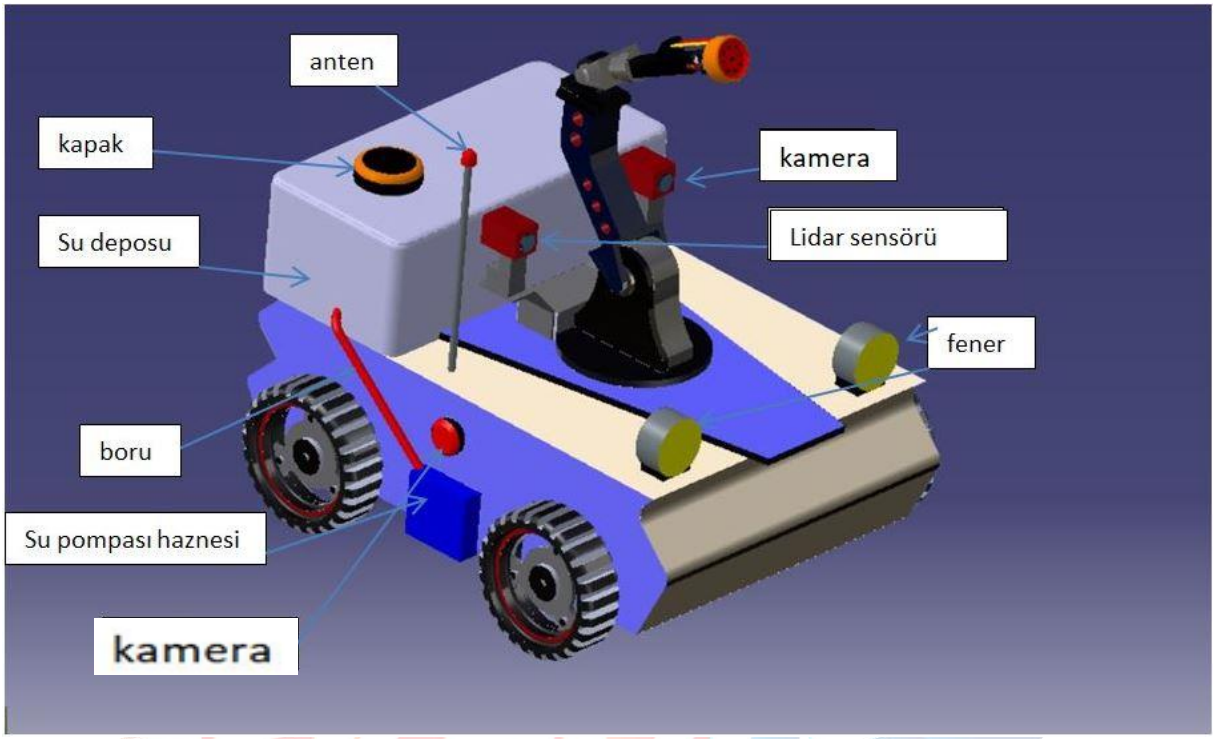
- Çalışma Gerilimi : 3.0 V – 5.0 V
- Gyro Ölçüm Aralığı : ±250, ±500, ±1000, ±2000 °/s
- Açısal İvme Ölçer Ölçüm Aralığı : ±2, ±4, ±8, ±16 g
- Haberleşme Arayüzü : I2C
- Boyutlar : 20.3mm X 15.6mm



Şekil 4.4: IMU Sensör

4.2 Sensör Konumları

Araçta kullanılan sensörlerin temsili yerleri tasarım üzerinde gösterilmiştir. Araçta 2 adet kamera kullanılmıştır öndeki kamera lidarla birlikte yol takibi görevini gerçekleştirmek için konumlandırılırken 2. Kamera aracın sağ tarafına yerleştirilmiştir ve otu tespit etmek için kullanılacaktır. Olası görüntü problemlerinin önüne geçmek için aracın önüne 2 adet fener yerleştirilmiştir. Bu sayede nesne tespiti daha kolay bir şekilde gerçekleştirilebilecektir.



Şekil 4.5: Sensör Yerleşimleri

TEKNOFEST
HAVACILIK, UZAY VE TEKNOLOJİ FESTİVALİ

5. Araç Kontrol Ünitesi

Tarımsal insansız kara aracının uzaktan kontrolü için araç tasarımında aracın beynini oluşturan NVIDIA Jetson Nano bilgisayarını ve PixHawk kontrol kartını kullandık. Pixhawk ve Jetson Nano haberleşmesi mavlink kütüphanesi üzerinden yapılacaktır. Jetson Nano otonom görevleri aynı anda yürütebilme, sensör verilerinin kontrol istasyonuna aktarılması, yabani ot algılama gibi görevleri üstlenir.

Şekil 5.1: Jetson Nano

JETSON NANO ÖZELLİKLER:

- GPU: 128 çekirdekli Maxwell ™ GPU
- CPU: dört çekirdekli ARM® Cortex®-A57 CPU
- Bellek: 4GB 64 bit LPDDR4
- Depolama: Micro SD kart yuvası (harici minimum 16G TF kart gerektirir)
- Video: Kodlama: 4K @ 30 (H.264 / H.265) Kod Çözme: 4K @ 60 (H.264 / H.265)
- Arabirimler: Ethernet: 10/100 / 1000BASE-T otomatik anlaşma Fotoğraf Makinesi: 12-ch (3x4 VEYA 4x2) MIPI CSI-2 DPHY 1.1 (1.5 Gbps) Ekran: HDMI 2.0, DP (DisplayPort) USB: 4x USB 3.0, USB 2.0 (Mikro USB) Diğerleri: GPIO, I2C, I2S, SPI, UART
- Güç: Mikro USB (5V 2A) DC jakı (5V 4A)
- Boyutlar: Çekirdek modül: 69,6 mm x 45 mm Bütün takım: 100mm × 80mm × 29mm

5.1 Haberleşme

Kara aracının kontrolü sağlanması için bir ara yüz programı bulunmaktadır. Araçta kontrollerin sağlanması araçtan gelen verileri işlemek ve ara yüzden araca veri göndermek için çift yönlü haberleşme sistemi olan XBee Pro 2.4 GHz 60 mW RF kablosuz haberleşme modülü yer

alacaktır. Ayrıca haberleşme mesafesini arttırmak için 2.4 GHz 4.5 Inch Yarım Dalga RP-SMA Xbee Antenle beraber kullanılacaktır.

Görüntü alma donanımı için NVIDIA Jetson Nano ile uyumlu Raspberry Pi Kamera Modülü kullanılmıştır. Pi kamera kullanılan tek kartlı bilgisayarla uyumlu çalışmasından ve kamera sensörünün çözünürlük kalitesinden dolayısıyla seçilmiştir.

Konum bilgisi için PixHawk kartının veri alabileceği GPS kullanılmaktadır. Konum verisindeki sapmayı en aza indirgeyebilmek için GNSS GPS kullanılmaktadır. Ayrıca acil durumlarda kullanılmak üzere Flysky FS-I6S Kumanda tercih edilmiştir. FS-16S aracın kontrolünü devralacaktır.

5.2 Yazılım

Kontrol yazılımı olarak kamera sensörünün yeterli çözünürlükte olması ve kullanılan tek kartlı bilgisayara uyumluluğundan dolayı YoloR algoritmasından faydalanılacaktır. Görüntü işleme için python üzerinden open cv kütüphanesi tercih edilmiştir.

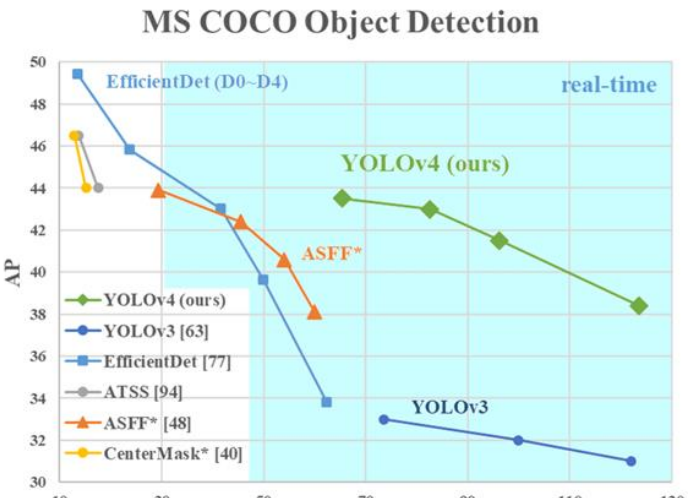
Araç güvenilirliğini sağlamak ve acil durumları önlemek için Acil durum uzaktan güç kesme sistemi kullanılacaktır. Acil durumlarda kullanım pilot kontrolüne verilmiştir.

6. Otonom Sürüş Algoritmaları

Otonom sürüş algoritmaları genel olarak aracın bulunduğu konumda, herhangi bir dış etkene veya araca zarar vermeden ve herhangi bir manuel (bir insan tarafından gerçekleştirilen) yönlendirmeye ihtiyaç duymadan araca önceden yazılımsal olarak tanımlanmış görevi eksiksiz ve sorunsuz bir şekilde yerine getiren algoritmaların bütünüdür.

6.1. Yabani Ot Algılama

Yabani otu algılamak için yapay zekâ teknolojisinden faydalanılmıştır. Bir bitkinin eğitimi, algılanması ve çalışma performansı açısından değerlendirilip en verimli algoritma araştırılmış ve bu algoritmanın YoloV4 algoritması olmasına karar verilmiştir. Darknet kütüphanesinden ve labelImg programından faydalanılarak gerekli eğitim işlemleri yapılmıştır.



Şekil 6.1: Eğitim Algoritmalarının Karşılaştırılması

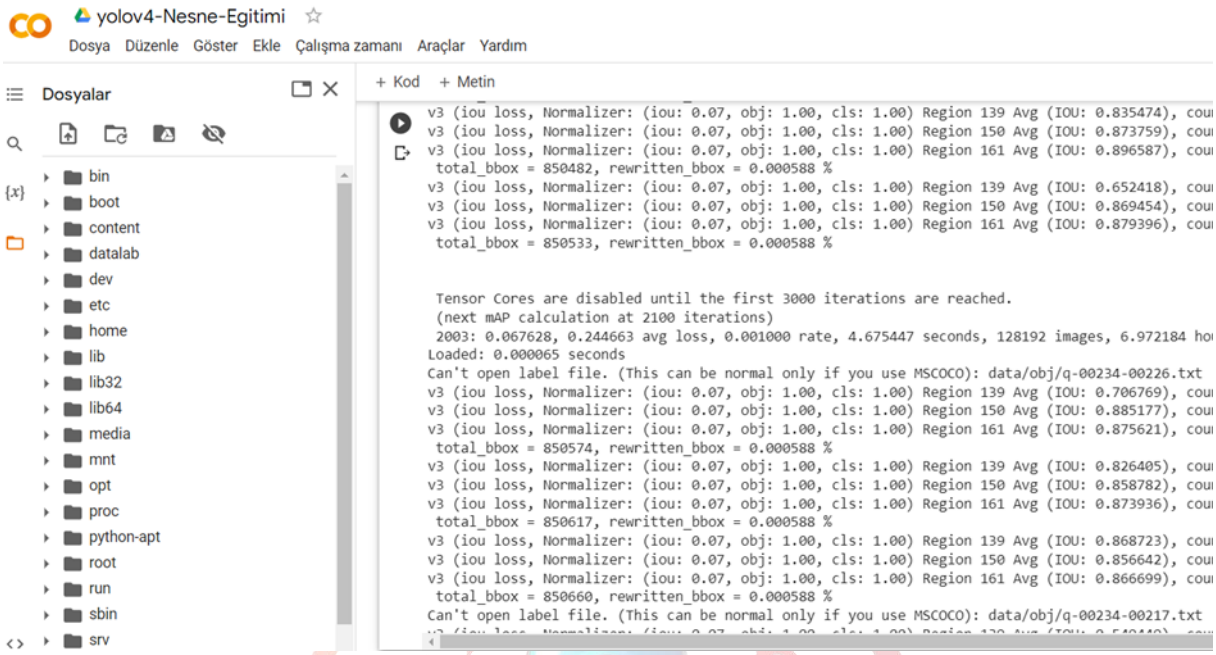
6.1.1. Verileri Etiketleme

Dataset oluşturmak için yukarıda da bahsedildiği gibi LabelImg programından faydalanılmıştır. 552 adet görsel “bitki” sınıf ismiyle etiketlenmiş ve “.txt” uzantılı dosyalara kaydedilmiştir (Yolo algoritması .txt uzantılı dosyaları eğitebilmektedir). Etiketlenen bu görsellerin 56 tanesi test için ayrılırken geriye kalan 496 görsel eğitim için kullanılmıştır.

Şekil 6.2: Veri Etiketleme Örneği

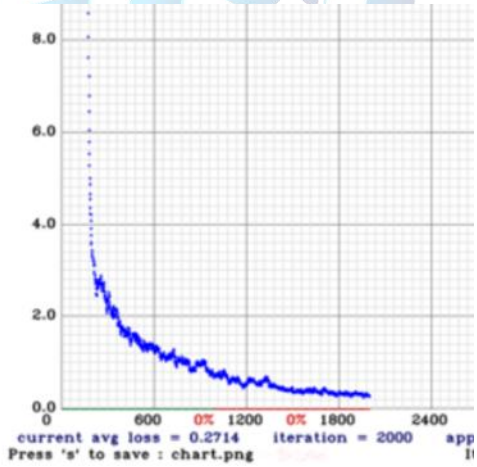
6.1.2. Veri Eğitimi

Etiketlenen verilerin eğitimi için Colab sisteminden faydalnılmıştır. Google Colaboratory veya Colab, Google Research'ten bir bulut hizmeti daha. Herhangi bir kullanıcının düzenleyicisine kaynak kodu yazmasına ve tarayıcıdan çalıştırmasına izin veren bir IDE'dir. Özellikle Python programlama dilini destekler ve makine öğrenimi görevlerine, veri analizine, eğitim projelerine vb. yöneliktir. Bu hizmete dayalı olarak Jupyter Not Defteri, barındırılıyor GMail hesabınızla tamamen ücretsiz, yapılandırma gerektirmez ve Jupyter'ı indirmeniz veya yüklemeniz gerekmez. Sunucularının GPU'ları vb. gibi kodunuzu düzenleyebilmeniz ve test edebilmeniz için size bilgi işlem kaynakları sunacaktır. Açıkçası, ücretsiz bir şey olarak, Google Colaboratory sınırsız kaynağa sahip değildir ve garanti edilmez, ancak sisteme verilen kullanıma göre değişir. Özetle colab kullanıcılara ücretsiz yüksek performanslı GPU hizmeti sunan bulut sistemidir ve kullanıcılar buradan kendi verilerini kolaylıkla eğitip yapay zekalarını kullanabilirler.



Şekil 6.3: Nesne Eğitimi

Colab'a yüklenen fotoğraflar ve etiket dosyaları Darknet kütüphanesinden yararlanılarak eğitilmeye başlanmıştır. Yaklaşık 2000 iterasyon eğitilmiştir. Eğitilen verilerin "avg. loss" değeri yukarıdaki görselde de görüleceği üzere 0,244663'e kadar düşmüş ve sağlıklı bir çıktı elde edebilecek seviyeye gelmiştir. Eğitim sonucunda elde edilen ağırlık dosyası test edilmiştir. Eğitim sonucunda oluşan eğitim grafiği görselde verilmiştir



Şekil 6.4: Hata Oranı Grafiği

6.1.3. Eğitilen Verinin Test Edilmesi ve Bitkinin Algılanması

Tüm etiketleme ve eğitim aşamaları bittikten sonra oluşan ağırlık dosyası test edilmek üzere indirilmiştir. Test görsel, video ve canlı kamera üzerinde test edilmiş ve bazı sonuçlar aşağıda paylaşılmıştır.

Şekil 6.5: Colab Üzerinden Yapılan Görsel Testi

Şekil 6.6: Bilgisayar Kamerasından Yapılan Real Time Test Alıntısı

Test Videosu Linki: https://youtu.be/uKiYjY3eRsw

Not: Video liste dışı olarak paylaşılmıştır. Linke sahip olan kişiler erişebilir.

```python
# -*- coding: utf-8 -*-
import cv2
import numpy as np

cap = cv2.VideoCapture(0)

while True:
    ret, frame = cap.read()

    frame = cv2.flip(frame,1)
    frame = cv2.resize(frame,(1200,800))

    frame_width = frame.shape[1]
    frame_height = frame.shape[0]

    frame_blob = cv2.dnn.blobFromImage(frame, 1/255, (416,416), swapRB=True, crop=False)

    labels = ["bitki"]

    colors = ["0,0,255","0,0,255","255,0,0","255,255,0","0,255,0"]
    colors = [np.array(color.split(",")).astype("int") for color in colors]
    colors = np.array(colors)
    colors = np.tile(colors,(18,1))


    model = cv2.dnn.readNetFromDarknet(r"D:/Programlar/AI/yolov4test.cfg",r"D:/Programlar/AI/yolov4-custom_last.weights")

    layers = model.getLayerNames()
    output_layer = [layers[layer[0]-1] for layer in model.getUnconnectedOutLayers()]
```

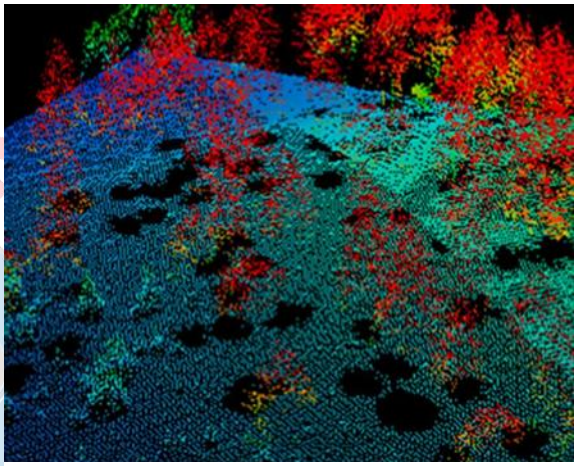
Şekil 6.7: Kamera ile Görüntü Alma ve Eğitilen Verinin Kullanıldığı Kod Bölümü

```python
for detection_layer in detection_layers:
    for object_detection in detection_layer:

        scores = object_detection[5:]
        predicted_id = np.argmax(scores)
        confidence = scores[predicted_id]

        if confidence > 0.20:

            label = labels[predicted_id]
            bounding_box = object_detection[0:4] * np.array([frame_width,frame_height,frame_width,frame_height])
            (box_center_x, box_center_y, box_width, box_height) = bounding_box.astype("int")

            start_x = int(box_center_x - (box_width/2))
            start_y = int(box_center_y - (box_height/2))



            ids_list.append(predicted_id)
            confidences_list.append(float(confidence))
            boxes_list.append([start_x, start_y, int(box_width), int(box_height)])


max_ids = cv2.dnn.NMSBoxes(boxes_list, confidences_list, 0.9, 0.9)
```

Şekil 6.8: Bitkinin Hangi Oranlarda Algılayacağını Belirlediğimiz ve İşaretlediğimiz Bölüm

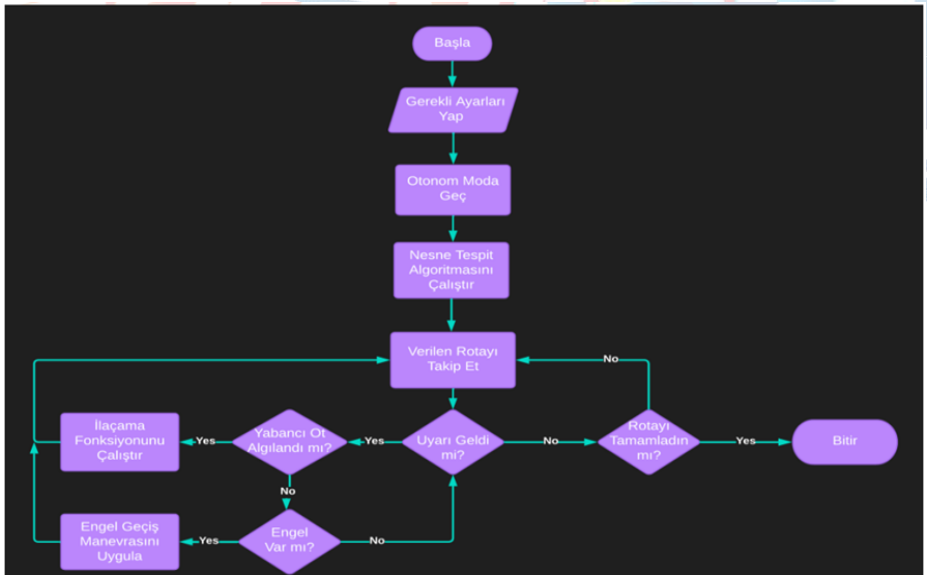
6.2. Yol Takip Algoritması

Aracımız yarışma alanına bırakılıp ilk aşamada manuel olarak hareket güzergahını takip edecek ve her saniye GPS konumları kaydetmeye başlayacak. Kaydedilen bu GPS konumları sırayla pointlere atanacak ve araç otonom olarak yarışmaya başladığında bu pointleri tek tek takip etmeye başlayacak ve bu aracımızın rotasını oluşturacak. Araç rotayı takip ederken asenkron olarak bitki tespit kamerası sürekli çalışacak ve araç bitkiyi algıladığı zaman araç duracak ve ilaçlama fonksiyonunu çalıştıracak. Konuyla ilgili algoritma aşağıda verilmiştir.



Şekil 6.9: Lidar Sensör Haritası

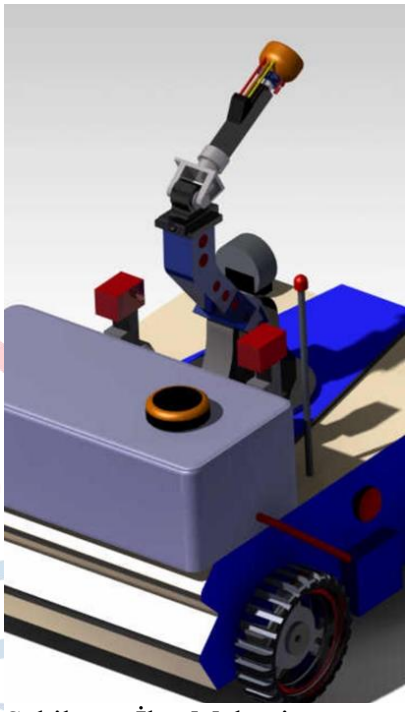
Aracımızın hareketlerini en kusursuz şekilde yapması ve farklı durumlara karşı en iyi şekilde cevap vermesi için aracımızı Pixhawk kontrol kartı kullanarak hareket ettirmeyi planlıyoruz. Pixhawk ve Jetson Nano haberleşmesi mavlink kütüphanesi, müdahale sistemi ile haberleşmesi ise SPI protokolü üzerinden yapılacaktır. Pixhawk ile birlikte Lidar sensör kullanılması hedeflenmektedir. Lidar sensör sayesinde çevredeki bitki ya da engellere ne kadar yaklaşılacağına karar verilecektir. Ayrıca IMU, encoder ve gps gibi sensörlerden faydalanarak aracın konum ve pozisyon bilgilerini elde edeceğiz. GPS verilerine ulaşamadığımız ya da bağlantı sorunları yaşadığımız durumlarda NoGPS moda geçerek görevi yapmaya ara vermeden devam edeceğiz. Pixhawktan erişebileceğimiz batarya seviyesi gibi verilerle aracın görevi sağlıklı bir şekilde yapmasına olanak sağlamayı hedefliyoruz.Araç herhangi bir engelle karşılaştığında devreye girecek olan engel geçiş manevrası, Lidar sensöründen gelen verileri sürekli tarayarak aracın engeli sorunsuz aşmasını hedefleyen manevradır. Bu manevrada araç engelle arasındaki mesafeyi sabit tutarak engeli aşacak ve kalan rotasını tamamlayacaktır. Aracın gideceği yolu belirlemek için araç üzerindeki Jetson Nanoya bağlı olan kameradan sürekli görüntüler alınacaktır.



Şekil 6.10 Görev Algoritması

7. Yabancı Otla Zirai Mücadele Yöntemleri

Yabani otla müdahale ilaç püskürtme yöntemi kullanarak yapılacaktır. İlaç, su pompası aracılığı ile yabani otların üzerine püskürtülecektir. İlaçlama mekanizması, bitki için gereken ideal ilaç miktarını efektif bir şekilde bitkiye tatbik edilmesini sağlayacaktır. İlaç bitkiye oldukça yakın bir mesafeden göreceli yüksek bir basınçla uygulanacağı için, uygulanan ilacın havaya ve toprağa karışma miktarı ihmal edilebilir bir seviyeye inecek, böylelikle uygun dozda ilaçlama yapmak mümkün olabilecektir. Bu uygulama vesilesiyle, gereğinden fazla ilaç kullanmanın doğaya verdiği zararı azaltacak olup, aynı zamanda ürün kalitesini ve verimini artıracaktır. Yabani ot ile mücadele ilaç sıkma yöntemi ile olacaktır. Sıkılacak olan ilaç pompa yardımı ile ota püskürtülecektir. Görüntü işleme kameramız aracın sağında ilaç pompasının üst kısmında yer alacaktır. Öncelikle servo ile ilaç kolumuz kameranın bulunduğu yöne hareket ettirilecektir ve orada sabit kalacaktır. Aracın yabani ot tespiti ve ilaçlama yapması için bitkileri sağına alması gerekmektedir. Bu da yazılım ile sağlanacaktır. Uygulanabilirlik açısından bu yöntem uygun görülmüştür. Bir aksaklık çıkması halinde aracın diğer tarafına da bir kamera eklenip kolun bitkilerin araca olan konumuna göre 180 derece hareket ettirildikten sonra yabani ot tespiti ve ilaçlamanın yapılması düşünülmüştür. İlaç sistemimiz ; ilaç deposu, ilaç pompası, sıvı taşıma hortumu, ve ilaç kolundan oluşmaktadır. İlaç depo hacmi aracın boyuna göre uygun olması açısından 1.5 L düşünülmektedir



Şekil 7.1: İlaç Mekanizması

7.1 YAZILIMSAL

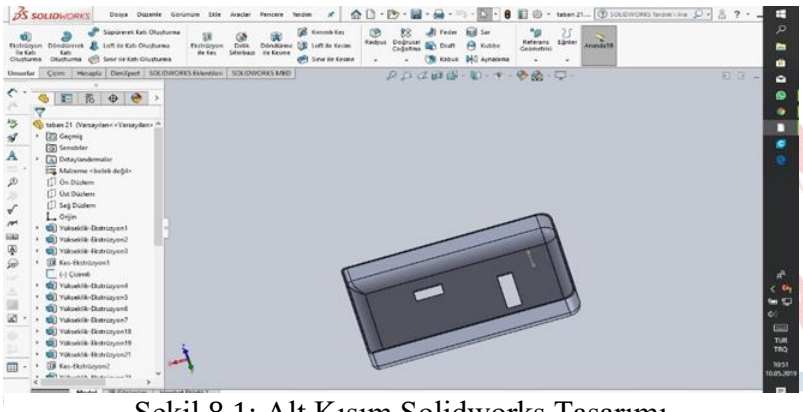
Aracımız hareket halinde iken kameralar aracılığı ile veriler alır. Alınan bu veriler pi kameranın bağlı olduğu Jetson Nano'ya gönderilir. Jetson Nano kullandığı Python tabanlı kodlar aracılığıyla, veri tabanında bulunan zararlı ot resimleri ile kıyaslar. Bunun üzerine zararlı ot, oluşturulan tahmin ile tespit edilir ve Jetson Nano'ya gönderilir.

7.2 DONANIMSAL

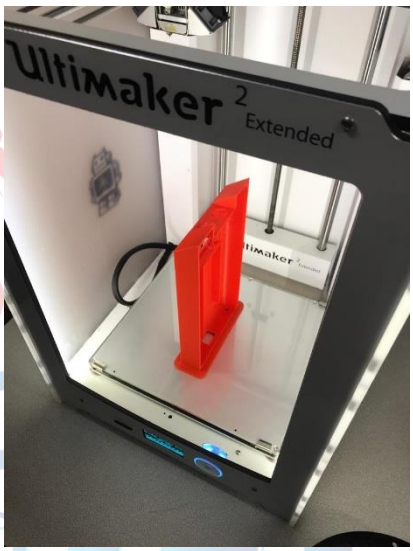
Alınan verilerle yabani ot tespit edilmeye çalışılır şayet yabani ot tespit edilirse aracımız manevra yaparak Jetson Nano üzerindeki GPIO pinlerinden faydalanılarak sürülecek olan Servo Motor sayesinde ilaçlama mekanizmasını çalıştırır ve üzerindeki pompa ile ilacı yabani ota püskürtür.

8. Özgün Bileşenler

Aracımızın mekanik bileşenlerinin büyük bir kısmı tasarım ekibimiz tarafından tasarlanıp geliştirilmiştir. Tasarlanan bileşenler, üretim ve montaj aşamaları göz önünde bulandırılarak olabildiğince üretile bilirliği yüksek ve piyasada bulunan bağlantı elemanlarıyla kolaylıkla montaj yapılabilecek şekilde tasarlanmıştır. Alt ve üst şase alüminyum saç malzemeden lazer kesimle üretilmiştir. Şase üzerine yerleştirilecek ana gövde üç boyutlu yazıcı ile basılmıştır.



Şekil 8.1: Alt Kısım Solidworks Tasarımı



Yabani Otla Mücadele Sistemi; kamera konumuna dönme kabiliyetine sahip 3B baskı ile üretilecek bir mekanizmadır. Bu mekanizma aracımızda bulunan depodan pompa ve hortumlar yardımıyla ilacın yabani ota püskürtülmesi sağlamaktadır.

9. Güvenlik Önlemleri

Araçta bulunan elektronik kısımlar ve motor darbelere ve sarsıntılara karşı zarar görmemeleri için son derece korunaklı bir şekilde yerleştirilecektir

Aracımızın kontrolden çıkması halinde kumanda ile kontrol edilip araç durdurulacaktır. Yüksek akım çekilmesi ya da kısa devre gibi durumlar için sigorta konulmuştur. Aracın üzerinde ve kontrol istasyonunda olmak üzere iki tane acil stop butonu tercih edilmiştir. Tasarladığımız acil stop butonu yanda gösterilmiştir. Test linki aşağıda verilmiştir. Ayrıca pil paketi, olası bir yangından korunmak için yanmaz bant ile sarılacaktır. Herhangi bir kaza anında veya planlananın dışında bir şey olması halinde aracımızda kullanılan ekipmanların zarar görmesini engellemek için ekipmanlar, şasenin içerisinde sabitlenmiş bir şekilde bulunacaktır.

Şekil 9.1: Uzaktan Güvenlik Butonu

Link: https://youtube.com/shorts/3EbMTzbxyrw
Not: Videolar liste dışı olarak paylaşılmıştır. Linke sahip olan kişiler tarafından görünecektir.

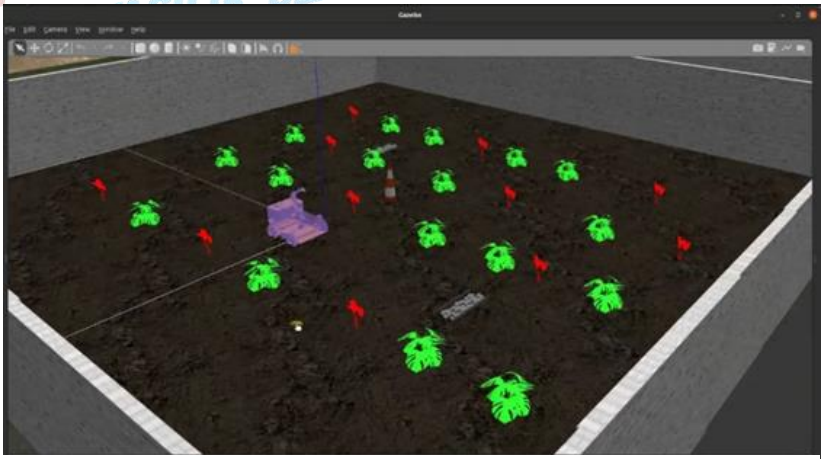
Şekil 9.2 Devre Kesici Anahtar

Devre kesici, anormal gerçekleşen akımı kesmeye yarayan bir anahtar cihazı olarak bulunuyor. Mekanik bir cihaz olan devre kesiciler, aşırı yüksek bir akımın akışını bozduğu gibi ayrıca bir anahtar işlevini de yerine getiriyor. Devre kesiciler esas olarak bir elektrik devresinin kapatılması veya açılması için tasarlanıyor. Böylece elektrik sistemini hasardan koruyor.

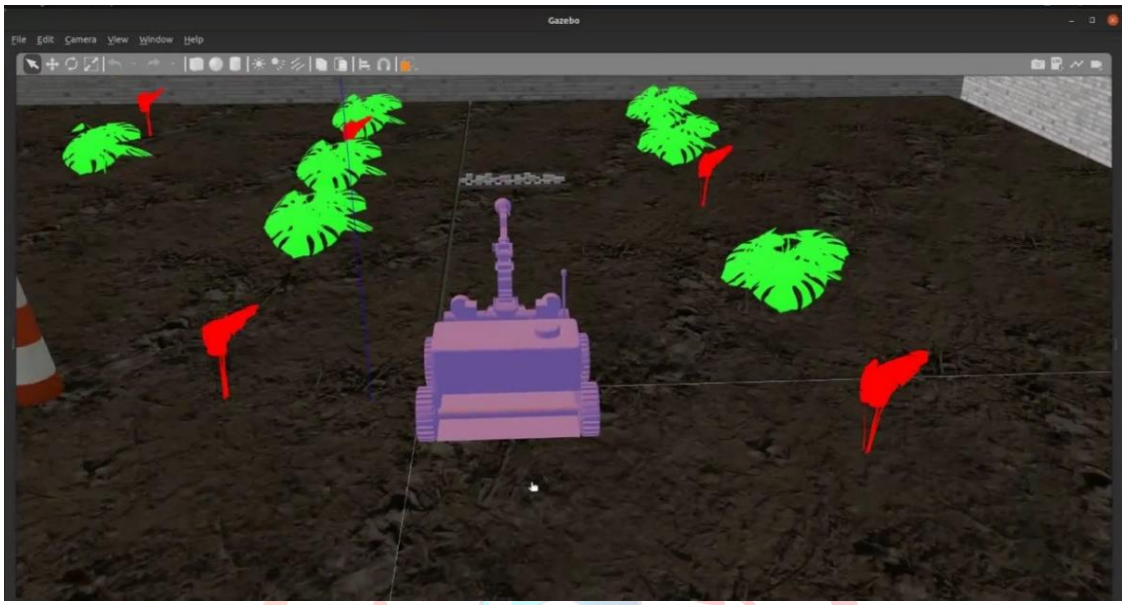
10. Simülasyon ve Test

10.1 Simülasyon

Araç için geliştirilen otonom algoritmasının denenmesi için Gazebo üzerinde bir simülasyon ortamı oluşturulmuştur. ROS robotlarda yaygın olarak kullanılan bir yazılımdır. Kod üzerinden çok az değişiklik yaparak diğer robotlar üzerinde de çalışabilecek bir yazılım parçası geliştirmeyi hedefler. ROS sayesinde standart bir robot için yazılan standart bir algoritma yükle-kullan hızında çalıştırılabilir konumda olacaktır. Robota takılan sensörlerden alınan veri ROS ile bilgisayara ulaşacaktır, bilgisayarda işlenen veri robota bir görev olarak geri dönebilecektir. Aradaki bu haberleşme sistemi ROS arayüzündeki topicler ve mesajlar yardımıyla yapılacaktır. ROS, Gazebo simülasyon ortamıyla tam uyumlu olarak çalışmaktadır. Bu sayede projenin prototip aşamasında ROS Package haline getirilen geliştirilen veri yapısını test etmek için Gazebo simülasyon ortamı hızlı bir geliştirme için kullanılabilir. Oluşturulan simülasyon ortamında otların yerleri her test edildiğinde yerleri karıştırılarak yarışma ortamına yakın simülasyon edilerek başarı oranı artırılmıştır. Gazebo ortamını diğerlerinden ayıran özellik olarak araç üzerinde pixhawk kullanılacağından dolayı araca yapay pixhawk olanağı sağlamaktadır ve simülasyon sonuçları aynı kodlar yazıldığında %85 uyumlu çalışmaktadır. Gazebo simülasyon ortamı şekil 10.1 de görülmektedir.



Şekil 10.1 Gazebo Görev Simülasyonu



Şekil 10.2: Gazebo Simülasyon Ortamı

ROS tabanlı gazebo simülasyon programında test edilen kodlar Şekil x'de görülmektedir. Şekil 10.3 deki kodlar İKA'da kullanılacak olan ve temel hareket mekanizmalarını sağlayacak olan pixhawk kontrol kartı ile MavLink haberleşme kütüphanesi ile Jetson Nano'nun haberleşmesi simülasyon ortamında kurularak bağlantının oluşturulması kontrol edilmiştir. Araca temel hareket manevraları ve daha öncenden belirlenen GPS konumlarına hareket etme kodları testi başarıyla gerçekleştirilmiştir.

```
print("Baglaniliyor")
baglanti_yolu = '/dev/ttyACM0'                  # Pixhawk ile mikroişlemci arasındaki bağlantı yolunu belirtir.

global drone
drone = connect(baglanti_yolu, wait_ready=True)   #              Pixhawk ile mikroişlemci ile bağlantısını yapar.
global vehicle
vehicle = drone

print("motorlar arm ediliyor")

vehicle.armed = True
time.sleep(5)
vehicle.mode = VehicleMode("GUIDED")              # Aracı otonom hareket ettirmek için GUIDED modona alınmalıdır.
while not vehicle.armed:
    print(" Arm edilmesi bekleniyor...")
    time.sleep(1)
```

```python
import cv2, imutils

def pozisyon(posx, posy, yaw_rate, posz, vehicle):
    msg = vehicle.message_factory.set_position_target_local_ned_encode(
        0,
        0, 0,
        mavutil.mavlink.MAV_FRAME_BODY_NED,
        0b0000011111111000,
        posx, posy, posz,
        0, 0, 0,
        0, 0, 0,
        0, math.radians(yaw_rate))

    vehicle.send_mavlink(msg)

pozisyon(x, 0, 0, 0, vehicle)
time.sleep(1)
print('ileri gidiyorum')

pozisyon(-x, 0, 0, 0, vehicle)
time.sleep(1)
print('geri gidiyorum')

pozisyon(0,0, 90, 0, vehicle)
time.sleep(1)
print('sağa dönüyorum')

pozisyon(0,0, 270, 0, vehicle)
time.sleep(1)
print('sola dönüyorum')

pozisyon(0,0, 180, 0, vehicle)
time.sleep(1)
print('geri dönüyorum')

pozisyon(0,0, y, 0, vehicle)
time.sleep(1)
print('y derece dönüyorum')
```

```python
def konuma_git(konum):                                          # Verilen GPS konumuna 2 metre kalana kadar gid
    vehicle.airspeed = 3                                        # Kalan manevraları ve kalan mesafeyi Lidar sen
    vehicle.groundspeed = 7
    point = konum
    print("Koordinat noktasına ilerliyorum")
    vehicle.simple_goto(point)
    while True:
        current_loc = [vehicle.location.global_relative_frame.lat, vehicle.location.global_relative_frame.lon]
        gidilecek_konum = [point.lat, point.lon]
        fark= np.array(np.absolute(current_loc)) - np.array(np.absolute(gidilecek_konum))
        mesafe = fark * 111000
        a = mesafe[0]
        b = mesafe[1]
        kalan_mesafe1 = (a ** 2) + (b ** 2)
        kalan_mesafe = np.sqrt(kalan_mesafe1)
        print("kalan mesafe: %s" % kalan_mesafe)
        time.sleep(1)
        if (kalan_mesafe < 2):
            print("Hedef noktaya ulaşıldı")
            break
```

Şekil 10.3: Gazebo Test Kodları

10.2 Test

Yarışmada kullanılacak olan temel ön tasarım oluşturulup aracın genel kapsamda hareket kabiliyetleri kontrol etmek için testler yapılmıştır.

Araç üzerindeki elektronik aksam laboratuvarda bulunan malzemeler ile araç elektroniği temel hareket ve testleri yapmak için oluşturulmuştur. Şekil 10.3'de görülen prototip araç testlerde kullanılmıştır. Testler araç temel hareket kabiliyeti, engelden geçme ve 15 derece eğim oluşturulup oluşturulan eğimden geçme testleri yapılmıştır. Araç tüm testleri başarılı bir şekilde geçmiştir. Amaç araç mekaniğinin optimum hale getirilmesidir.

Şekil 10.4: LETO V1 Prototip Çalışması

Şekil 10.5: Engel Testi

Şekil 10.6: Eğim Testi

Not: Test videolarının linkleri aşağıdaki gibidir. Videolar liste dışı olarak paylaşılmıştır. Linke sahip olan kişiler tarafından görünecektir.

Hareket Videosu: https://youtube.com/shorts/QnRZGD-hMug
Engel 1 videosu: https://youtu.be/y6z1zyzKhik
Toprak arazi ve engel 2 videosu: https://youtu.be/88oROr1YLqg

11. Referanslar (Kaynakça)

[1] A. Sathi, T. E. Morton, and S. F. Roth, “Callisto: An Intelligent Project Management System”, AIMag, vol. 7, no. 5, p. 34, Dec. 1986.
[2] T.Rajani Devi, V.Shobha Reddy, “Work Breakdown Structure of the Project”, International Journal of Engineering Research and Applications (IJERA), vol. 2, pp.683-686, Mar-Apr 2012.
[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 v4-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] T. Wang, Y. Wu, J. Liang, C. Han, J. Chen ve Q. Zhao, «Analysis and Experimental Kinematics of a Skid-Steering Wheeled Robot Based on a Laser Scanner Sensor,» Sensors, cilt 15, no. 5, p. 9681–9702, 2015.
[6] «Understanding Sensor Fusion and Tracking,» [Çevrimiçi]. Available: https://www.mathworks.com/videos/sensor-fusion-part-1-what-is-sensor fusion1569410785813.html. [Erişildi: 28 Haziran 2021].
[7] «Intel RealSense,» [Çevrimiçi]. Available: https://www.intelrealsense.com/depthcamera-d435i/. [Erişildi: 27 Haziran 2021].
[8] «MAVLink Developer Guide,» [Çevrimiçi]. Available: https://mavlink.io/en/. [Erişildi: 25 Haziran 2021].
[9] https://ecorobotix.com/en/avo/
[10] Mennan H., Uygur F.N. (1994). Samsun ili buğday ekim alanlarında görülen yabancı otların saptanması. OMÜ Ziraat Fakültesi Dergisi, 9(2):25-35, 1994.

TEKNOFEST
HAVACILIK, UZAY VE TEKNOLOJİ FESTİVALİ