TEKNOFEST

HAVACILIK, UZAY VE TEKNOLOJİ FESTİVALİ

ROBOTAKSİ – BİNEK OTONOM ARAÇ YARIŞMASI KRİTİK TASARIM RAPORU

TAKIM ADI : O GACA

TAKIM ID : 463971

.

İÇERİK

1. Takım Organizasyonu **3**

2. Ön Tasarım Raporu Değerlendirmesi **4**

3. Araç Mekanik Özellikleri **8**

4. Donanım Mimarisi **16**

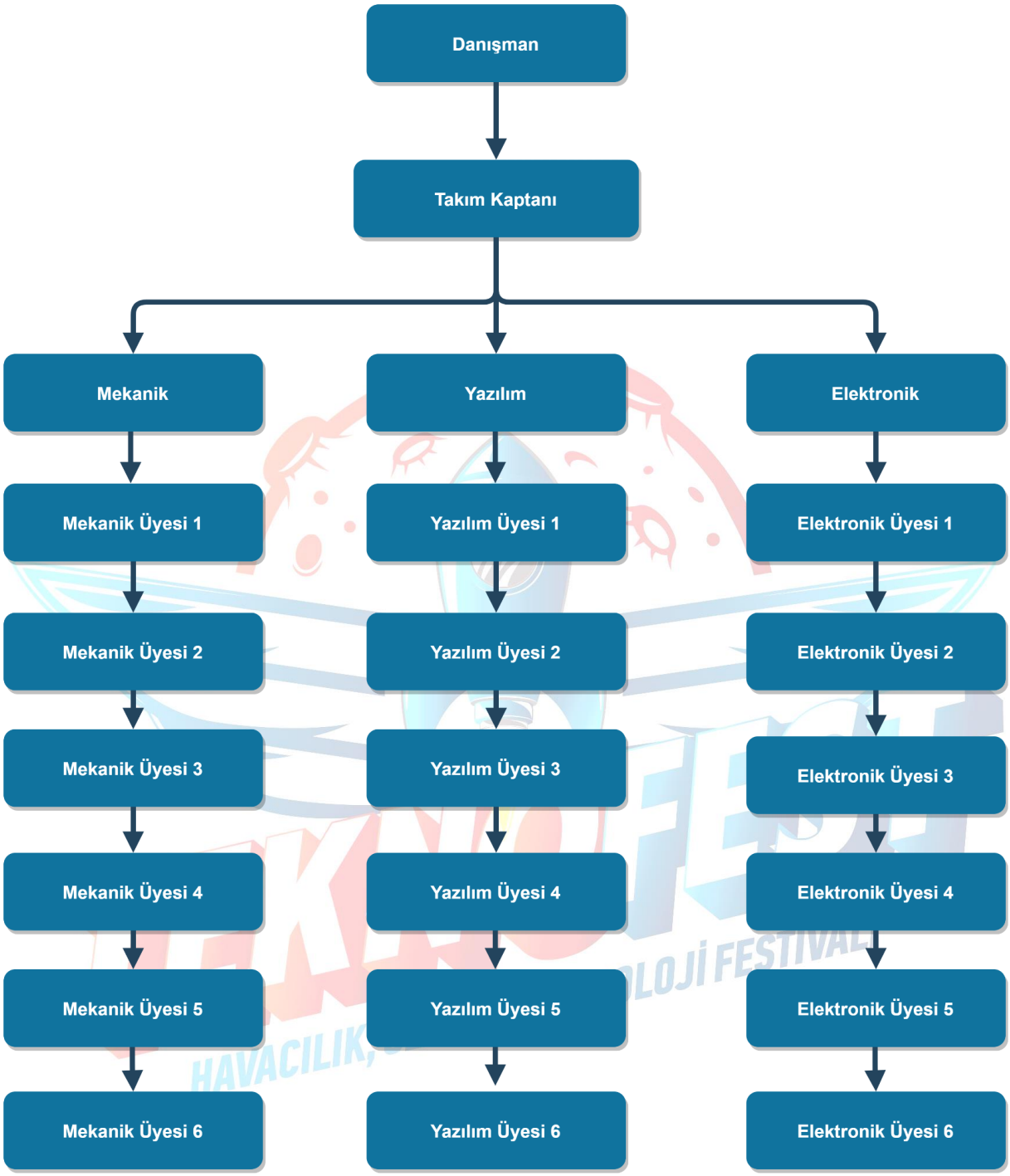
5. Yazılım Mimarisi **30**

6. Özgün Bileşenler **50**

7. Test **51**

8. Referanslar **68**

1. Takım Organizasyonu



Tablo 1: Organizasyon tablosu

Mekanik Üyesi 1: Elektronik Gaz Pedalı Montajı ile ilgilenmiştir.

Mekanik Üyesi 2: Elektronik Fren Montajı ile ilgilenmiştir.

Mekanik Üyesi 3: Elektronik Direksiyon Montajı ile ilgilenmiştir.

Mekanik Üyesi 4: Kaynak İşleri ile ilgilenmiştir.

Mekanik Üyesi 5: Üç boyutlu tasarımla ilgilenmiştir.

Mekanik Üyesi 6: İmalat süreci ile ilgilenmiştir.

Yazılım Üyesi 1: Şerit Tanıma ve Takibi ile ilgilenmiştir.

Yazılım Üyesi 2: Trafik İşareti Tanıma ve Takibi ile ilgilenmiştir

Yazılım Üyesi 3: Araç testleri ile ilgilenmiştir.

Yazılım Üyesi 4: Park Algoritması ile ilgilenmiştir.

Yazılım Üyesi 5: Araç Kontrol Yazılımı ile ilgilenmiştir.

Yazılım Üyesi 6: Simülasyon yapımında görev almıştır.

Elektronik Üyesi 1: Araç Kart Tasarımı ile ilgilenmiştir.

Elektronik Üyesi 2: Elektronik Devre Montajı ile ilgilenmiştir.

Elektronik Üyesi 3: Elektronik Fren Montajı ile ilgilenmiştir.

Elektronik Üyesi 4: Kamera Montajı ile ilgilenmiştir.

Elektronik Üyesi 5: Güvenlik Sistemi Montajı ile ilgilenmiştir.

Elektronik Üyesi 6: Kablolama işlemlerini yapmıştır.

2. Ön Tasarım Raporu Değerlendirmesi

Tırnak içinde verilen cümleler Robotaksi ÖTR Hakem Değerlendirmesinden alıntı yapılmıştır.

"Ön tasarımda planlanan bütçe ve son bütçe arasında karşılaştırma yapılır."
ÖTR raporumuzun geçmesinin ardından DDK hakemlerinin yazdığı geri bildirimleri tek tek inceledik ve bu eksikleri nasıl telafi edebileceğimizi, takım üyelerimizce yaptığımız yenilikleri size nasıl aktarabileceğimizi tartıştık.

"Sunulan bilgilerin daha detaylı olması gerekir."

Bu hususta haklı olduğunuzu ancak ÖTR'nin bir sayfa kısıtlaması bulunduğunu belirtmek istiyoruz. ÖTR'de son sayfasına kadar her şeyi açıklamaya çalıştık. Bahsedilen eksikleri KTR

raporunda istenilene uygun bir biçimde açıklamaya çalışacağız. Lise grubu olarak bu işlerde çok yeniyiz ve meşguliyetimizi uygulamasını yapmaya ayırdığımızdan terimleri ifade etmekte zorlanıyoruz. Ancak elimizden geleni yapıyoruz, yapacağız.

"Sensörlerin seçim nedeni belirtilmemiştir"

Elektronik takım üyeleri kullanılan yahut ileriki süreçte kullanılacak sensörleri tablolarla listelemiş ve görsellerle desteklemiştir. Aracın üzerinde bulunan tüm sensörler araç şasesi çiziminin üzerinde oklar ve yazılarla belirtilmiştir.

Ultrasonik sensör

Yapılan araştırmalar sonucu URM06-UART ultrasonik sensörü kullanmanın proje açısından daha avantajlı olduğuna karar verildi. Küçük boyutuyla entegrasyonu kolaylaştırması, her tür mikrodenetleyici ile uyumlu olması, şeffaf hedefleri bile algılayabilmesi ve ayrıca fiyat performans ürünü olması sebebiyle en optimize seçeneğin bu olduğu belirlenmiştir..

Kamera

Otonom aracın ve çevrenin güvenliği açısından aldığımız bir önlemdir. Aracın kör noktalara karşı duyarlılığını artırmak amaçlanmıştır.

Rotary Encoder

Tekerleğin dönme hareketine karşılık üretilen elektrik sinyalini yazılan dönüş algoritmasına optimize ederek dönüş hızını en hassas şekilde tespit etmeyi ve donanımla haberleşmeyi hedeflemekteyiz.

Lineer Cetvel

Encoder kullanımının yanında ölçümümüzü daha sağlıklı ve hassas bir hale getirmek adına lineer cetvel kullanmaya karar verilmiştir.

Sıcaklık Sensörü

Sürüş sırasında gereken ani önlemler içim motorlara taktığımız 8 adet sıcaklık-ısı sensörleri aşırı ısınan bölgeleri algılayarak ana motor sürücüsüyle haberleşecektir.

"Otonom sürüşte stereo kameralar ve 3b lidar kullanılması avantaj sağlayacaktır. Ultrasonik sensör kullanılabilir ancak yeterli değildir."

Hakemlerden aldığımız dönüşler ve ultrasonik sensörle yaptığımız testler sonucunda engel tanıma hususunda sensörün tek başına yeterli olmadığına karar verdik. Ultrasonik sensör

yeterince geniş açılarda çalışamadığından lidar da ışık faktöründen olumsuz etkilendiğinden daha sağlıklı bir çevre algılama için hem lidar hem de ultrasonik sensör kullanmaya karar verdik.

“ Görüntü işleme için kullanılması planlanan veritabanı ya da tabanları hakkında bilgiler verilebilir. Takımın varsa ön çalışmalardan sonuçlar verilebilir. Görüntü işleme ve otonom sürüş algoritmaları hangi kartlar üzerinde çalıştırılacağı verilebilir.”

Otonom sürüş algoritması olarak da kullandığımız YOLOV4 algoritmasıyla uyumlu olan ve TXT tabanlı bir veri tabanı olan YOLO DARKNET’i kullandık. Veri setimizi Roboflow üzerinden 3 aşamada (training-test-validation) üzerinden eğiterek oluşturduk.

“Kablosuz haberleşme sistemi ile ilgili daha fazla ayrıntı verilebilir. Tüm sisteme ait blok şema verilebilir. Devre şeması bölümler halinde verilmesi okunurluğu artıracaktır.”

“Kablosuz haberleşme sistemi biraz daha geniş açıklanabilir. Yazılımsal güvenlik önlemleri açıklanmamıştır. Detaylandırılabilir. Yarışmacı güvenliği konusunda önlemler eklenebilir.”

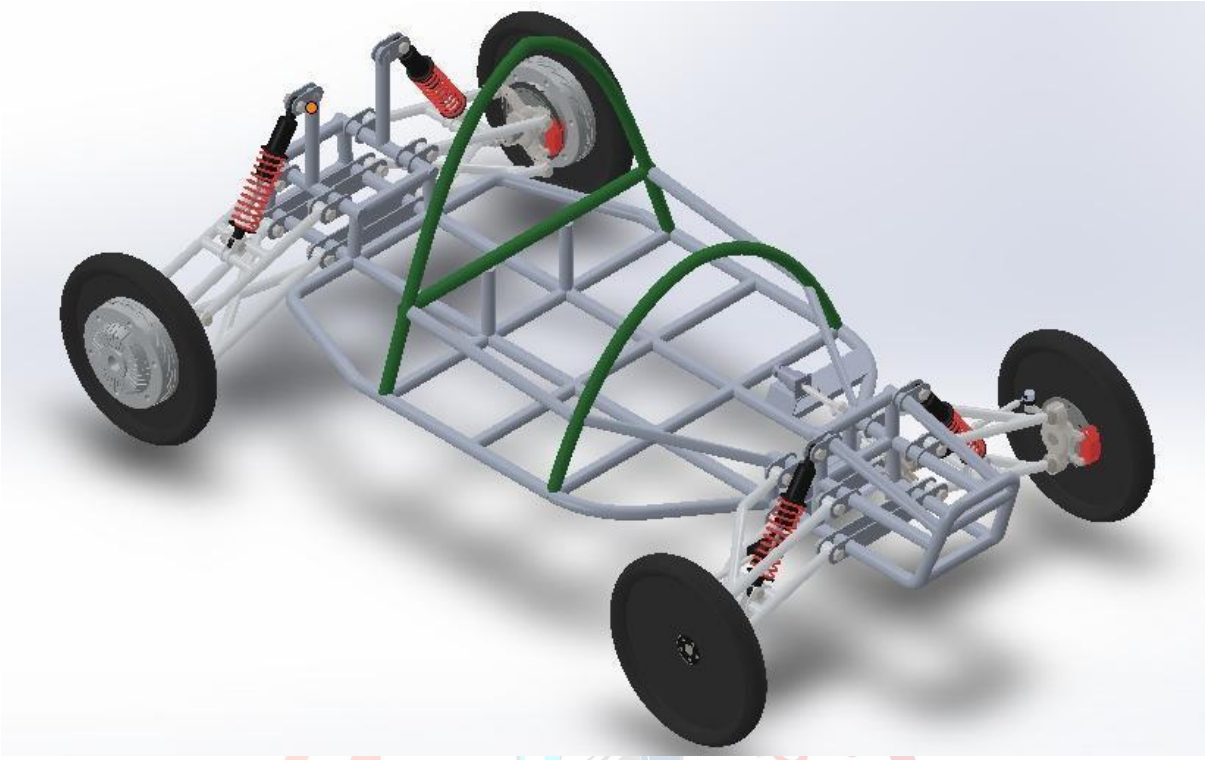
Kablosuz haberleşme konusunda 2 sistem kullanıyoruz. 1. si SIM800 Modülü üzerinden GSM aracılığıyla güvenli veri aktarımını gerçekleştirilecektir 2. sistemimiz ise RF haberleşmesi kullanılacaktır. İki sistemde de uçtan uca şifreleme kullanılacaktır. RF modülün yakaladığı sinyaller mikrodenetleyicide işlemden geçirilerek verinin bir sayısal/analog dönüştürücüye gönderilerek analog bir işarete dönüştürülür. Elde edilen analog veriler iki taraflı haberleşmede kullanılır.

Parçanın Adı	Ön Tasarım Raporundaki Fiyatı	Ön Tasarım Raporundaki Adedi	Kritik Tasarım Raporundaki Fiyatı	Kritik Tasarım Raporundaki Adedi	Oluşan Fiyat Farkı
URM06-UART Ultrasonik Sensörü	885,45 TL	3 Adet	1037,35 TL	3 Adet	455,70 TL
Sony IMX323 H.264 Kamera	1.049,72 TL	4 Adet	1.236,80 TL	4 Adet	748,32 TL
Omron E6B2-CWZ6C Rotary	814,20 TL	4 Adet	949,50 TL	4 Adet	541,20 TL

Encoder					
DRO TTL Lineer Cetvel	872,54 TL	3 Adet	938,14 TL	3 Adet	196,80 TL
DS18B20 Dijital Sıcaklık Sensörü	17,90 TL	8 Adet	25,61 TL	8 Adet	61,68 TL
700Watt 24Volt DC Motor	7.404,30 TL	2 Adet	7.794,00 TL	2 Adet	389,7 TL
120 Watt 24 Volt DC Motor	832,50 TL	2 Adet	982,35 TL	2 Adet	299,7 TL
ESP32 işlemci	108,61 TL	1 Adet	190,56 TL	1 Adet	81,95 TL
STMF103C8T6 İşlemci	95,63 TL	1 Adet	122,70 TL	1 Adet	27,07 TL
Motor Sürücü Half Dublex	71,98 TL	6 Adet	83,40 TL	6 Adet	11,42 TL
RF Haberleşme Modülü	17,43 TL	1 Adet	25,20 TL	1 Adet	7,77 TL
DC Motor Sürücü	780,35 TL	2 Adet	1.200,72 TL	2 Adet	840,74 TL
GSM	82,60 TL	1 Adet	146,80 TL	1 Adet	64,2 TL
12V 20A Akü	546,78 TL	2 Adet	639,40 TL	2 Adet	185,24 TL
				Toplam Fiyat Farkı =	3862,20 TL

Tablo 2: Fiyat Tablosu

3. Araç Mekanik Özellikleri



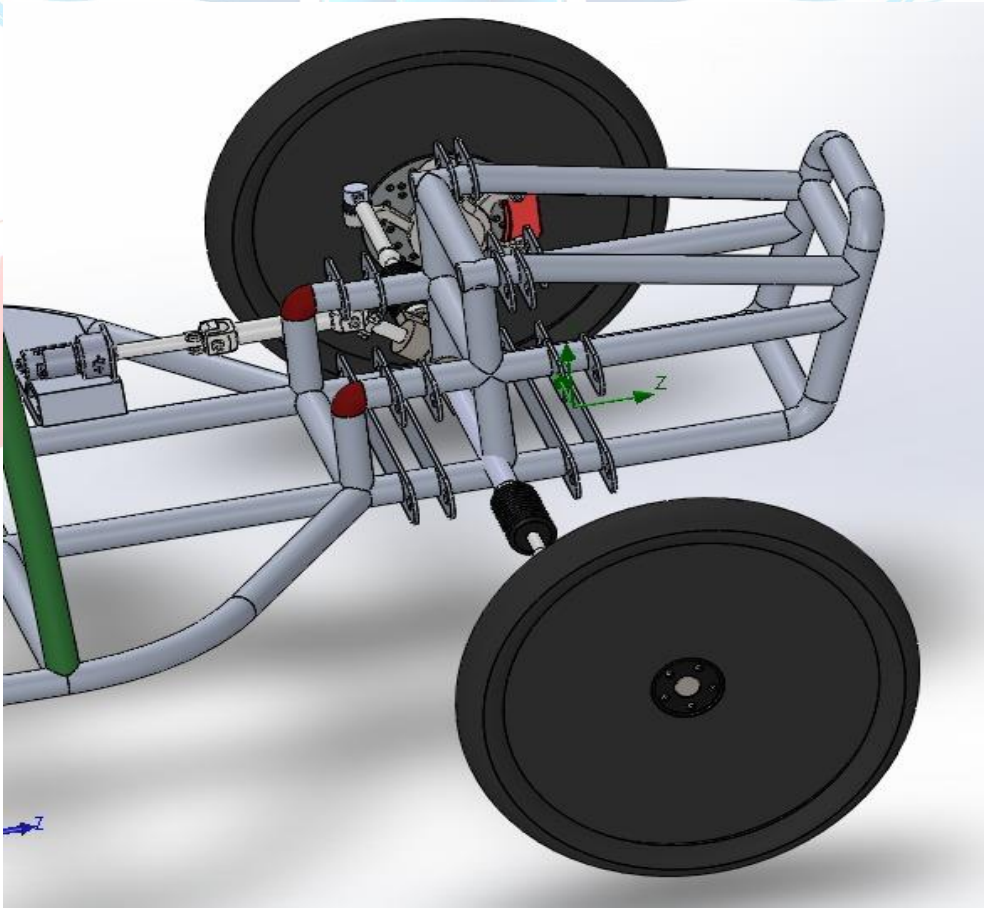
Şekil 3.1 Araç Kabuksuz Genel Tasarımı

Şekil 3.2 Aracımıza ait Sistem Ürün Kırılımı

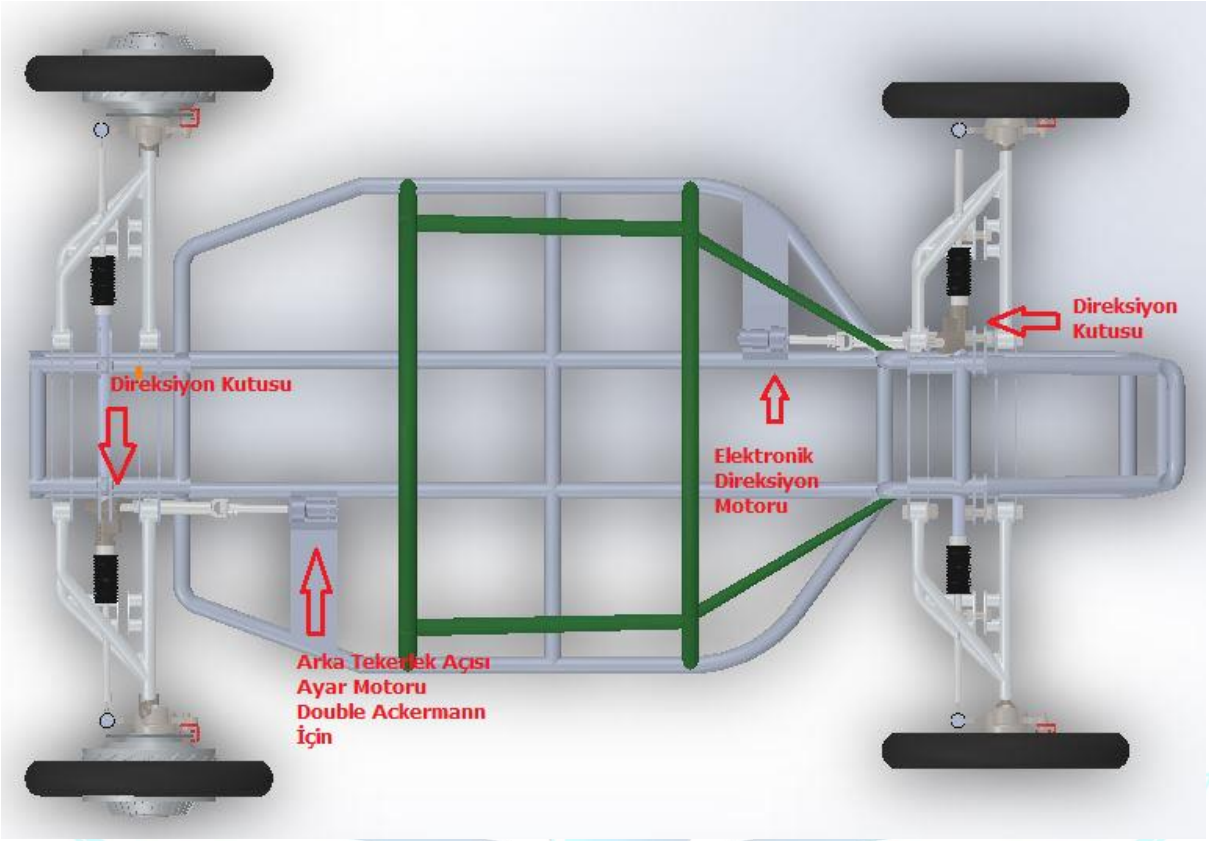
3.1 Elektronik Direksiyon Sistemi (steer-by-wire) : Aracımıza uygun direksiyon sistemi olarak bulunabilirlik ve ucuzluk açısından Lada Samara araçların direksiyon kutuları kullanılmaya karar verilmiştir. Bu kutulardan 2 adet kullanılacaktır. Bir tanesi klasik direksiyon sistemi olarak ön tekerlek açılarını, diğeri double Ackermann geometrisi kullanacağımızdan ötürü arka tekerleklerin açılarını değiştirmekte kullanılacaktır. Ackermann modeline (Şekil 5) uygun olarak monte edilecektir. Direksiyon milini döndürmek için 24V, 100W fırçalı DC motor kullanılacaktır. Motoru mile bağlamak için uygun ebatta kaplin kullanılacaktır. Direksiyon dönüş açısını ölçmek için endüstriyel lineer cetvel kullanılacaktır.

Daha dar açılı dönüşler yapabilmek için Double Ackermann Modeli (Şekil x) kullanımı planlanmıştır. Ön direksiyon gibi yazılım ile kontrol edebilmek için arka tekerleklere de bir adet direksiyon kutusu ve bu kutunun miline bağlı bir adet 24V, 100W fırçalı DC motor kullanılacaktır. Dönüş açısını ölçmek için endüstriyel lineer cetvel kullanılacaktır. Sistemin genel tasarımı Şekil x de görülmektedir.

Her iki sistemin de dönüş açıları Sürüş Kontrol Bilgisayarımız tarafından ayarlanacaktır. Yapı Otonom Sürüşe uygundur.

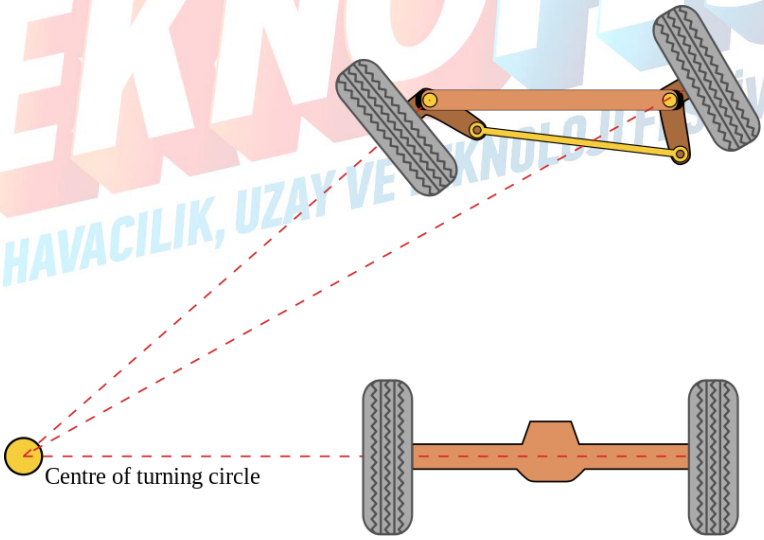


Şekil 3.1.1 Elektronik direksiyon sistemi

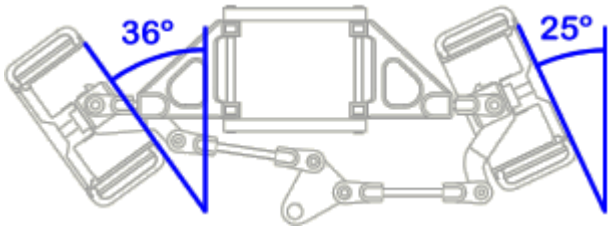


Şekil 3.1.2 Elektronik Direksiyon ve Arka Teker Açı Ayarlama Sistemi

Ackermann direksiyon geometrisi araç dönüş yaparken içteki ve dıştaki tekerlerin farklı açılarla dönmesini sağlayarak araç için kusursuz dönüş sistemini sağlamaktadır. Viraj alma ve engelden kaçma algoritmaları başlıklarında açılar ve etkiler detaylandırılmıştır.

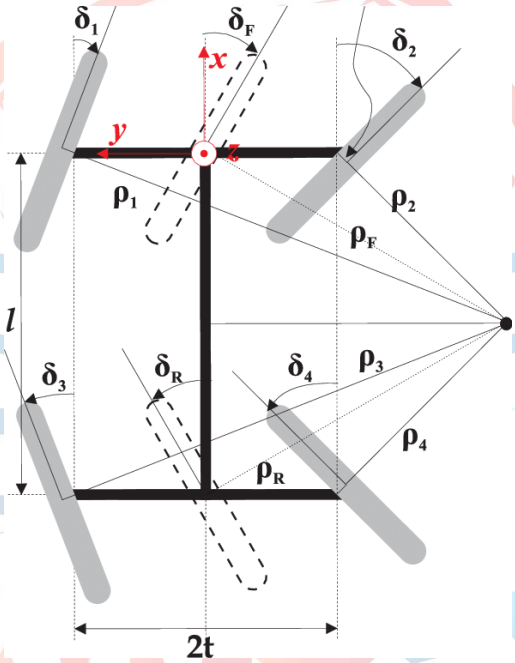


Şekil 3.1.3 Ackermann geometrisi



Şekil 3.1.4 Ackermann prensibinin ön tekerlere verebildiği açıların gösterimi.

Şekil 3.1.4'te sola dönen bu aracımızda viraj iç tarafında bulunan tekerin açısının daha fazla olduğu şekilde gözükmektedir, bu da bize daha keskin ve savrulma olmaksızın bir dönüş sağlamaktadır.



Şekil 3.1.5 Double Ackermann Prensibi

Ön tekerlekleri kapsayan Ackermann geometrisi hesaplarından(şekil 3.1.6) farklı olarak ön tekerleklere bağlı olan arka tekerleklerin çalışmalarını da etkilemektedir. Otonom aracımız arkadan itişli bir sisteme sahiptir.

$$x = \omega \times \left(\frac{\sin \delta_i - \sin \delta_0}{\sin \delta_i + \sin \delta_0}\right)$$

Şekil 3.1.6 Ön tekerlek Ackermann Prensibi hesaplaması

Arka itki motorlarına bağlı arka tekerlekler double Ackermann geometrisine uygun biçimde şaft ve ön tekerlek aksı üzerinden ön tekerleklerle bağlıdır. Bu yöntemde

uyarılacak tekerleklerin momenti ölçülerek aynen mekanik sistemdeki gibi ters orantılı bir güç dağıtımı, dolayısı ile hız oranının istenildiği gibi kararlı bir dönüş yapmayı sağlayacak şekilde ayarlanması sağlanabilir.

Double Ackermann'ı kullanmamızın sebebi virajları en etkin biçimde alabilmektir. Şerit takip ile viraja gelindiğinde eğimli çizgilerin orta noktası alınır. Araç şeritte kalmak şartı ile elektronik direksiyon sürüş kontrol bilgisayarı aracılığıyla otomatik olarak o noktaya yönlendiriliyor. Yapılan tekerlek açısı bu esnada oluşan tekerlek açısı lineer cetvelle ölçülerek aşağıdaki zımbırtı formüllerine konulup sağ ve sol itki motorlarının açısal hızları hesaplanıyor. Bu değerler sürüş kontrol bilgisayarına yollanarak motor hızları ayarlanır encoderlarla hız ölçümü yapılarak açısal hızların doğruluğu teyit edilir.

Araç viraja girdiği anda iç ve dış tekerleğin açısal hızlarının ne kadar değişim gösterdiğinin matematiksel olarak ifadesi elde edilmiş olur.

Belirli bir dönüş yarıçapı R, dingil mesafesi L ve iz genişliği T için gerekli ön direksiyon açıları (δ_(f,in) ve δ_(f,out)) aşağıdaki formülle hesaplanır:

$$\delta_{f,in} = \tan^{-1}\left(\frac{L}{R - \frac{T}{2}}\right) \quad \delta_{f,out} = \tan^{-1}\left(\frac{L}{R + \frac{T}{2}}\right)$$

Şekil 3.1.7 Double Ackermann prensibi hesaplaması

3.2 Elektronik Gaz Pedalı (pedal-by-wire) : Aracımız arkadan itişli olup 2 adet birbirinden bağımsız 24V, 700W DC motor ile hareket ettirilmektedir. Bu motorlar Şekil x de görülmektedir. Her motorun kendi motor sürücüsü vardır. Motor sürücülerin Sürüş Kontrol Bilgisayarından aldıkları PWM sinyallerine göre motor hızları ayarlanmaktadır. Bu yapı elektronik diferansiyel yapımına uygundur. Bu yüzden araç için bir elektronik diferansiyel de tasarlayacağız. Buna ek olarak sürücü ile kullanılmak istenmesi durumunda Sürüş Kontrol Bilgisayarına bağlanacak bir pedal ile de kontrol edilebilecektir. Manuel Kullanım İçin Elektronik Gaz Pedalı Şekil x'de görülmektedir. Yapı Otonom Sürüşe uygundur.

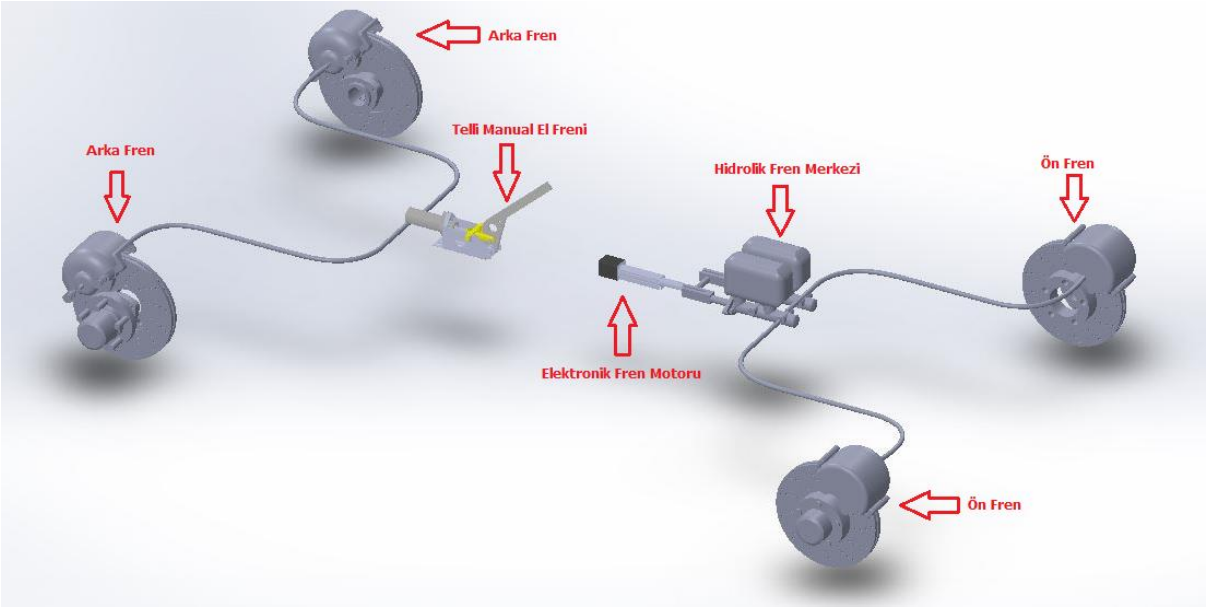
Şekil 3.2.1 24v 700W Ana İtki Motoru

Şekil 3.2.2 Manuel Kullanım İçin Elektronik Gaz Pedalı

3.3 Elektronik Fren Sistemi (brake-by-wire) : Aracımızda birbirinden bağımsız çalışan iki adet fren sistemi bulunmaktadır. Bunlardan birincisi ön tekerleklere takılı olan ve bir 24V, 100W fırçalı DC motor ile aktive edilen hidrolik fren sistemidir. Otonom sürüş esnasında kullanılacaktır. Emniyet için bir merkezli çift çıkışlı olarak seçilmiştir. Fren basma ve bırakma oranlarını ölçebilmek için bir adet endüstriyel lineer cetvel monte edilmiştir. Sürüş Kontrol Bilgisayarımız tarafından ayarlanacaktır. İstendiği takdirde Şekil x Manuel görülen pedaldan bir tane daha Sürüş Kontrol Bilgisayarına takılarak manuel sürüş fren işlevi yerine getirilebilir. Elektromekanik fren sistemi Şekil x'da görülmektedir.

Diğer fren sistemi ise aracın park halinde kullanacağımız arka tekerleklere takılı klasik telli el frenidir. Şekil 3.3 km.1'de görülmektedir.

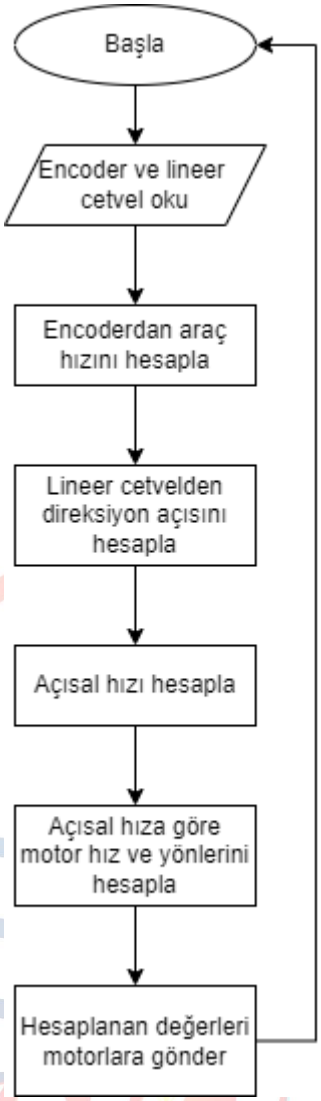
Yapı Otonom Sürüşe uygundur.



Şekil 3.3 Elektromekanik Ön ve Mekanik Arka Fren Sistemi

3.4 Elektroniksel Diferansiyel Sistemi : Algoritmada da belirtildiği gibi aracın tekerleğine yerleştirilen encoder aracılığıyla tekerleğin dönüş sayısını hesaplayarak elde edilen hız verileri ile direksiyon açısı değerlerini araç kontrol sisteminde işlenip bu değerlerin motor sürücüye iletilmesi gerçekleştirilmiştir. Viraj alma ve Engelden kaçma algoritmaları da diferansiyel sisteme bağlı olarak çalışacaktır.

TEKNOFEST
HAVACILIK, UZAY VE TEKNOLOJİ FESTİVALİ



Şema 3.4 Elektroniksel diferansiyel sistemi algoritması

3.4.1 Viraj Alma Algoritması : Aracımız üzerinde bulunan çizgi izleme kamerasından gelen verileri kullanarak hata oranını hesapladıktan sonra hata oranı (okunan çizginin olması gereken noktadan uzaklığı) küçülmekte ise çizginin sağa kaydığı (sağa doğru bir viraj olduğu) büyümekte ise çizginin sola kaydığı (sola doğru viraj olduğu) tespit edilmekte ve bu oranın hedefe olan uzaklığıyla direksiyon değeri oranlanarak direksiyonun ne kadar döneceği belirlenmektedir. Eğer sensörden gelen değer hedef değerle aynı ise araç direksiyonu yeniden ortalar ve yoluna normal bir şekilde devam eder.

3.4.2 Engelden Kaçma Algoritması : Aracın önüne bir engel çıktığında takip edeceği basamaklar:

Görüntü işleme ve şerit takip sisteminden doğrultu, yön, ve yolun açı verilerini alma

Önündeki engelle olan mesafesini ultrasonik sensör yahut lidar ile ölçme ve ekrana yazdırma

Tekerleklerde bulunan encoder aracılığıyla aracın seyir halinde yaptığı hız verisini alma

Bu verileri sensör füzyona sokup çıktıları motorlara aktarma

Double Ackermann geometrisi ile arka itki motorlarına bağlı tekerlekleri ön tekerleklerle koordineli olarak hareket ettirme

Trafiğin Türkiye gibi soldan aktığı ülkelerde "sollama" adı verilen hareketi gerçekleştirme.

Engeli aştıktan sonra şerit takibi ile belirlenen güzergahta ilerleme.

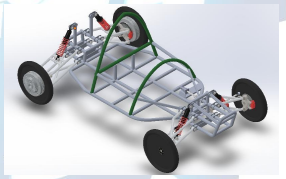
4. Donanım Mimarisi

4.1 Sensörler

Tablo 4.1.1 Kullanılan malzemeler ve özellikleri

Malzeme Özellikleri	Adet	Seçilme Nedeni	Resim
Direksiyon Motoru 24V 380rpm. 75 kg-cm torka sahiptir. Boşta çektiği akım 800 mA dir. Zorlanma anında 1,75 Amper akım çekmektedir. Motorun baştan sona uzunluğu 17cm'dir. Motor mili 10mm kalınlığında 14mm	2	Güçlü ve hızlı oluşu, ayrıca kendinden kilitlemeli olduğundan ön ve arka tekerleklerin açılarını kontrol eden direksiyon motoru olarak seçilmiştir.	

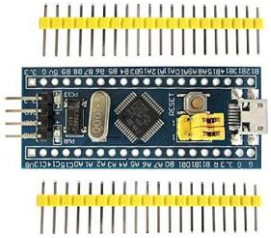
Malzeme Özellikleri	Adet	Seçilme Nedeni	Resim
uzunluğundadır.			
Ana İtki Motoru 24V, 700 W	2	Ana itki motoru olarak kullanılmak üzere seçilmiştir. Redüktörü sonsuz vida mil olduğu için yüksek torklu olması ve kendiliğinden frenli olmasından dolayı seçilmiştir. Aracımız arkadan itişli olduğu için arka tekerleklerde bulunmak üzere 2 tane kullanılmıştır.	
Encoder Performans: 400 itki/devir çalışma voltajı: DC5-24V Maksimum mekanik hızı: 5000 dev/dk Elektriksel yanıt frekansı: 20K/sn Entegre hızı: 2000 dev/dk	4	Araçta dört tekerleğe takılmış dört adet Omron E6B2-CWZ6C 1000 P/R Rpm ABZ Phase 5-24V 1000 Pals Optik Döner Encoder kullanmayı planlıyoruz. Bu encoderler ile tekerlek dönüş hızlarını hassas bir biçimde ölçmeyi ve en uygun dönüş algoritmasını buna göre optimize etmeyi hedefliyoruz.Araç üzerinde her tekerlekte bir adet bulunmaktadır. Tekerlek hızları ayrı ayrı ölçülebileceğinden elektronik diferansiyel yapımında da kullanılacaktır.	
MONSTER Tulpar T7 V22.2 17.3” *Intel® Tiger Lake Core™ i7-11800H 8C/16T; 24MB L3;	1	Görüntü işleme bilgisayarı olarak kullanılacaktır. Nvidia RTX 3060 ekran kartında bulunan 3584 adet NVIDIA CUDA Çekirdekleri sayesinde görüntü işleme	

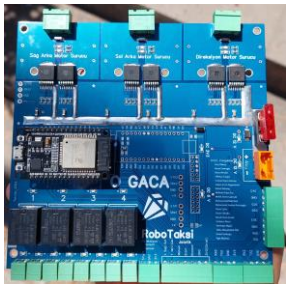
Malzeme Özellikleri	Adet	Seçilme Nedeni	Resim
8GT/s; 2.3GHz > *Nvidia RTX3060 6GB GDDR6 192-Bit *32GB DDR4		performansı muadillerine kıyasla yüksek olduğu için seçilmiştir. Araçta bir adet kullanılacaktır.	
GPS Çalışma gerilimi : 3.3V-5V Haberleşme Birimi: UART (RX-TX) 25x25mm Seramik Anten Default Baud Rate: 9600	1	Aracın konumunu hareketle eş zamanlı olarak bilmek için kullanılıyor.	
Alüminyum Şase 6070	1	Aracın ölü ağırlığını hafifletebilmek için alüminyum kullandık. Bu tercih araç gövdesinin paslanmasının önüne geçecektir. Daha hafif gövde daha uzun akü ömrü ve daha uzun çalışma süresini sağlar.	
Amortisör Havalı/gazlı 100nm	4	Aracın zorlu arazide giderken sarsılmasını önlemek için birbirinden bağımsız dört adet amortisör kullanılmıştır. Daha az sarsılan aracın görüntü işleme ve ilaçlama performansı artacaktır.	

TEKNOFEST HAVACILIK UZAY VE TEKNOLOJİ FESTİVALİ

Malzeme Özellikleri	Adet	Seçilme Nedeni	Resim
Lastik Tekerlek Havalı Lastik ANLAS MG 1.05X16	4	Yarışma koşulları gereğince bize verilen standart lastikleri kullanıyoruz.	
Akü 12V Kurşun Akü	2	Bakımı kolay ve daha ucuz olduğundan klasik kurşun akü kullanılacaktır. Bu aküler Li-Po vb. pillere göre aşırı ısınma, aşırı şarj/deşarj, darbe dayanımı gibi konularda daha güvenlidir. Daha ucuzdur. Aracımızın performans özelliklerin karşılamaktadır.	
Lidar 5.5hz (2000 devir/ sn) 6 metre mesafe aralığı 360 derece dönüş açısı	1	Aracın karşılaşabileceği engelleri tespit edip güvenli rotası bulmada kullanılacaktır.	
Sıcaklık Sensörü DS18B20 Kablolu Tepki Süresi:750 mS'den daha az Sıcaklık Ölçüm Aralığı: -55°C – +125°C	5	Batarya paketinde bir adet, her bir motor için bir adetten iki adet ve Ana Motor sürücülere iki adet olmak üzere toplam 5 adet DS18B20 dijital sıcaklık sensörü bulunacaktır. Bu sensörler sayesinde aşırı ısınan yerleri tespit edip tehlike oluşmadan önlemeyi planlıyoruz.	

Malzeme Özellikleri	Adet	Seçilme Nedeni	Resim
Acil Durum Devre Kesici Çift Kontaklı	1	Acil durumlarda aracın tüm elektrik enerjisini elle müdahale ile kesmede kullanılacaktır.	
Sigorta B Tipi 10A Sigorta	2	Olası elektrik kısa devrelerinde atarak aracın elektriğini kesip emniyete alması için uygun akımda seçilmiştir.	
RF Haberleşme Modülü RFD868 Haberleşme Mesafesi 40 Km Frekans 868Mhz	1	Araç ile yönetim bilgisayarı arasında haberleşme yapmak için kullanılacaktır. Yarışmada kullanımı için uygundur.	
ESP 32 Çift Çekirdekli 520 KB SRAM 448 KB ROM. Kablosuz bağlantı: Bluetooth, v4.2 BR/EDR ve BLE. Wİ-Fİ, 802.11 b/g/n 2 adet I2S ara yüzleri. 2 adet I2C arayüzleri. 3 adet UART	1	Sürüş bilgisayarı ana işlemcisi olarak kullanılacaktır. Çift çekirdekli oluşu, yüksek performansı ve yerleşik wifi ve bluetooth barındırması tercih nedenimizdir. Daha önce kullandığımız bir işlemcidir.	

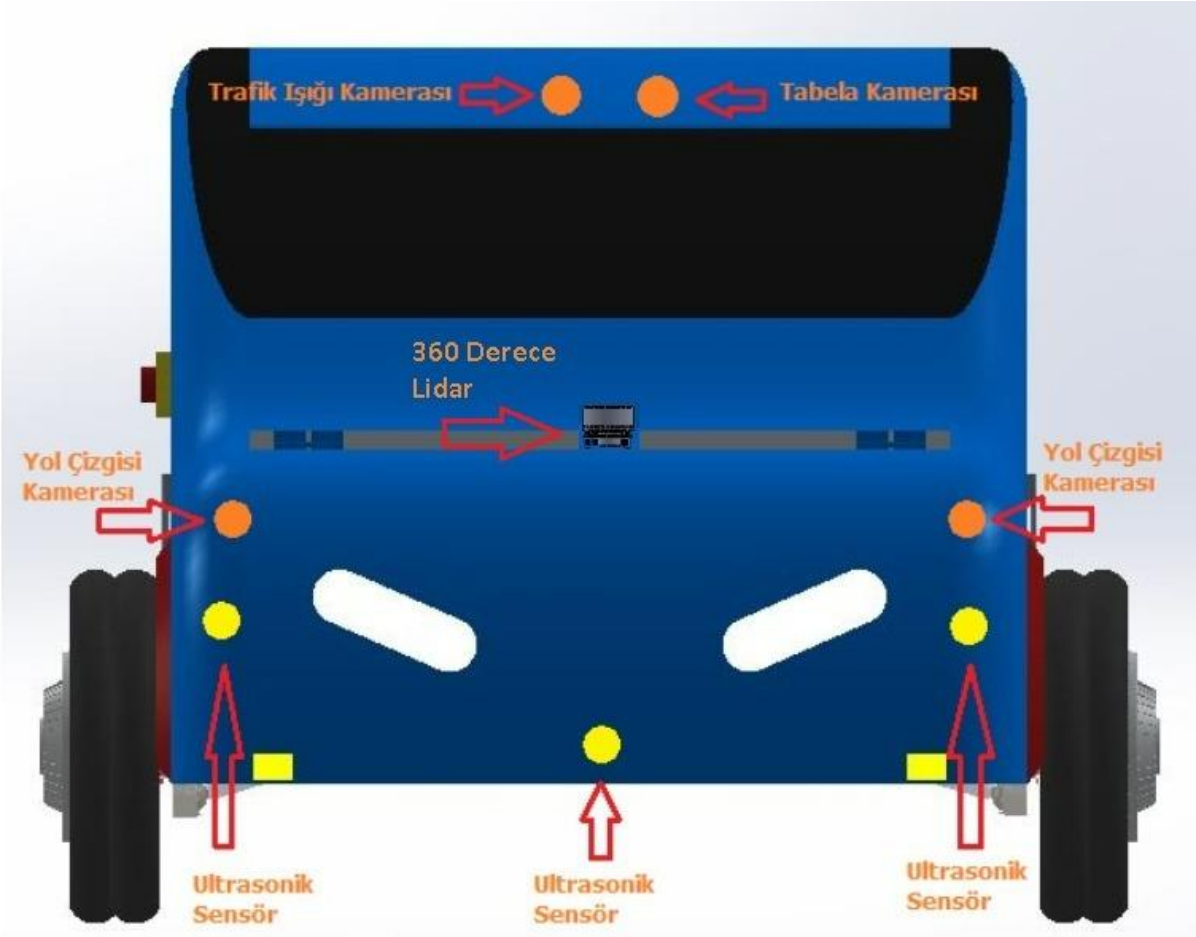
Malzeme Özellikleri	Adet	Seçilme Nedeni	Resim
STMF103C8T6 Pin sayısı 48 İşlemci Cortex-M3 Çalışma Frekansı 72MHz Depolama Kaynakları 64K Byte Flash, 20KByte SRAM Arayüz Kaynakları 2x SPI, 3x USART, 2x I2C, 1x CAN, 37x I / O ports Analog-digital çevirim 2x ADC (12-bit / 16-kanal)	1	Sürüş Bilgisayarı yardımcı işlemcisi olarak kullanılacaktır. Yüksek performansı ve elektriksel parazitlere olan dayanımı tercih nedenimizdir. Daha önce kullandığımız bir işlemcidir.	
Direksiyon Motor Sürücü Half Dublex 40 amper	6	BTS7960B motor sürücü yüksek güçlü ve yüksek akım verebilen sürücü kartıdır. 40 ampere kadar akım çeken motorların kontrolünde kullanılabilir. Kendi tasarladığımız kontrol devresinde Half Bridge olan bu sürücü entegreden 6 adet kullanılacaktır. böylece toplamda 3 adet Full Bridge DC motor sürücümüz olmuştur.	

Malzeme Özellikleri	Adet	Seçilme Nedeni	Resim
Sürüş Kontrol Bilgisayarı Anakartı	1	Kendi tasarladığımız Sürüş Kontrol Kartımız çift katmanlı olup üzerinde 6 adet BTS7960 motor sürücü bulunmaktadır. Bunu ile 3 adet motor sürüş imkanı sağlanmaktadır. üzerinde bulunan 4 Adet röle sayesinde sinyal lambalarını kontrol etmekteyiz. Koruma olarak çıkartıp takılabilen 5 amperlik bıçak sigorta ve ters akıma karşı 1N4007 kullandık. Kartın alt kısmında bulunan mp2307 voltaj regülatörü sayesinde kartın 3.3V ve 5V ihtiyacı karşılanmaktadır. Veri aktarımında oluşacak parazitlere karşı kondansatör ile filtre devresi kullandık. Motorların aralarında oluşabilecek istenmeyen yüksek akımlarda iki motor arasına TVS diyot koyarak koruma sağladık. Üzerinde kontrolcü bilgisayarı olarak ESP32 kullandık yardım etmesi için STM32f103c8t6 kullandık.	
Ultrasonik Mesafe Sensörü URM06-UART Çalışma voltajı: 6V-12V (5V kabul	3	Araçta şekil 4.1.1te görüldüğü gibi üç adet önde URM06-UART ultrasonik sensör bulunmaktadır. şekil 4.1.1'de belirtildiği gibi 10 metreye kadar etkili olan bu sensörle	

Malzeme Özellikleri	Adet	Seçilme Nedeni	Resim
edilebilir ancak önerilmez) Nominal Akım: 16mA Maksimum akım: 2A Çalışma frekansı: 49.5KHZ 15°(-6dB) Mesafe Tespiti: 20cm ~ 10m(7.87"~393.70")		araç önündeki 10mX3m=30m2 lik bir alanı gözlemleyip tehlike içeren durumlarda aracı ani fren ile durdurulur. Can ve mal güvenliğini gözetmek amaçlı kullanılacaktır.	
Kamera Sony IMX323 Kamera	4	4 adet önde 2MP 1080P Sony IMX323 H.264 düşük ışık 0.01Lux endüstriyel makine görüş Mini Usb kamerası bulunmaktadır. Bu kameralar ile aracın önünde 3m X 10m X 4m=120m3 lük bir alandan görüntü alıp işlemeyi planlıyoruz.	
GSM Modülü Sim800L GSM/GPS	1	Kullanımlarında çeşitli nedenlerden dolayı araç ile Yönetim Bilgisayarı arasında uzun mesafe veri aktarımı için alternatif bir bağlantı olması için seçilmiştir. GSM sinyalinin olduğu her yerden haberleşme yapabilecektir.	

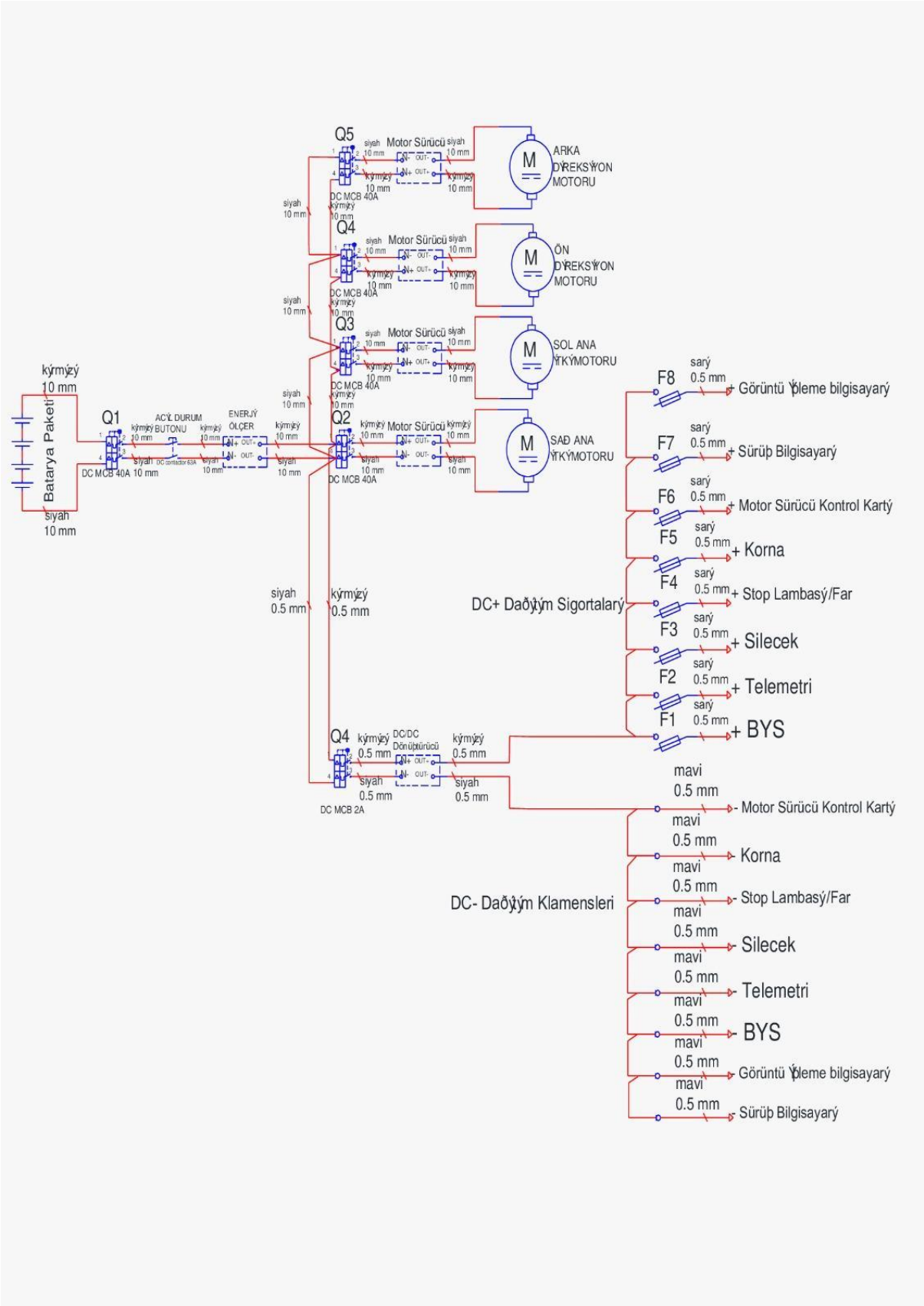
Malzeme Özellikleri	Adet	Seçilme Nedeni	Resim
Lineer Cetvel Giriş voltajı: 5VDC Çıkış sinyali: TTL Ölçek:50mm, 100mm,150mm, 200mm,250mm, 300mm,350mm, 400mm,450mm (özelleştirilebilir) Ölçek toplam uzunluğu: + 138mm Genişlik * yükseklik: 21mm*33mm (20*29mm ve 26*42mm	2	Double Ackermann geometrisinde direksiyonun dönüş açısını hesaplamak için kullanılır. Hassas ölçüm yapabilmek için tercih edilmiştir.	
Ana İtki Motor Sürücüsü	2	Güçlü (700 W), sonsuz vida dişli sistemiyle kendinden frenli, ayriyeten kendi üzerinde elektronik fren olması, kararlı ve torklu bir sürüşünün olduğu ana itki motorumuzla uyumlu olduğu için bu özel yapım motor sürücüyü seçtik.	

TEKNOFEST
HAVACILIK, UZAY VE TEKNOLOJİ FESTİVALİ

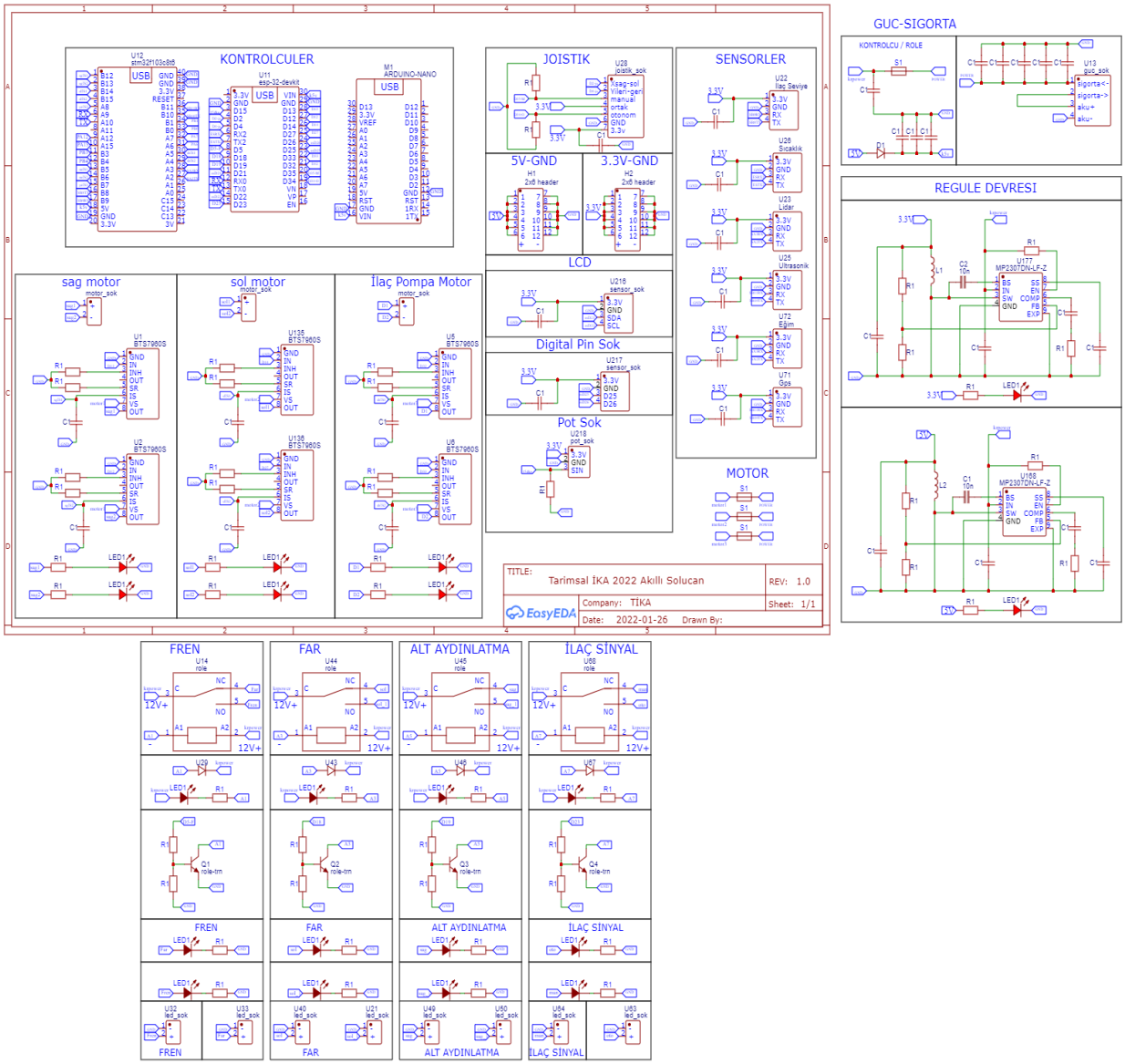


Şekil 4.1.1 Aracın Önden Görünüşü ve Temel Sensör Yerleri

TEKNOFEST
HAVACILIK, UZAY VE TEKNOLOJİ FESTİVALİ

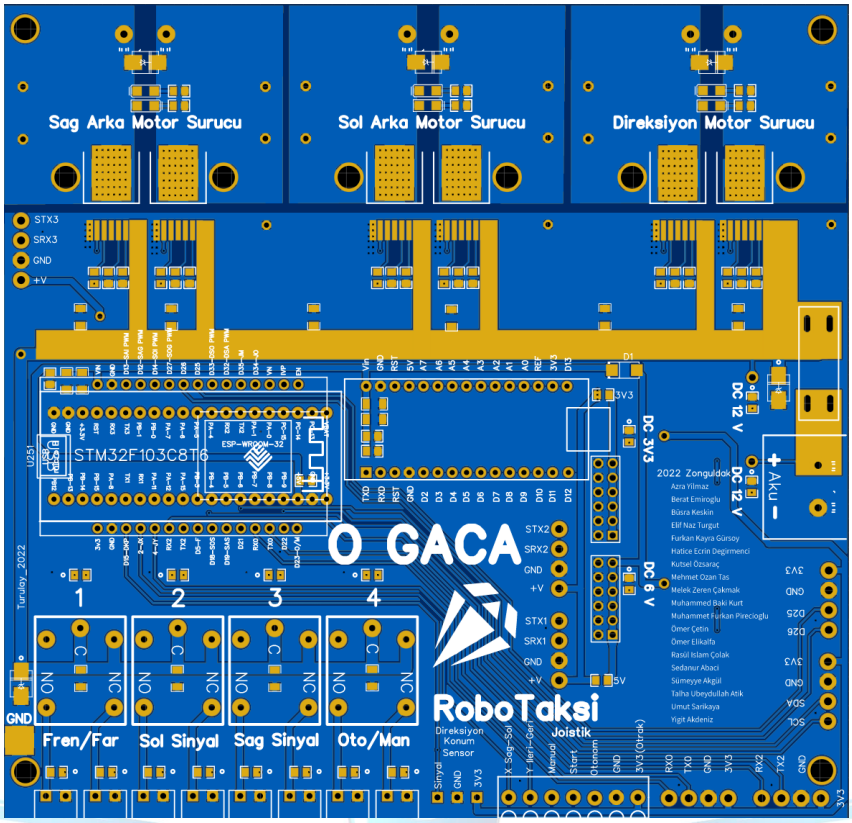


Şekil 4.1.2 Elektrik Donanım Şeması

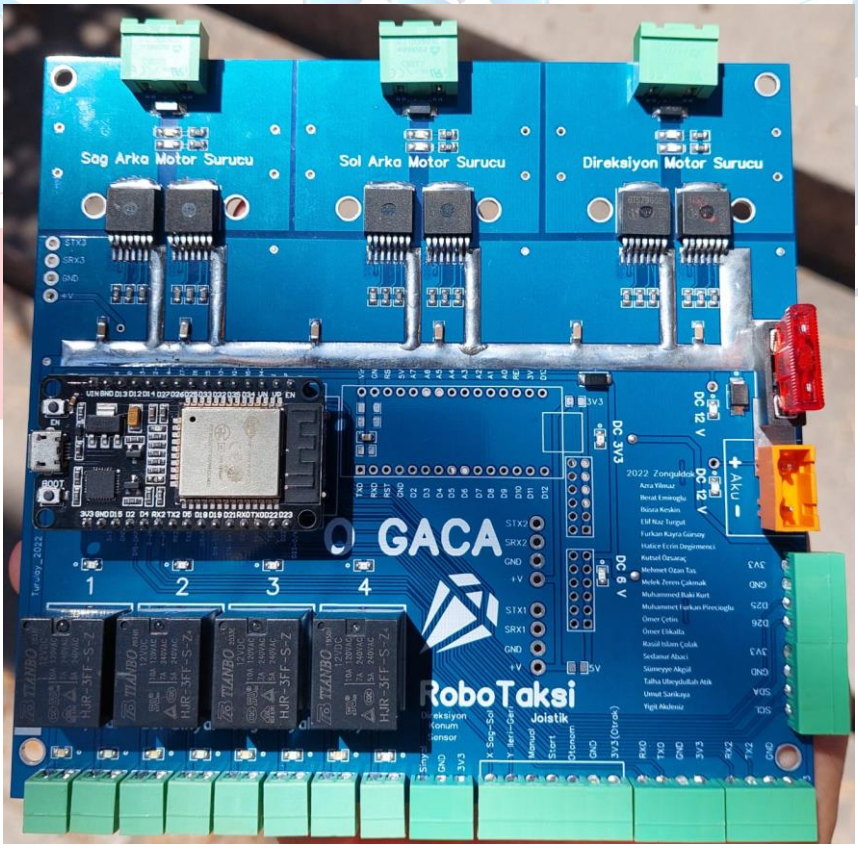


Şekil 4.1.3 Araç Ana kontrol kartı üzerindeki mikrodenetleyici ve bağlantıların gösterimi

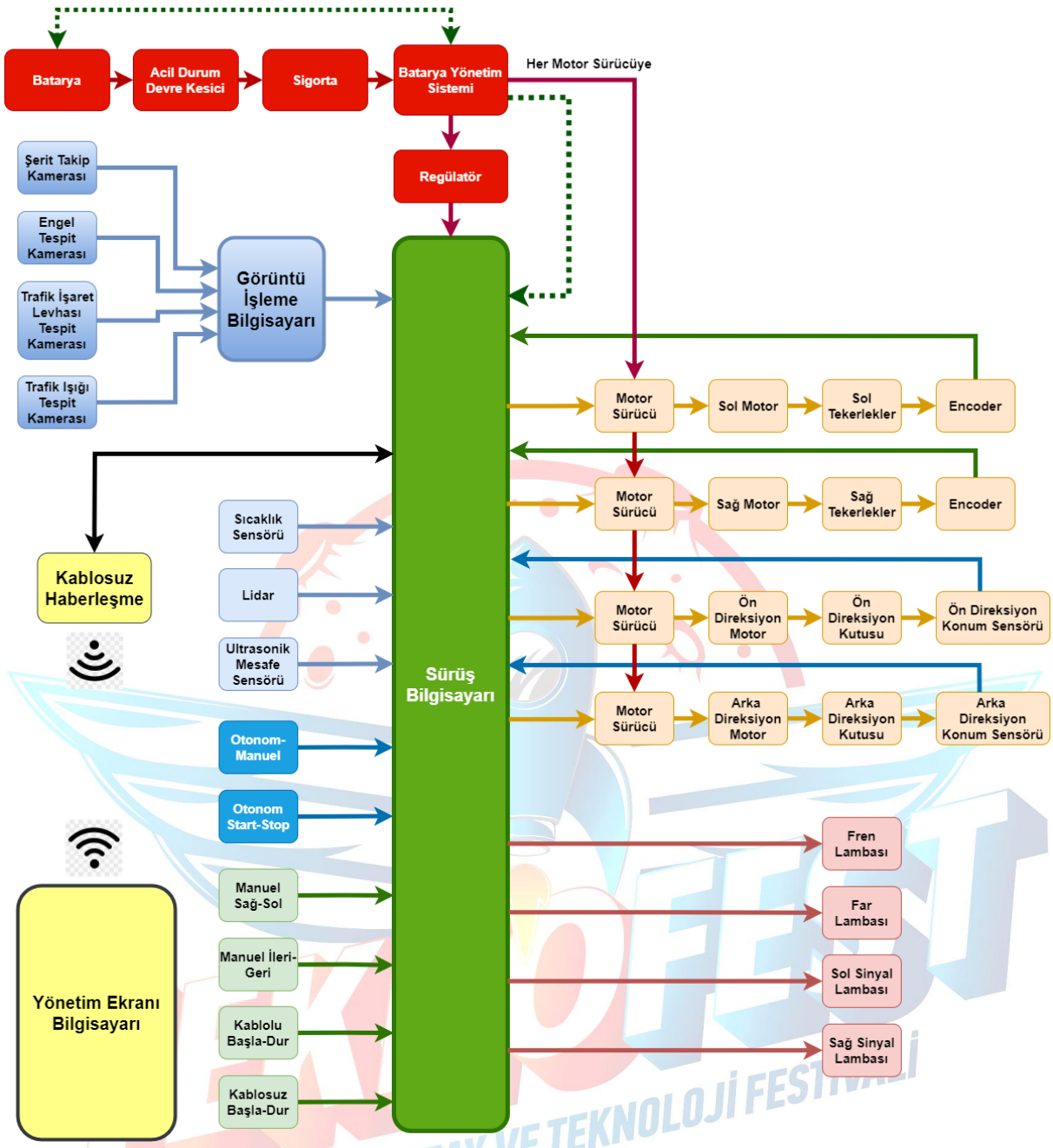
HAVACILIK, UZAY VE TEKNOLOJİ FESTİVALİ



Şekil 4.1.4 PCB Tasarımın 2D Görüntüsü



Şekil 4.1.5 PCB Tasarımın Montajlanmış Hali



Şema 4.1.6 Kullanılan tüm sensörler ve sistemlerin bağlantılarını gösteren Harness diyagramı

4.2 Güvenlik Önlemleri

1-Yarışma alanında bir kaza oluşması ihtimaline karşın araç bileşenleri araca sıkıca sabitlenmiştir.

2-Aracın önüne bir engel çıkması riskine karşı otonom aracımızın önüne konumlanmış 4 tane kamera bulunur. Bunlar: 1.Trafik ışığı algılama kamerası, 2. Tabela okuma kamerası, aracın iki tarafına konumlanmış 3. ve 4. kameralarsa şerit takip kameralarıdır. yukarıdaki *Şekil Aracın Önden Görünüşü ve Temel Sensör Yerleri"* nde gösterilmiştir)

4-Tasarlanan otonom araç, 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.

5-Kullandığımız havasız lastik sayesinde lastiğin patlama şansı ortadan kalkacaktır Tercih ettiğimiz lastikler sahip oldukları dayanıklı yapı sayesinde özellikle lastiğin çok esnediği kenar, köşe ve kayalık şartlarda üstün tutunma kabiliyeti göstererek aracın ters dönme ihtimalini de ortadan kaldıracaktır.

6-Tüm bunlara ek olarak elektronik parçalarda oluşabilecek herhangi bir kısa devre durumunda gücün kesilebilmesi için bataryanın "+" ucu ve devre arasına sistemin talep edebileceği maksimum akımdan %10 fazla akım geçirme değerine sahip bir sigorta yerleştirilecektir.

7- Herhangi bir çarpışmaya karşın lidar ve ultrasonik sensörler birlikte kullanılmaktadır. Engel tespit edildiğinde çarpmayı engellemek için tüm motorlar durdurulacaktır.

8- Kullanılan kablolar çekilecek maksimum akımın 3 katı kalınlığında yanmaz kablo olarak seçilmiştir.

9- Tüm kablolar orta destekli dayanıklı kablo kanalı içinden geçirilmiştir. Açıkta kablo bulunmamaktadır.

10- Aşırı ısınmaya karşı ısınması muhtemel motor ve batarya sıcaklıkları ısı sensörleri ile ölçülmektedir. Aşırı ısınma tespit edildiğinde sistem tüm motorları kapatıp yönetim bilgisayarına alarm verecektir.

5. Yazılım Mimarisi

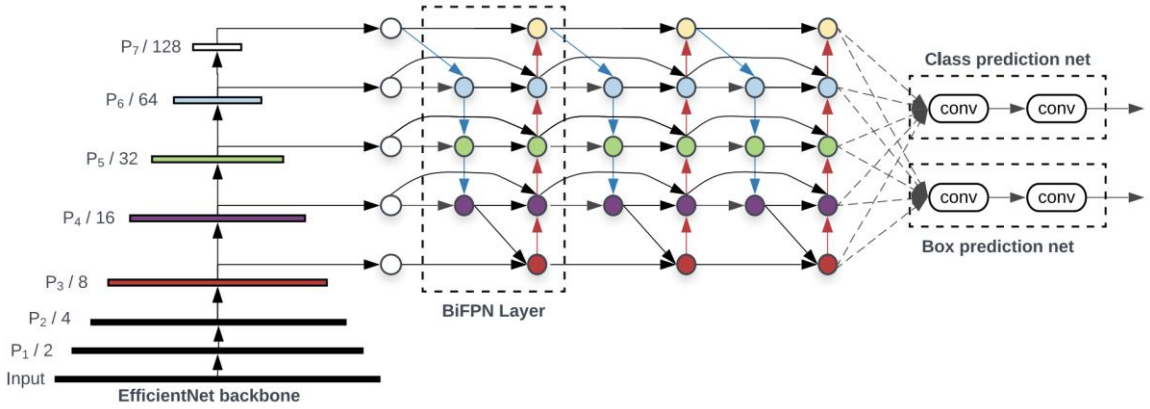
5.1 Görüntü İşleme ve Otonom Sürüş Algoritmaları

5.1.1 Görüntü İşleme Algoritmaları

5.1.1.1 YOLOv4 (You Only Look Once)

YOLO modeli, uçtan uca türevlenebilir ağda sınıf etiketleriyle sınırlayıcı kutuları tahmin etme prosedürünü birleştiren ilk nesne dedektörüdür. Her bir ana bileşende farklı mimarileri birleştirmek için alınabilecek birçok yaklaşım vardır. YOLOv4 ve YOLOv4 Tiny'nin katkıları, en başta bilgisayar görüşünün diğer alanlarındaki buluşları entegre etmede ve bir koleksiyon olarak YOLO nesne algılamasını iyileştirdiklerini kanıtlamaktadır. R-CNN gibi diğer birçok nesne tespit algoritması ilk başta görüntü üzerinde bölge öneri metodları kullanarak aranan nesnenin bulunma olasılığı yüksek yerleri tespit eder. Daha sonra bu bölgelerde nesne olup olmadığını kontrol etmek için Destek Vektör Makineleri (Support Vector Machines - SVM) gibi sınıflandırma algoritmalarını kullanır. Sınıflandırmadan sonra görüntü, nesnenin görüntü üzerindeki sınırlarının

belirlenmesi, aynı nesnenin birden fazla tespiti olduysa bunların kaldırılması ve nesne tespit başarı puanının yeniden hesaplanması için tekrar işlenir. Bu karmaşık metodlar beraber çalıştıklarında çok yavaş ve hızlandırılması zor bir hal almaktadır. Bunun sebebi her adımdaki görevleri yapan modelin ayrıca eğitilmesinin gerekmesidir. YOLO algoritmasının farkı sırayla görüntü üzerindeki farklı bölgeler için tahmin modeli çalıştırmak yerine görüntüye bir bütün olarak bakıp sadece bir defada görüntüyü işlemesidir. Biz işaret tabelaları üzerinde görüntü işlemek için YOLOv4'ün C++ kütüphanesini kullandık.



Şekil 1.1.1.1 YOLOV4 katmanları

5.1.1.2 YOLOv4 Tiny (You Only Look Once)

Kullandığımız YOLOv4 Tiny bir nesne dedektörüdür. Girdi görüntülerinden feature mapping oluşturmak ardından bu özellikleri, nesnelerin etrafına kutular çizmek ve sınıflarını tahmin etmek için tasarlanmıştır. YOLOv4 Tiny gerçek zamanlı bir nesne algılama modelidir.

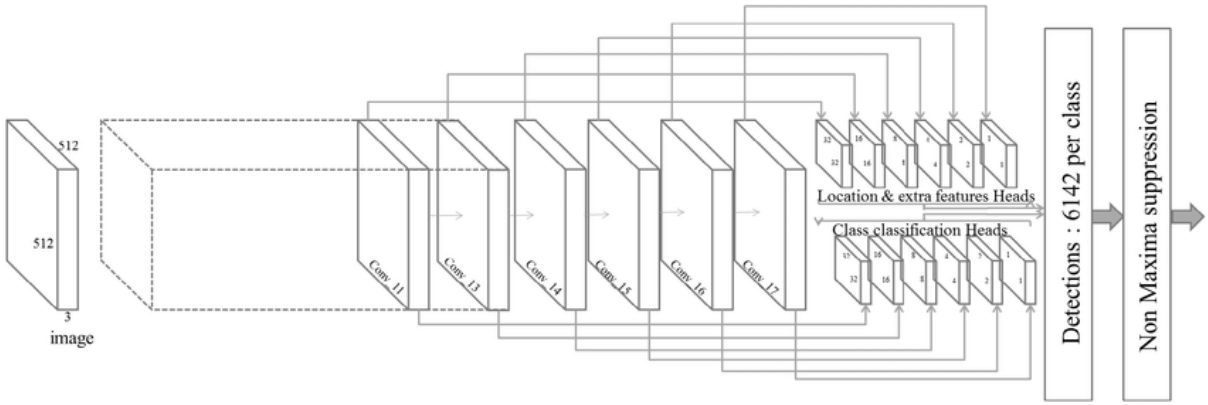
Bu görevde kullanılan YOLOv4 Tiny algoritması görüntüyü adından da anlaşılacağı üzere yalnızca bir kez bir evrişimsel sinir ağı modelinden geçirmektedir. Bu şekilde saniyede işlenen kare sayısı büyük ölçüde artmakta ve gerçek zamanlı nesne tespitine olanak sağlanmaktadır.

Nesne tespiti için geliştirilmiş algoritmalardan hangisinin kullanılması gerektiği görevin amacına göre değişiklikler gösterebilir. Bize trafik ışıklarını tanırken neredeyse gerçek zamanlı bir algoritmaya ihtiyaç olduğundan hızlı olmasına rağmen tespit başarısı diğerleriyle yarışır düzeyde bir algoritma olan YOLOv4 Tiny algoritmasından faydalanılabilir. YOLOv4 Tiny'den tabela veri setimizin dağılımını her jenerasyondan sonra kabaca test ederken yararlandık.

5.1.2 SSD MobileNet

MobileNet, derin sinir ağları oluşturmak için derinlemesine ayrılabilir evrişimler kullanan modern bir mimariye dayanmaktadır.

SSD mimarisi, sınırlayıcı kutu konumlarını tahmin etmeyi ve bu konumları tek geçişte sınıflandırmayı öğrenen tek bir evrişim ağıdır. Bu nedenle, SSD uçtan uca eğitilebilir. SSD ağı, temel mimariden (bu durumda MobileNet) ve ardından birkaç evrişim katmanından oluşur.

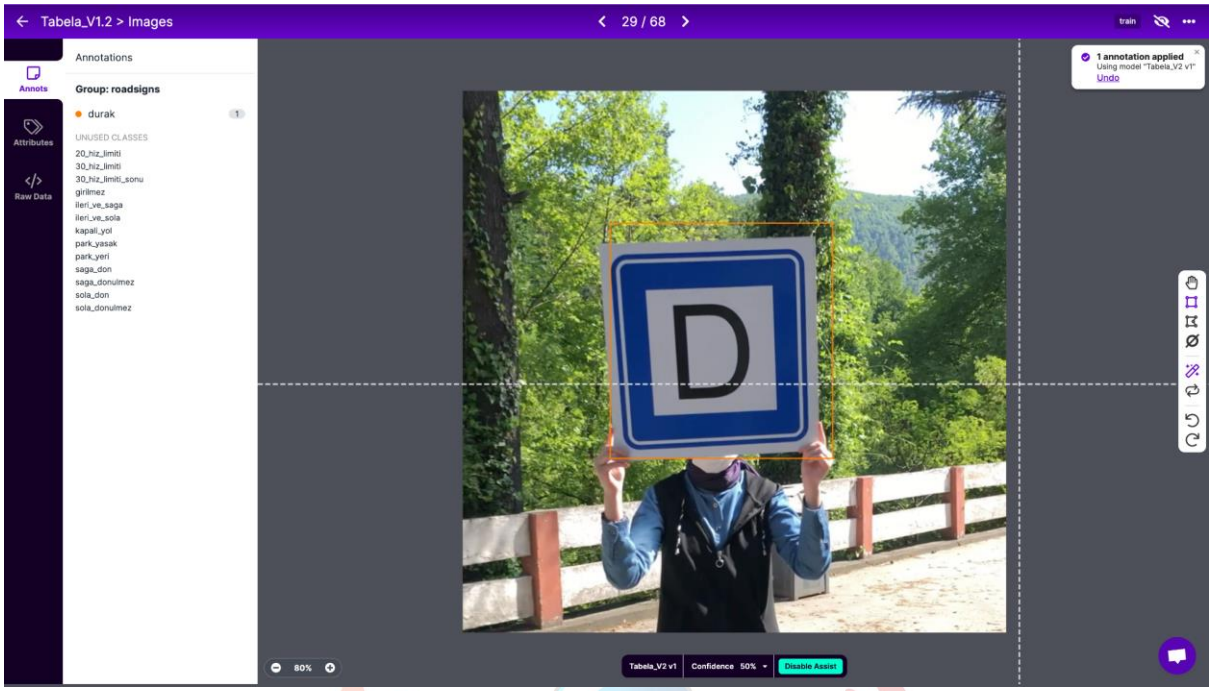


Şema 5.1.1.2 SSD MobileNet sinir ağı şeması

Trafik ışıklarının koordinatlarını ve renklerini bulmak için bu derin sinir ağını kullandık.

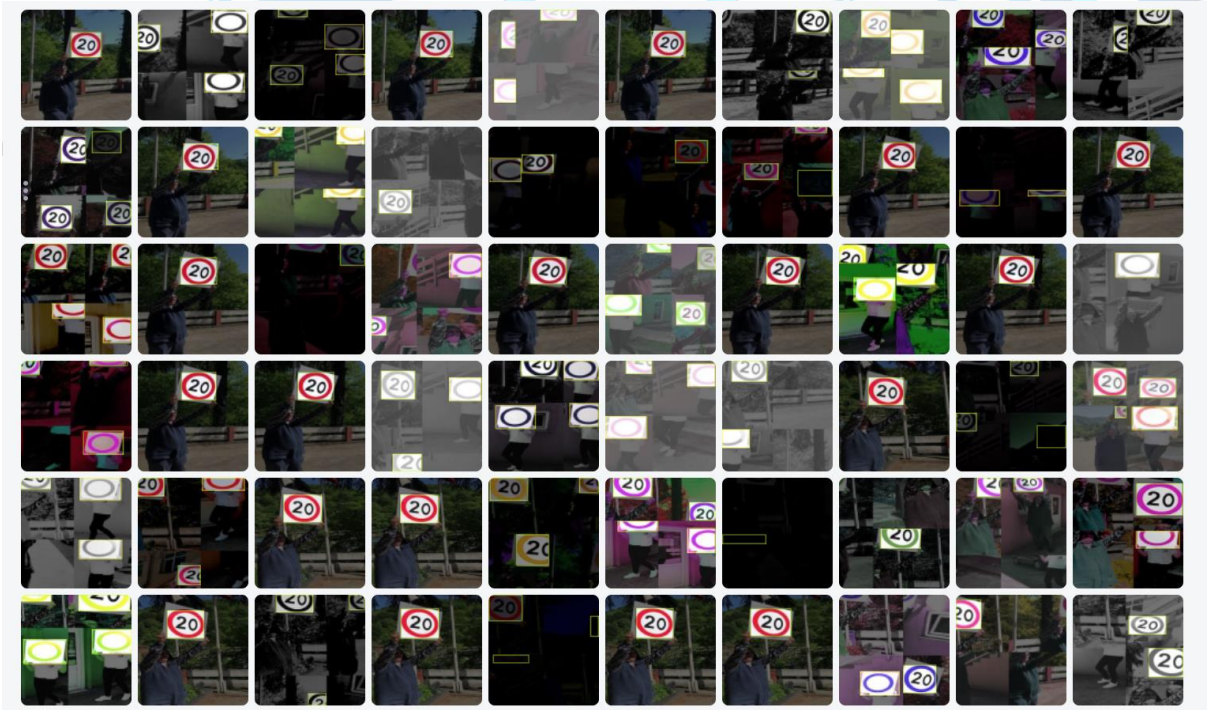
5.2.1 Trafik İşaret Levhalarını Tanıma

Trafik işaret levhalarını algılamada YOLOv4 algoritmasını kullandık. Veri seti oluşturmak için levhaların çeşitli koşullarda videolarını çekip bu videolar içinden veri seti eklenecek fotoğraf kareleri seçtik ve böylece görüntü işlerken gerçek ortamda işaret levhalarında oluşabilecek yansıma ve benzeri durumlara karşı önlem almış olduk. İlk aşamadan sonra veri setimizde etiketlenmemiş yaklaşık 10000 fotoğraf kalmış bulunmaktaydı. Bu veri setini bizim fotoğraflar üzerinde daha verimli çalışmamıza olanak veren Roboflow uygulamasına yükledik. Etiketlemeyi kolaylaştırmak amacıyla toplam fotoğrafın 2000 civarını etiketledikten sonra bu etiketlediğimiz fotoğrafları Roboflow'un Roboflow Train altyapısından geçirerek küçük bir veri seti elde ettik ve bu veri setini sonraki fotoğrafları etiketlerken kolaylık olması amacıyla Roboflow'un özelliklerinden biri olan ön algılama sisteminden geçirdik. Bunun sonucunda geri kalan fotoğraflar otomatik olarak etiketlenmiş oldu, bizim sadece etiketlerin doğru olup olmadığını kontrol etmemiz gerekti ve böylece zamandan tasarruf etmiş olduk.



Şekil 5.2.1 Roboflow tabela etiketleme görseli

Etiketleri kontrol ettikten sonra veri setini Roboflow üzerinden YOLO Darknet formatında dışa aktardık. Görüntü işlerken doğruluk oranımızın artması için ve sonuçların daha tutarlı olması için dışa aktarırken fotoğraflara gürültü, kontrast, parlaklık gibi modifikasyonlar ekledik.



Şekil 5.2.2 Etiketlenmiş ve renk filtrelerinden geçirilmiş görseller

Veri setinin eğitimini hızlandırmak için Google'ın hizmetlerinden biri olan Colab'i kullandık. Google Colab görüntü işleme ve benzeri görevleri daha hızlı ve verimli yapmak için merkezi

serverları kullanan ve altyapısında işletim sistemi olarak Ubuntu 18.04 LTS, CPU olarak 2.3GHz saat hızında Intel Xeon ve GPU olarak Nvidia Turing ve Tesla modellerini kullanan bir kod çalıştırma hizmetidir. Eğitimi hızlandırmak için eğitimde Nvidia cuDNN 7.6.5 ve CUDA 11.2 kullandık. CUDA ve CUDNN, Nvidia'nın geliştirdiği, görüntü işleme ve benzeri işlerde GPU ile kod arasındaki iletişimi evrenselleştiren ve kolaylaştıran bir yazılımdır.

```
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 470.74       Driver Version: 460.32.03    CUDA Version: 11.2     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|                               |                      |               MIG M. |
|===============================+======================+======================|
|   0  Tesla P100-PCIE...  Off  | 00000000:00:04.0 Off |                    0 |
| N/A   39C    P0    26W / 250W |      0MiB / 16280MiB |      0%      Default |
|                               |                      |                  N/A |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                                  |
|  GPU   GI   CI        PID   Type   Process name                  GPU Memory |
|        ID   ID                                                   Usage      |
|=============================================================================|
|  No running processes found                                                 |
+-----------------------------------------------------------------------------+
```

Şekil 5.2.3 Google Colab eğitim görselleri

Veri setini eğitirken AlexeyAB'nın YOLOv4 ve YOLOv4 Tiny için geliştirdiği eğitim kodunun Roboflow için düzenlenmiş halini kullandık.

Eğitirken ilk önce YOLOv4 eğitim kodunu Colab'e yükledikten sonra bu kodu Roboflow ile daha uyumlu hale getirecek Roboflow eklentisini yükledik.

```
[ ] #we clone the fork of darknet maintained by roboflow
    #small changes have been made to configure darknet for training
    !git clone https://github.com/roboflow-ai/darknet.git

    Cloning into 'darknet'...
    remote: Enumerating objects: 13289, done.
    remote: Total 13289 (delta 0), reused 0 (delta 0), pack-reused 13289
    Receiving objects: 100% (13289/13289), 12.13 MiB | 9.35 MiB/s, done.
    Resolving deltas: 100% (9105/9105), done.
```

```
#follow the link below to get your download code from from Roboflow
!pip install -q roboflow
from roboflow import Roboflow
rf = Roboflow(model_format="darknet", notebook="roboflow-yolov4")
```

Şekil 5.2.4 Google Colab üzerinden Darknet sinir ağı kullanılarak veri seti eğitme görselleri

Sonra Roboflow'dan veri setimizi indirip eğitim klasörlerine kopyaladık ve veri setinin büyüklüğü, içindeki isimler ve benzeri bilgileri belirten .data, .names, .obj ve .cfg dosyalarını oluşturarak eğitime hazır hale getirdik.

```
#we build config dynamically based on number of classes
#we build iteratively from base config files. This is the same file shape as cfg/yolo-obj.cfg
def file_len(fname):
  with open(fname) as f:
    for i, l in enumerate(f):
      pass
  return i + 1

num_classes = file_len(dataset.location + '/train/_darknet.labels')
print("writing config for a custom YOLOv4 detector detecting number of classes: " + str(num_classes))

#Instructions from the darknet repo
#change line max_batches to (classes*2000 but not less than number of training images, and not less than 6000), f.e. max_batches=6000 if you train for 3 classes
#change line steps to 80% and 90% of max_batches, f.e. steps=4800,5400
if os.path.exists('./cfg/custom-yolov4-detector.cfg'): os.remove('./cfg/custom-yolov4-detector.cfg')


with open('./cfg/custom-yolov4-detector.cfg', 'a') as f:
  f.write('[net]' + '\n')
  f.write('batch=64' + '\n')
  #####smaller subdivisions help the GPU run faster. 12 is optimal, but you might need to change to 24,36,64####
  f.write('subdivisions=24' + '\n')
  f.write('width=416' + '\n')
  f.write('height=416' + '\n')
  f.write('channels=3' + '\n')
  f.write('momentum=0.949' + '\n')
  f.write('decay=0.0005' + '\n')
  f.write('angle=0' + '\n')
  f.write('saturation = 1.5' + '\n')
  f.write('exposure = 1.5' + '\n')
  f.write('hue = .1' + '\n')
  f.write('\n')
  f.write('learning_rate=0.001' + '\n')
  f.write('burn_in=1000' + '\n')
  #####you can adjust up and down to change training time#####
  ##Darknet does iterations with batches, not epochs####
  # max_batches = num_classes*2000
  max_batches = 2000
  f.write('max_batches=' + str(max_batches) + '\n')
  f.write('policy=steps' + '\n')
  steps1 = .8 * max_batches
  steps2 = .9 * max_batches
  f.write('steps='+str(steps1)+','+str(steps2) + '\n')
```

Şekil 5.2.5 Eğitime hazırlık son aşama görselleri

Hazırlıklar tamamlandığında eğitimi başlattık.

Train Custom YOLOv4 Detector

```
!./darknet detector train data/obj.data cfg/custom-yolov4-detector.cfg yolov4.conv.137 -dont_show -map
#If you get CUDA out of memory adjust subdivisions above!
#adjust max batches down for shorter training above
```

Şekil 5.2.6 Veri seti eğitimi

Eğitimlerimiz YOLOv4 ile ortalama 5 saat, YOLOv4 Tiny ile ortalama 2,5 saat sürdü ve sırasıyla %89-94 ve %92-%97 civarında doğruluk oranı aldık. Eğitim bittikten sonra görüntü işlemek için gerekli dosyaları Google Colab'den bilgisayarımıza indirdik. Burada elde ettiğimiz eğitilmiş veri setiyle realtime görüntü işlemek için bilgisayarımıza gerekli olan CMake, CUDA, cuDNN, make, git, g++ yazılımlarını yükledik. CMake yazılımının CUDA'yı destekleyen 3.16 sürümünü APT (Advanced Package Tool) paket yönetim sistemiyle yükledik. CMake kurulduktan sonra bilgisayar CUDA yüklemek için hazır hale gelecektir. CUDA'yı Nvidia'nın serverlarından indirdiğimiz gerekli Nvidia sürücüleri ile birlikte gelen CUDA Toolkit debian local yükleyicisi ile bilgisayarımıza kurduk ve görüntü işleme kütüphanesinin CUDA'yı

bulabilmesi için CUDA'nın yüklü olduğu yeri bilgisayarın PATH dosyasına ekledik.

```
CUDA Installer
- [X] Driver
     [X] 515.43.04
+ [X] CUDA Toolkit 11.7
  [X] CUDA Demo Suite 11.7
  [X] CUDA Documentation 11.7
  Options
  Install

Up/Down: Move | Left/Right: Expand | 'Enter': Select | 'A': Advanced options
```

Şekil 5.2.7 Eğitim Dosyası

Bir sonraki adımda Nvidia'nın özellikle derin sinir ağlarına destek olması için CUDA'ya ek olarak çıkardığı cuDNN eklentisini yükledik. Son olarak YOLOv4 kütüphanesini Github'dan indirmemizi sağlayacak git, indirilen kütüphanenin makefile dosyasını düzenleyip ona göre kütüphaneyi ayarlayacak make ve kütüphanenin C++ kodunu derleyecek g++ yazılımlarını apt ile yükledik. Bu aşamadan sonra veri setini eğitirken kullandığımız AlexeyAB'nin YOLOv4 inference kodunu bilgisayarımıza git komutu ile indirip,

```
/content/darknet# git clone https://github.com/AlexeyAB/darknet
Cloning into 'darknet'...
remote: Enumerating objects: 15424, done.
remote: Counting objects: 100% (1/1), done.
remote: Total 15424 (delta 0), reused 1 (delta 0), pack-reused 15423
Receiving objects: 100% (15424/15424), 14.03 MiB | 14.36 MiB/s, done.
Resolving deltas: 100% (10365/10365), done.
```

Şekil 5.2.8 Cuda İnference Kodu

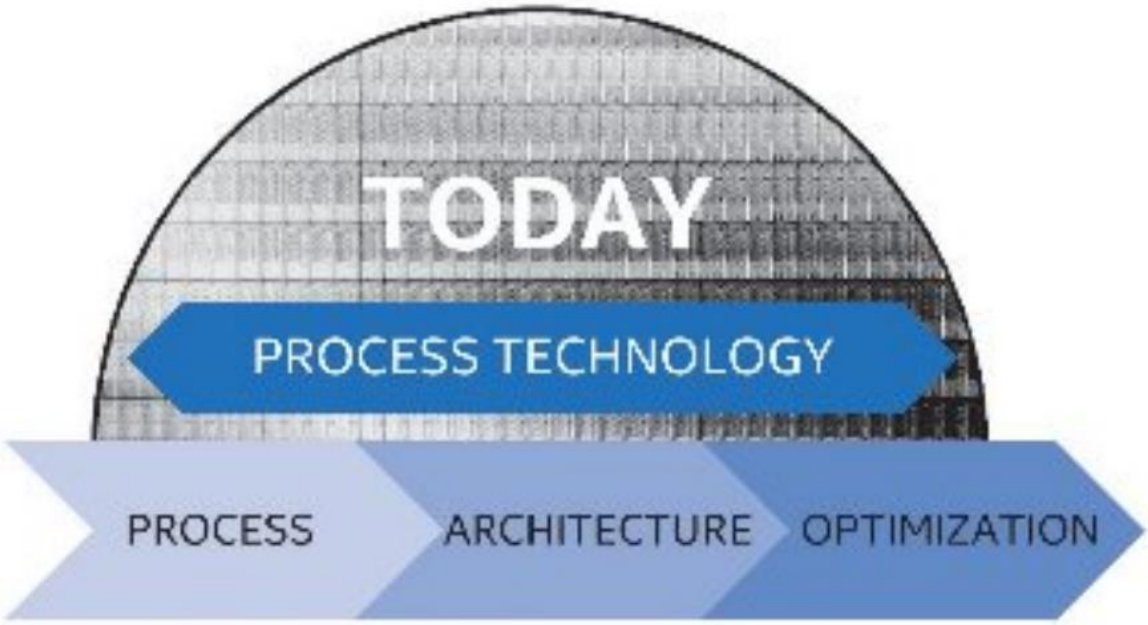
CUDA'nın yüklü olduğunu Makefile'da belirttikten sonra make komutu ile kütüphaneyi düzenleyip derledik.

```
/content/darknet/darknet# make
mkdir -p ./obj/
mkdir -p backup
chmod +x *.sh
g++ -std=c++11 -std=c++11 -Iinclude/ -I3rdparty/stb/include -Wall -Wfatal-errors -Wno-unused-result -Wno-unknown-pragmas -fPIC -Ofast -c ./src/image_opencv.cpp -o obj/image_op
encv.o
g++ -std=c++11 -std=c++11 -Iinclude/ -I3rdparty/stb/include -Wall -Wfatal-errors -Wno-unused-result -Wno-unknown-pragmas -fPIC -Ofast -c ./src/http_stream.cpp -o obj/http_stre
am.o
./src/http_stream.cpp: In member function 'bool JSON_sender::write(const char*)':
./src/http_stream.cpp:253:21: warning: unused variable 'n' [-Wunused-variable]
                int n = _write(client, outputbuf, outlen);
                    ^

./src/http_stream.cpp: In function 'void set_track_id(detection*, int, float, float, float, int, int, int)':
./src/http_stream.cpp:867:27: warning: comparison between signed and unsigned integer expressions [-Wsign-compare]
         for (int i = 0; i < v.size(); ++i) {
                         ~~^~~~~~~~~
./src/http_stream.cpp:875:33: warning: comparison between signed and unsigned integer expressions [-Wsign-compare]
     for (int old_id = 0; old_id < old_dets.size(); ++old_id) {
                          ~~~~~~~^~~~~~~~~~~~~~~
./src/http_stream.cpp:894:31: warning: comparison between signed and unsigned integer expressions [-Wsign-compare]
     for (int index = 0; index < new_dets_num*old_dets.size(); ++index) {
                         ~~~~~~^~~~~~~~~~~~~~~~~~~~~~~~~~~
./src/http_stream.cpp:930:28: warning: comparison between signed and unsigned integer expressions [-Wsign-compare]
     if (old_dets_dq.size() > deque_size) old_dets_dq.pop_front();
         ~~~~~~~~~~~~~~~~~~^~~~~~~~~~~
gcc -Iinclude/ -I3rdparty/stb/include -Wall -Wfatal-errors -Wno-unused-result -Wno-unknown-pragmas -fPIC -Ofast -c ./src/gemm.c -o obj/gemm.o
./src/gemm.c: In function 'convolution_2d':
./src/gemm.c:2044:15: warning: unused variable 'out_w' [-Wunused-variable]
     const int out_w = (w + 2 * pad - ksize) / stride + 1;    // output_width=input_width for stride=1 and pad=1
               ^~~~~
./src/gemm.c:2043:15: warning: unused variable 'out_h' [-Wunused-variable]
     const int out_h = (h + 2 * pad - ksize) / stride + 1;    // output_height=input_height for stride=1 and pad=1
               ^~~~~
gcc -Iinclude/ -I3rdparty/stb/include -Wall -Wfatal-errors -Wno-unused-result -Wno-unknown-pragmas -fPIC -Ofast -c ./src/utils.c -o obj/utils.o
```

Şekil 5.2.9 Kütüphane Görseli

Google Colab'den indiridiğimiz veri setini darknet klasörüne taşıdıktan sonra sabit 60 FPS görüntü işleme yapar hale geldik. Görüntü işlerken teknik özellikleri Nvidia RTX 3060 GPU, Intel Tiger Lake jenerasyonu CPU ve Samsung marka 2 TB NVMe SSD olan bir bilgisayar kullandık ve bu bilgisayara yazılım sistemi olarak Ubuntu 22.04 LTS yükledik. GPU olarak Nvidia RTX 3060 kullanmamızın sebebi cuDNN ve CUDA hızlandırıcılarının sadece Nvidia GPU'larda çalışması ve RXT 3060'ın diğer benzer fiyattaki diğer Nvidia ürünlerinden daha fazla CUDA çekirdeklerine sahip olmasıdır (3584 tane). CPU'muzu Intel'in Tiger Lake jenerasyonundan seçmemizin sebebi bu nesilden CPU'ların Intel'in 10nm Willow Cove mikromimarisini kullanmasıdır ve Tiger Lake jenerasyonunun Intel'in CPU geliştirme takviminde optimizasyon yılına denk geldiği için daha stabil olmasıdır.

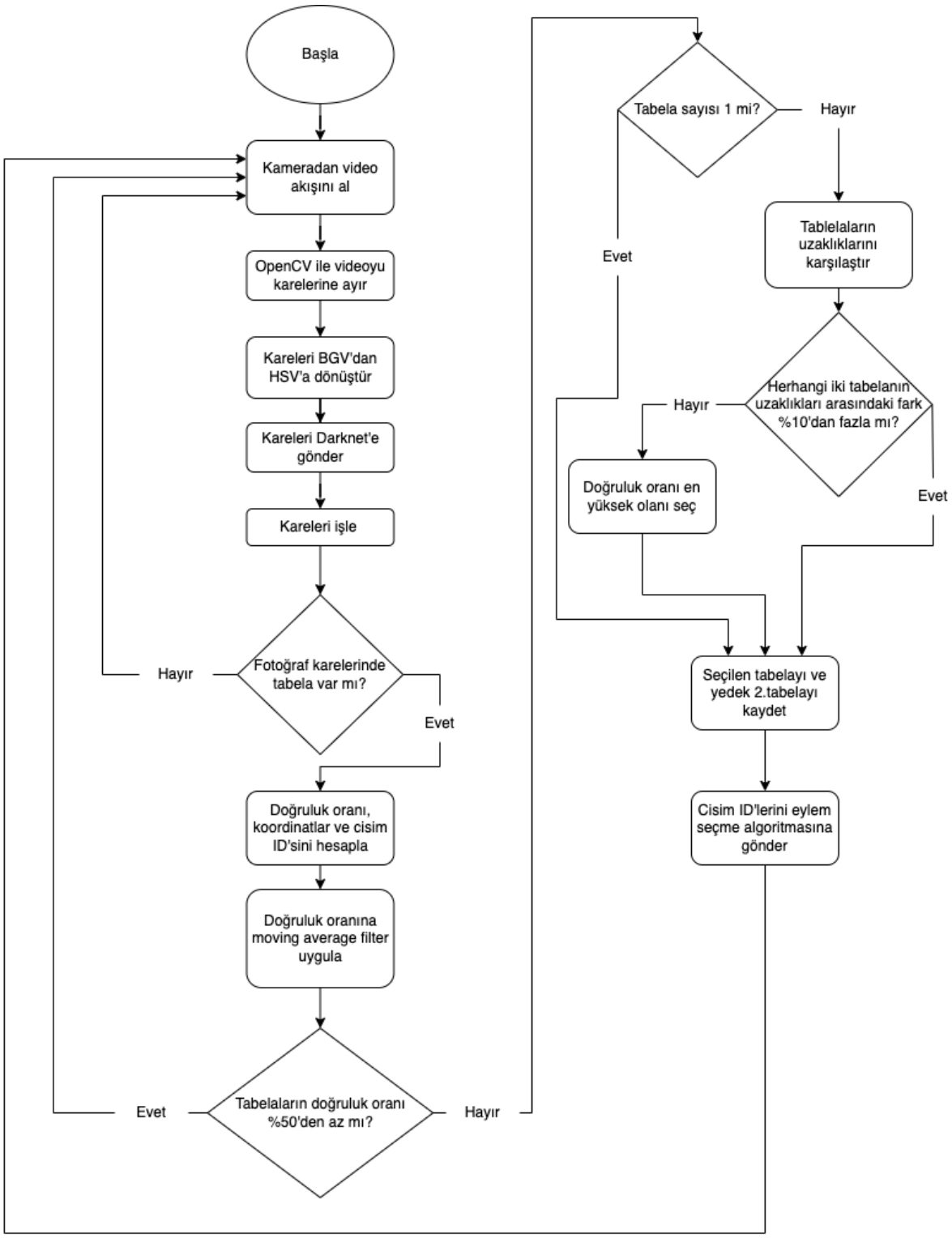


Şekil 5.2.10 Teknoloji Basamakları

İşletim sistemi olarak Ubuntu'yu seçmemizin sebebi Ubuntu'nun bizim ihtiyaçlarımıza en uygun işletim sistemi olmasıdır ve özellikle Ubuntu 22.04 LTS versiyonunu seçmemizin sebebi Ubuntu'nun şuan en güncel LTS (Long-Term Support) versiyonu olmasıdır. LTS versiyonu olmasının özelliği Ubuntu'yu yöneten şirket Canonical'ın LTS versiyonları için en az 7 yıl güvenlik güncellemesi + 3 yıl paralı ekstra destek güvencesi vermesidir. Görüntü işleme kamerası olarak Logitech C310 HD 720p WebCam kullanmamızın nedeni bilgisayarımızın GPU'sunu çok yüksek çözünürlüklerle meşgul etmezken yinede görüntü işlemenin verimli ve doğruluk oranı yüksek bir şekilde devam edebilmesidir.

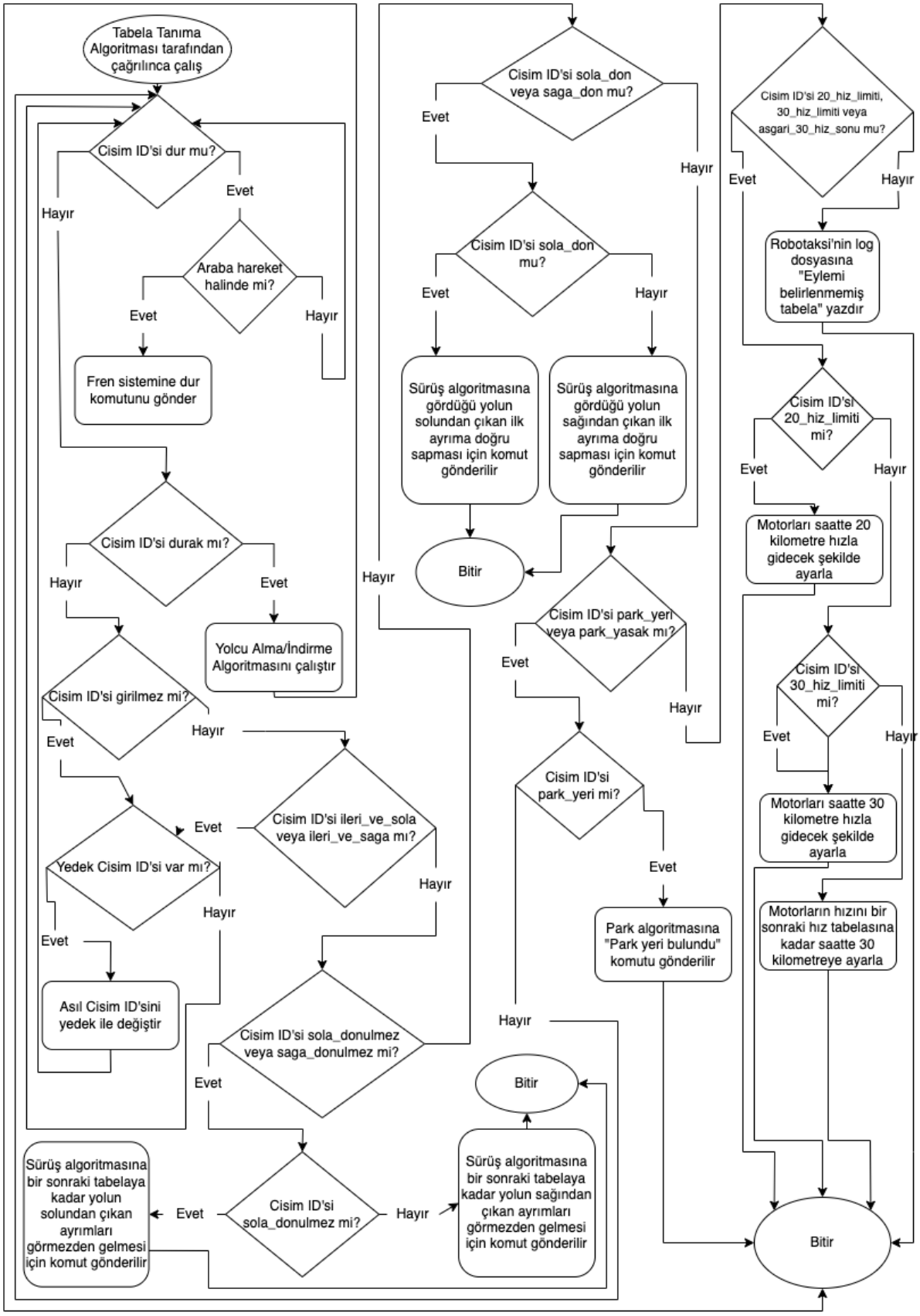
Trafik Tabelası Tanıma Algoritması

TEKNOFEST
HAVACILIK, UZAY VE TEKNOLOJİ FESTİVALİ



Şema 5.1.1 Trafik Tabelası Tanıma Algoritması

Tabela Eylem Seçme Algoritması



Şema 5.1.2 Tabela Eylem Seçme Algoritması

5.2.2 Trafik Işıkları

Trafik işaret ve ışıkları tespit etmek için nesne tespit algoritmalarından yararlanılmıştır.

Nesne tespiti basitçe, verilen görüntüde istenilen nesnelerin görüntüdeki pozisyonunun ve nesnenin sınıfının belirlenmesidir.

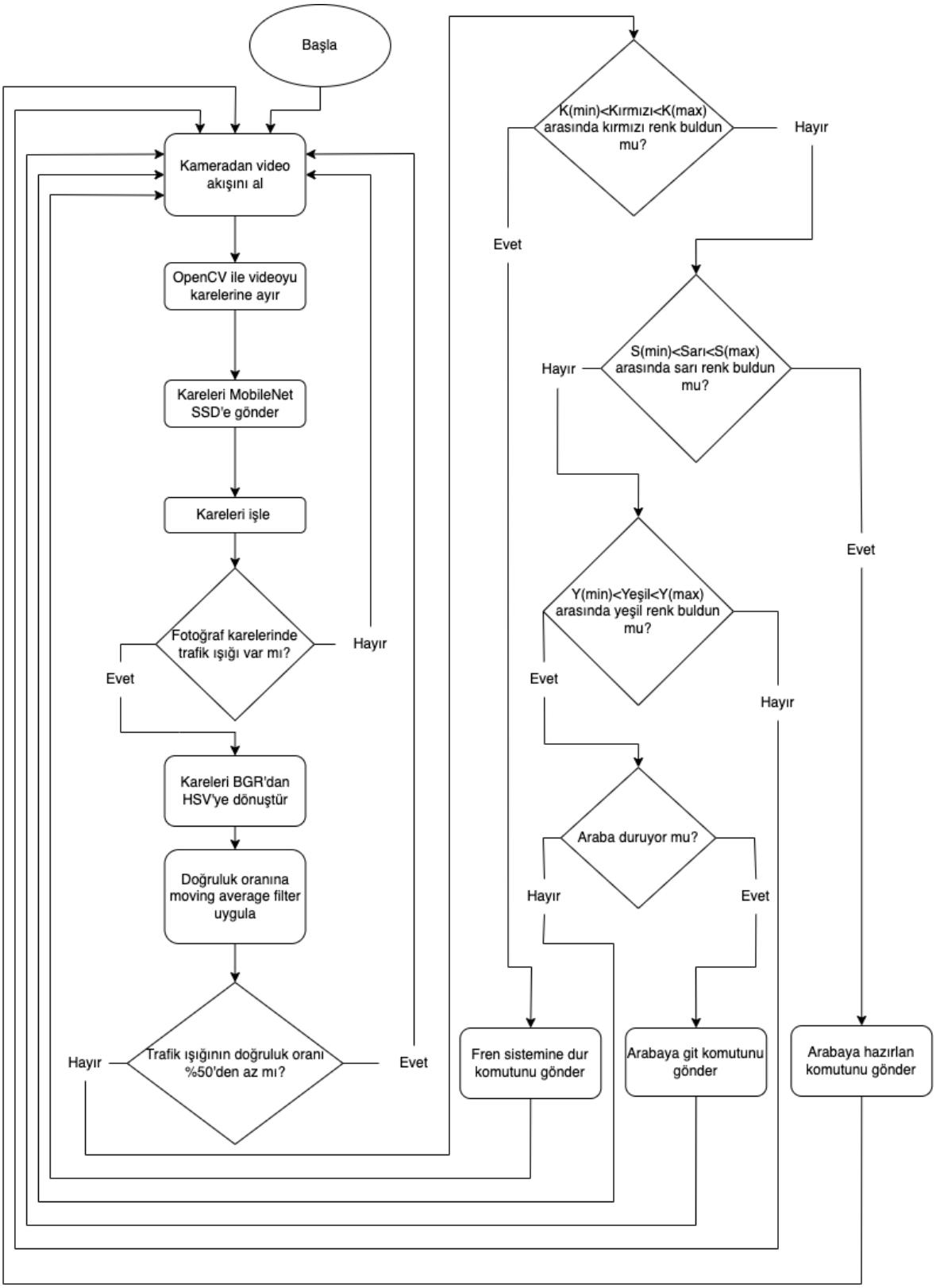
Güçlü grafik işlemciler sayesinde bilgisayar görüşü ve derin öğrenme yöntemleri kullanılarak geliştirilen algoritma sayısı her geçen gün artmıştır. Bu şekilde yapay sinir ağlarını kullanan birçok nesne tespit algoritması üretilmiştir.

Örnek bir senaryoda park edilebilir levhası gören araç, görüntüyü işlemeye dayanarak park modu kararını almış olur ve böylece park modu aktif edilmiş olur. Park modu aktif edilince araç sol şeridi park manevrasına başlayana kadar park alanını gezmiş olur. Nesne tespiti problemi için standart bir yapay sinir ağı kullanılamaz, bunun ilk sebebi görüntüdeki nesne sayısının belli olmaması ve bundan dolayı çıktı katmanının büyüklüğünün belirli olmamasıdır. Aynı zamanda her bir pikselin renk değerinin yapay sinir ağına girdi olarak verilmesi, ağın çok yavaş çalışmasına ve çok büyük bilgisayar hafızası ihtiyacına yol açacaktır.

Bu tip problemleri çözmek için bir yaklaşım, görüntüden farklı bölgeler almak ve bu bölgedeki nesnenin varlığını evrişimsel sinir ağlar (CNN) ile sınıflandırmaktır. Ancak bu yöntemin de belli başlı eksik olduğu noktalar vardır. Görüntüdeki nesnelerin alabileceği şekillerden ötürü seçilecek bölgenin en boy oranı da herhangi bir oran olabilir. Bundan dolayı aşırı miktarda bölge seçip işlem gücü gereksinimini devasa boyutlara çıkarmak yerine R-CNN, YOLO, MobileNet, Tensorflow vb. gibi bu işlemi hızlandırmak için geliştirilmiş algoritmalar kullanılması en efektif çözümlerden biridir.

Her bir hücre içerisinde bulunan nesne veya nesneye ait bir parça için 5 farklı sınırlama kutusu tahmini yapar. Aynı zamanda her bir sınırlama kutusu için içerisinde herhangi bir nesne olma ihtimali hesaplanır. Ancak bu skor kutuların içindeki nesnenin sınıfıyla ilgili bir bilgi vermez.

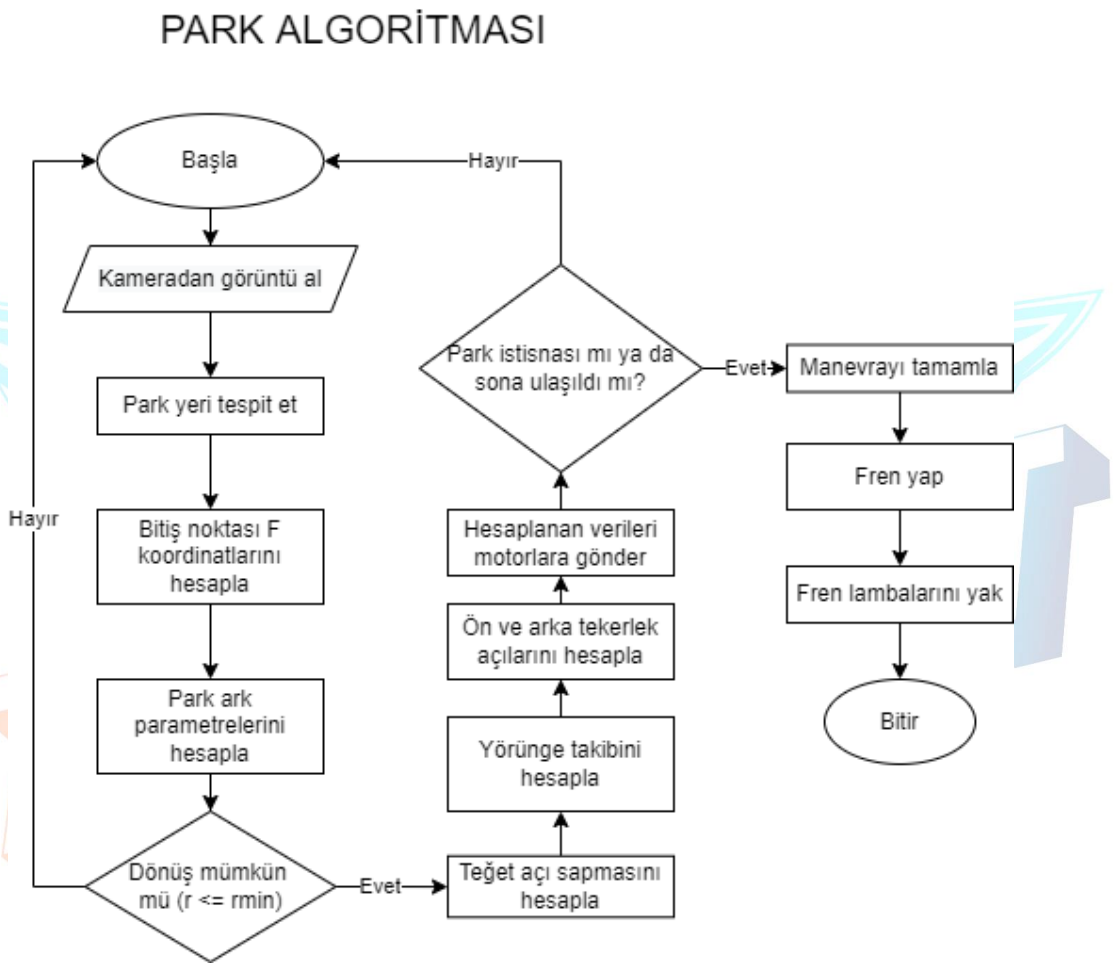
Trafik Işıkları Tanıma Algoritması



Şema 5.2.2 Trafik Işıkları Tanıma Algoritması

5.2.3 Otonom Park Etme Sistemleri

Otonom aracımız şartnamede belirtilen güzergahta ilerleyecek ve park algoritması kullanılarak belirtilen noktaya park edecektir. Bu işlemi öncelikle aracın üstünde bulunan kameralar ile park noktasını belirten trafik levhasını nesne tanıma (object detection) algoritmasıyla algılayarak park edeceği noktayı belirleyecek. Park yeri tespit edildiğinde park etme işlemini de içinde bulunan bulunan gömülü yazılımı sayesinde gerçekleştirecek.



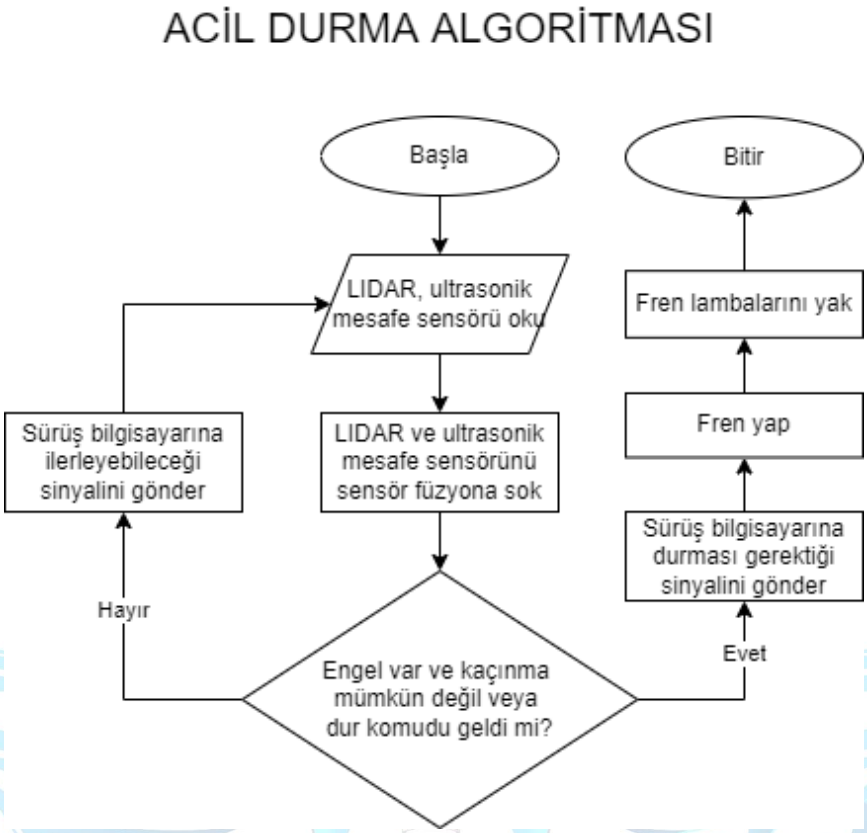
Şema 5.2.3 Park Algoritması

5.2.4 Kaza Önleme Algoritması

Aracımız önünde bulunan sensörlerden gelen verileri alır ve sensör füzyon uyguladıktan sonra aracın ön tarafında 1 metre içinde herhangi bir nesne, insan, engel vs. var mı diye bakar.

Varsa bu durum için üstünde çalışacağımız ani fren sistemi devreye sokulur, fren lambası yakılır ve araç durur. Araç bu kodun devreye sokulmasına karar verdikten sonra oluşacak olan gecikme,

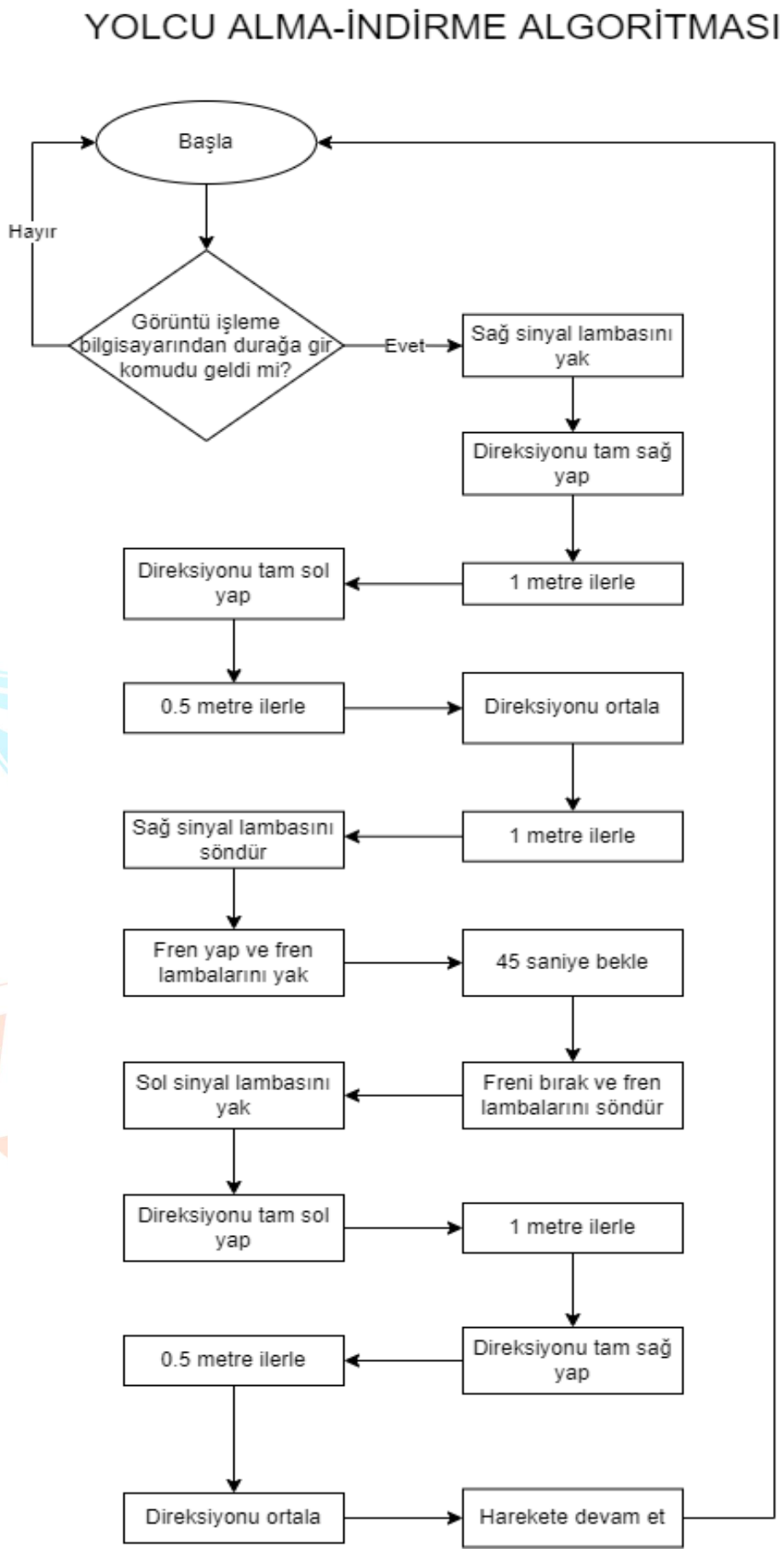
lastikler bir anda durduktan sonra yaşanacak olan kayma vs. hesaba katılarak en uygun mesafenin 1 metre olduğuna karar verilmiştir.



Şema 5.2.4 Acil Durma Algoritması

5.2.5 Otonom Yolcu Alma İndirme

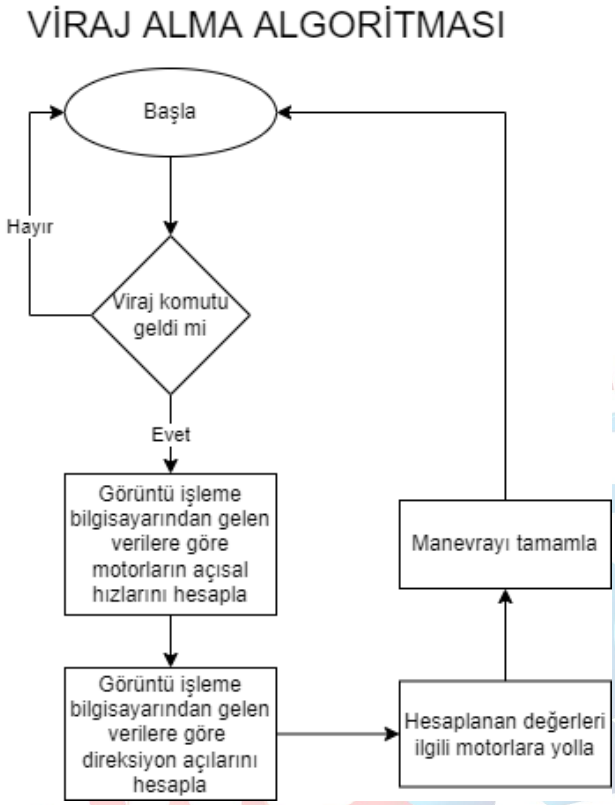
Görüntü işleme bilgisayarı durak tespit ettiğinde sürüş bilgisayarına durağa gir komutu gönderecek. Ardından kendini otomatik olarak ortalayarak duracak 45 saniye bekleyecek ve yolcuları alma işlemini gerçekleştirecektir. Aynı yöntemle indirme işlemini de gerçekleştirecek ve görevini bu şartlarda tamamlayacaktır.



Şema 5.2.5 Yolcu Alma-İndirme Algoritması

5.2.6 Viraj Algoritması

Görüntü işleme bilgisayarı viraja yaklaştığını tespit ettiğinde sürüş bilgisayarına viraj komutu gönderecek. Ardından aracımız görüntü işleme bilgisayarından gelen verilere göre motorların açısal hızlarını ve direksiyon açılarını hesaplayacak ve hesaplanan değerleri ilgili motorlara yollayacak ve bunun sonucunda manevra tamamlanacak. Ardından aracımız görüntü işleme bilgisayarından tekrar viraj komutu gelmesini bekleyecek.



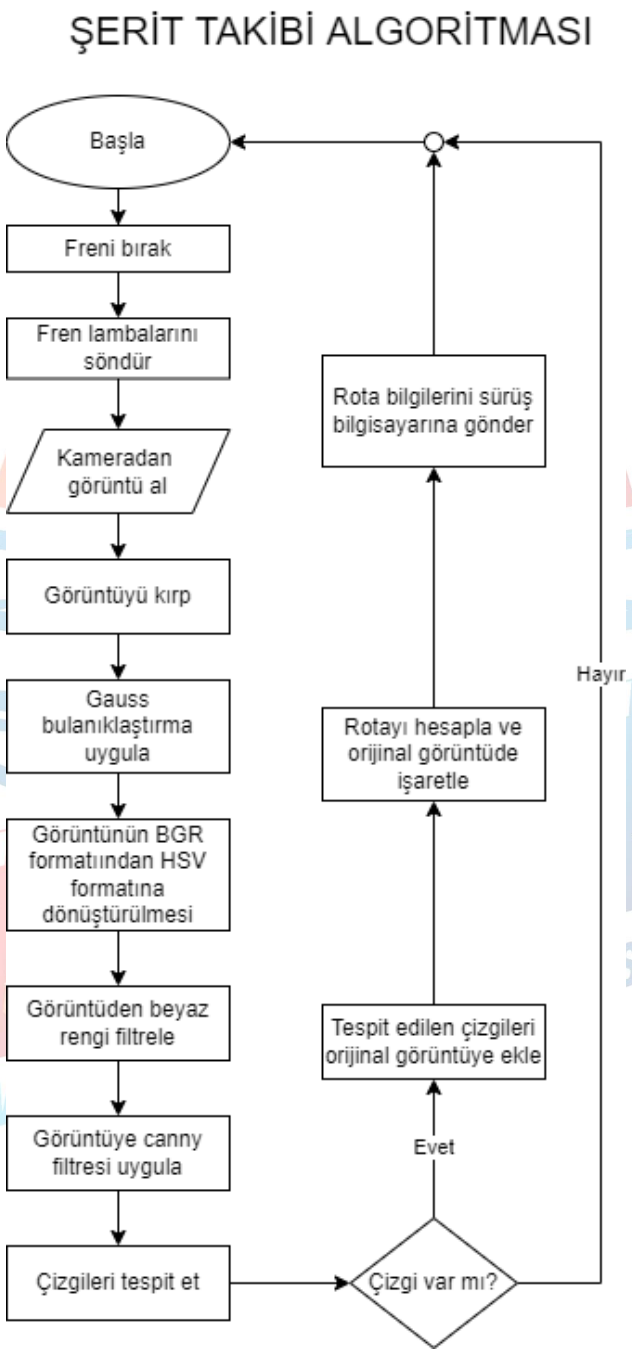
Şema 5.2.6 Viraj Alma Algoritması

5.3 Şerit Takibi

Aşağıdaki adımlarla şerit takibi yapılmaktadır:

1. Kameradan görüntüyü al
2. Gereksiz kısımları kırp
3. Gürültüden kurtulmak için Gauss bulanıklaştırma uygula
4. Görüntüyü BGR formatından HSV formatına dönüştür
5. Görüntüden beyaz rengi filtrele
6. Görüntüye Canny Edge Detector uygula
7. Çizgileri tespit et (HoughLines metodu ile)
8. “Çizgi bulunduysa devam et, bulunmadıysa 1. adıma dön” karar adımı

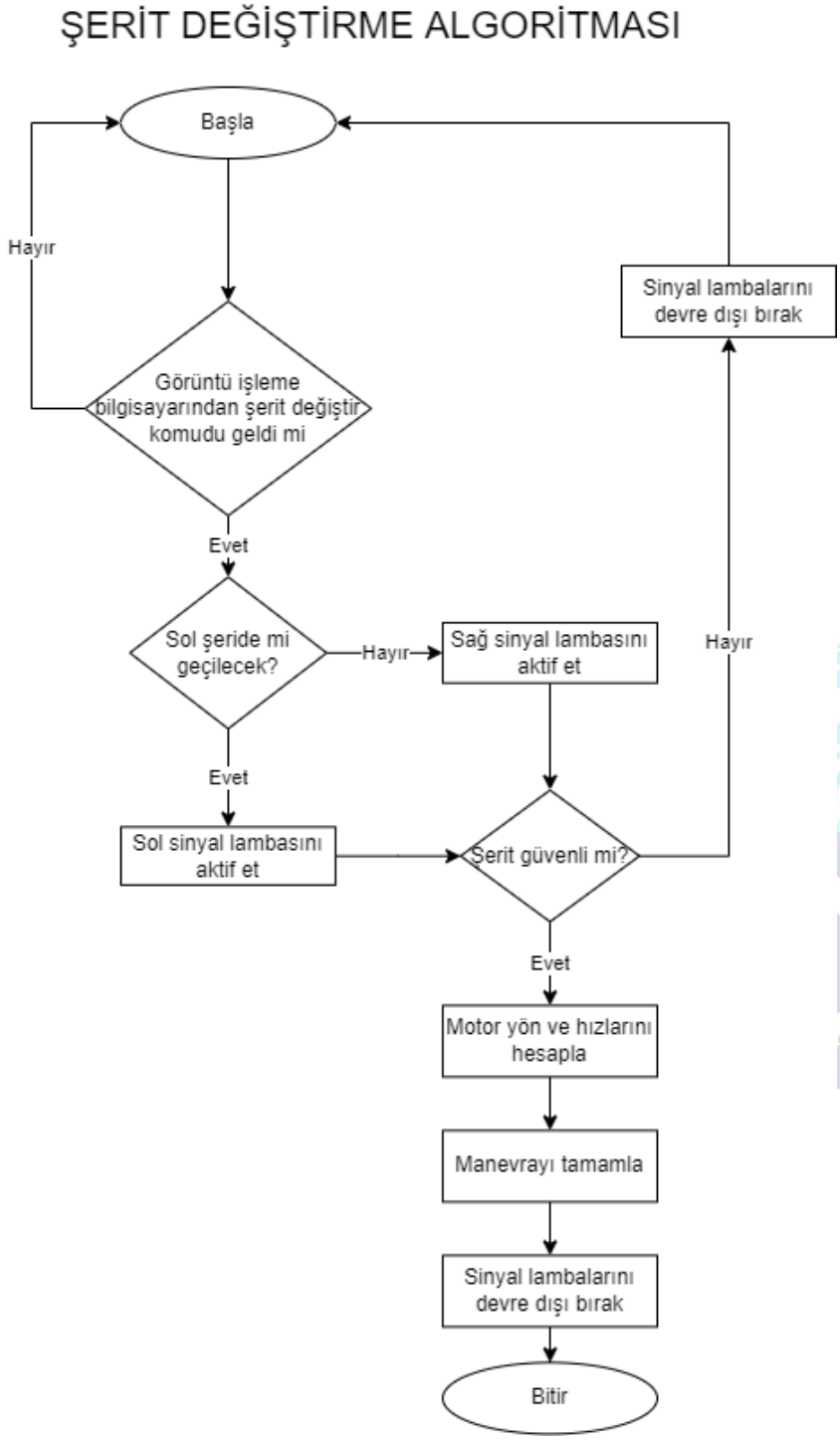
9. Çizgileri orijinal görüntüye çiz
10. Takip edilecek noktayı bul ve orijinal görüntüye çiz
11. Yeni çizginin yatay koordinatını sürüş bilgisayarına gönder



5.3 Şerit Takip Algoritması

5.4 Şerit Değiştirme Algoritması

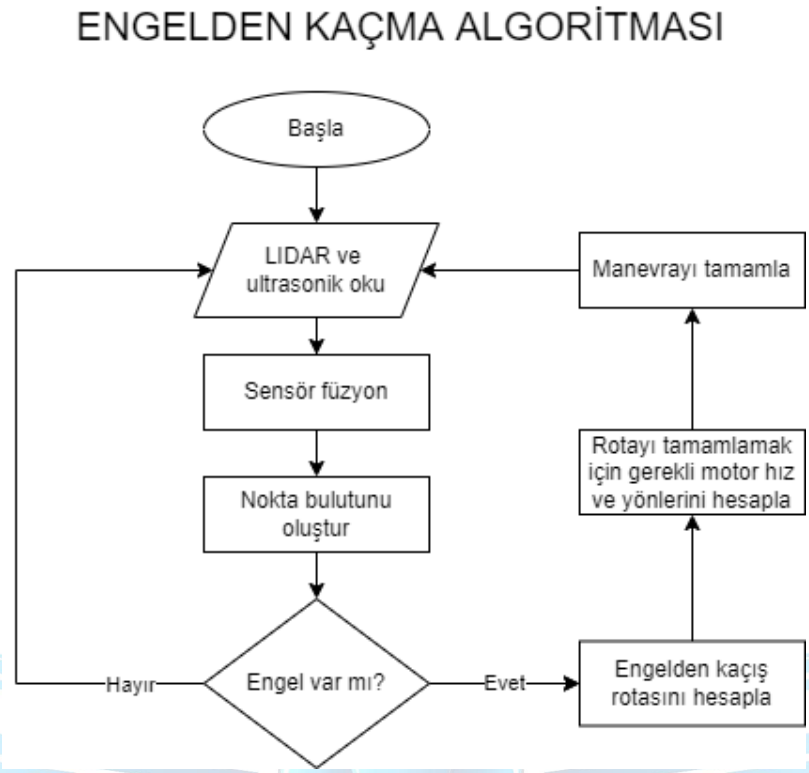
Sürüş bilgisayarına görüntü işleme bilgisayarından şerit değiştir komutu geldiğinde araç bu algoritma mantığı ile hareket eder.



Şema 5.4 Şerit Değiştirme Algoritması

5.5 Engelden Kaçma Algoritması

LIDAR ve ultrasonik sensörlerden alınan verilere göre aracın önüne engel çıktığına karar verilirse araç bu algoritmaya göre hareket eder.



Şema 5.5 Engelden Kaçma Algoritması

6. Özgün Bileşenler

6.1 Tasarımsal Özgünlük

6.1.1 Araç Şasesi : Yarışma şartnamesindeki araç özellikleri dikkate alınarak aracımızın şasesi solidWorks programında tasarlanmıştır.

6.1.2 Araç Kabuğu : Yarışma şartnamesindeki araç özellikleri dikkate alınarak aracımızın şasesi solidWorks programında tasarlanmıştır.

6.1.3 Elektronik Direksiyon Sistemi : Araçta Double Ackermann steering system kullanabilmek için ön ve arkada olmak üzere iki adet direksiyon sistemi bulunmaktadır. Tasarımını kendimiz yaptık. Şekil 3.1.1'de görüldüğü gibi bizim tarafımızdan tasarlanmıştır.

6.1.4 Elektronik Gaz Pedalı Sistemi : Manuel kullanım için Madde 3.2.1 de anlatıldığı gibi bizim tarafımızdan tasarlanmıştır.

6.1.5 Elektronik Fren Sistemi : Şekil 3.3'te görüldüğü gibi bizim tarafımızdan tasarlanmıştır.

6.1.6 Arka Teker Açısı Ayarlama Sistemi : Şekil 3.1.4 de görüldüğü gibi bizim tarafımızdan tasarlanmıştır.

6.1.7 Elektronik Diferansiyel : Kendimiz tarafından yapılmıştır.

6.1.8 Sürüş Kontrol Bilgisayarı : Kendimiz tarafından tasarlanmıştır.

6.2 Yazılımsal Özgünlük

6.2.1 Ani Fren Yazılımı : Sürüş Kontrol Bilgisayarı üzerine yazılacak olan ani fren yazılımı kendimiz tarafından geliştirilmiştir.

6.2.2 Yazılımsal Diferansiyel : Sürüş Kontrol Bilgisayarı üzerine yazılacak olan yazılımsal diferansiyel yazılımı kendimiz tarafından geliştirilmiştir.

6.2.3 Yazılımsal Double Ackerman : Sürüş Kontrol Bilgisayarı üzerine yazılacak olan yazılımsal double Ackermann yazılımı kendimiz tarafından geliştirilmiştir.

6.2.4 Batarya Yönetim Sistemi : Sürüş Kontrol Bilgisayarı üzerine yazılacak olan batarya yönetim sistemi yazılımı kendimiz tarafından geliştirilmiştir.

6.2.5 Veri Seti : Görüntü işlemede kullanılacak veri setini videolama yöntemiyle Roboflow'a aktararak kendimiz oluşturduk.

6.2.6 Görüntü İşleme Karar Yazılımı : Görüntü işleme bilgisayarlarından gelen verinin hangisinin kullanılacağına karar verecek olan yazılımı kendimiz hazırladık.

6.2.7 Yer İstasyonu Yazılımı : Araç bilgilerinin yer alacağı yer istasyonu yazılımını kendimiz hazırladık.

6.2.8 Görüntü işleme Algoritmaları : Görüntü işlemede kullanacağımız algoritmalarını yazılım mimarisi bölümünde adım adım nasıl gerçekleştirdiğimizi anlattık. Darknet sinir ağı üzerinden eğittiğimiz YOLOv4 (You Only Look Once) algoritmasını kullanarak görüntü işlememizi yaptık. Şerit takip algoritmasını yine yazılım mimarisinde açıkladığımız üzere kendimiz yaptık. Detaylı tasarım videosunda bunlar gözlemlenebilir olacaktır.

6.2.9 Kablolu Haberleşme Yazılımı: Tüm kablolu haberleşme yazılımları tarafımızdan yazılmıştır.

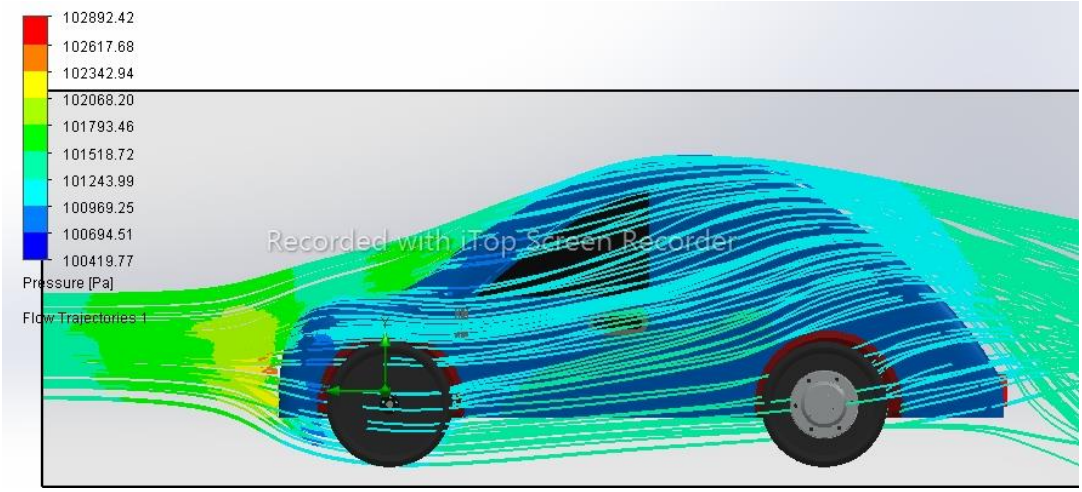
6.2.10 Kablosuz haberleşme Yazılımı: Tüm kablosuz haberleşme yazılımları tarafımızdan yazılmıştır.

7. Test

Aracımız Solidworks'te tasarlanmış ve mekanik testleri yapılmıştır. Sonrasında Unity'de simülasyonları yapılmıştır. En son gerçek hayat testleri yapılmıştır.

7.1 Mekanik Testler

7.1.1 Rüzgar Direnci Testleri



Şekil 7.1.1.1 Aracımızın ön rüzgar direnci testi

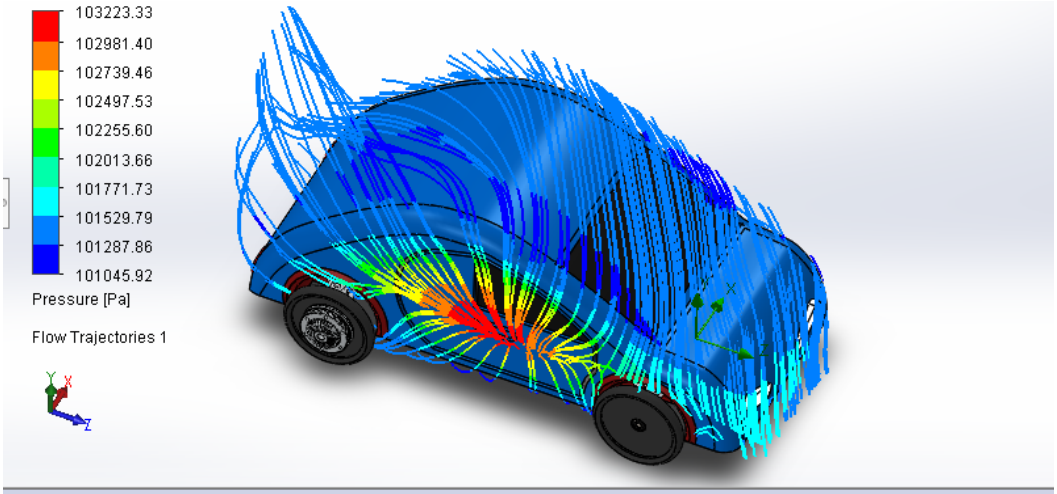
Havada hareket eden ya da bir hava akımının etkisinde kalan her türlü araç ve yapıların üzerine, aerodinamik kuvvet ve momentler etki eder. Bu kuvvet ve momentlerin bulunması, akım şeklinin ve yapısının belirlenmesi, tasarım açısından büyük önem taşır. Bu olguları belirlemek için havanın ağırlığı, basıncı. yoğunluğu, atmosferi ve denize yüksekliği gibi atmosferik değerleri hesapladık. Aracımızın aerodinamiğini rüzgar direncinden en az etkilenecek şekilde tasarladık. Rüzgar direnci testlerimizi ilimizde ya da yakın çevremizde rüzgar tüneli bulunmadığından dolayı ilçemize yakın yüksek rakımlı bi köy alanında yaptık. Köy alanında sonbahar aylarında 40 kilometre hıza ulaşan rüzgar, aracımıza azami bir direnç uyguladı. Bu kuvvetin etkisiyle eğimlenen aracı iki farklı şekilde test ettik: İlk olarak aracın rüzgara karşı aldığı eğimi göz önünde bulundurduk. İkinci yolumuz ise rüzgarın kuvvetini ölçmektir. Bunun için endüstriyel tarzı hassas tartı kullandık.

Rüzgar eğim testi

Rüzgar sebebiyle eğim alan aracı eğim ölçer sensörüyle ölçtük.

Rüzgar kuvvet testi

Rüzgar yükü, yapının geometrisi ve yüksekliğine bağlı olarak değişebilmektedir. Bu yük, basınç olarak etkilediği alana dik yayılı yük olarak ortaya çıkar. Kuvveti basit olarak rüzgar hızı ve basıncını kullanarak hesaplıyoruz. Yaptığımız hesaplamalar sonucu ortalama olarak 40 kilometre hızda rüzgarın 81,50 kg/m2 basınç uyguladığı tespit edilmiştir.



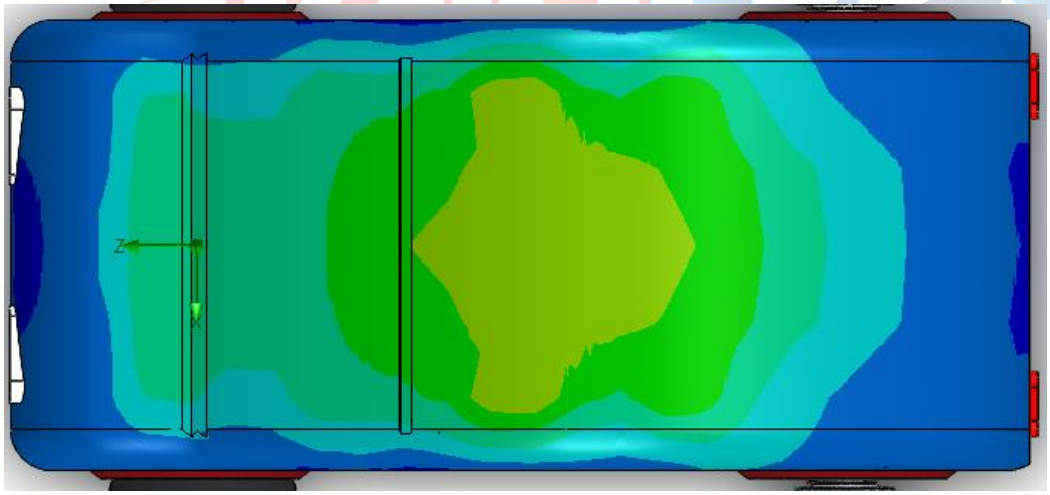
Summary

Goal Name	Unit	Value	Averaged Value	Minimum Value	Maximum Value	Progress [%]	Use In Convergence	Delta	Criteria
GG Normal Force (X) 1	[N]	1736.559	1736.851	1736.359	1737.702	100	Yes	1.343	63.521
GG Normal Force (Y) 1	[N]	60.764	62.895	60.764	64.391	100	Yes	2.738	6.750
GG Normal Force (Z) 1	[N]	-155.022	-155.799	-156.519	-155.022	100	Yes	0.700	17.169
GG Force (X) 1	[N]	1737.946	1738.276	1737.762	1739.197	100	Yes	1.435	63.476
GG Force (Y) 1	[N]	60.770	62.889	60.770	64.376	100	Yes	2.673	6.753
GG Force (Z) 1	[N]	-156.785	-157.576	-158.281	-156.785	100	Yes	0.607	17.458
GG Friction Force (X) 1	[N]	1.386	1.424	1.378	1.498	100	Yes	0.119	0.150
GG Friction Force (Y) 1	[N]	0.006	-0.007	-0.021	0.010	100	Yes	0.031	0.046
GG Friction Force (Z) 1	[N]	-1.763	-1.777	-1.827	-1.743	100	Yes	0.083	0.290

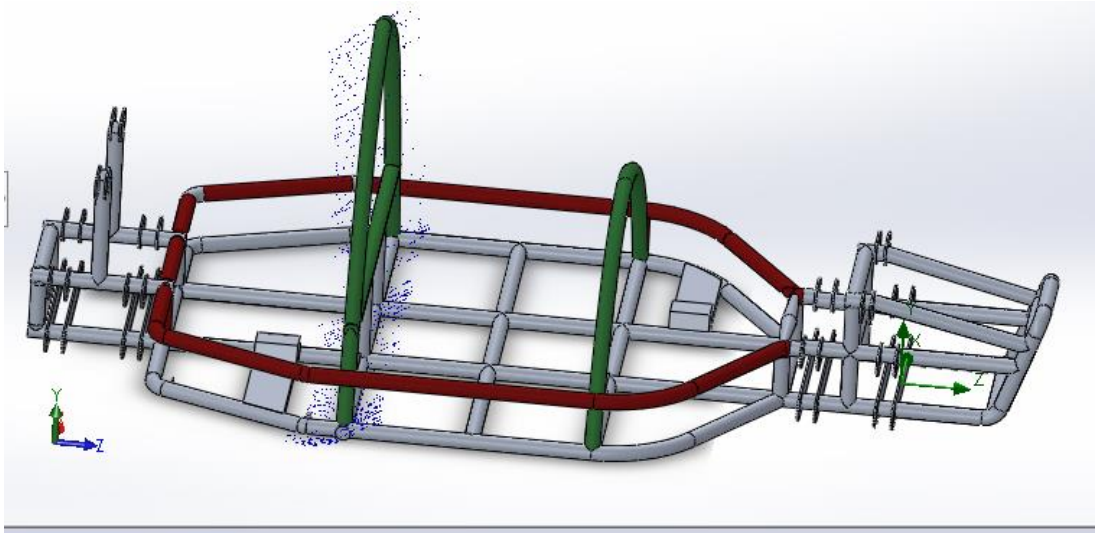
Şekil 7.1.1.2 Aracımızın yanal rüzgar direnci testi ve x-y-z düzlemlerine göre kayma bilgileri

Aracımıza uyguladığımız rüzgar direnci testlerinde aracımızın tasarımının yeterli olduğu görülmüştür.

7.1.2 Dinamik Dayanım Testleri



Şekil 7.1.2 Aracımızın dış kabuğuna uyguladığımız dinamik dayanım testi



Summary

Goal Name	Unit	Value	Averaged Value	Minimum Value	Maximum Value	Progress [%]	Use In Convergence	Delta
GG Force (X) 1	[N]	641.699	640.265	638.679	641.699	0	No	0.370
GG Force (Y) 1	[N]	-1793.296	-1795.436	-1797.819	-1792.356	0	No	0.362
GG Force (Z) 1	[N]	-2537.322	-2537.087	-2537.636	-2536.513	0	No	0.211
GG Friction Force (X) 1	[N]	5.792	5.770	5.697	5.865	0	No	0.008
GG Friction Force (Y) 1	[N]	-9.008	-8.987	-9.086	-8.947	0	No	0.011
GG Friction Force (Z) 1	[N]	-8.887	-8.879	-8.946	-8.810	0	No	0.008

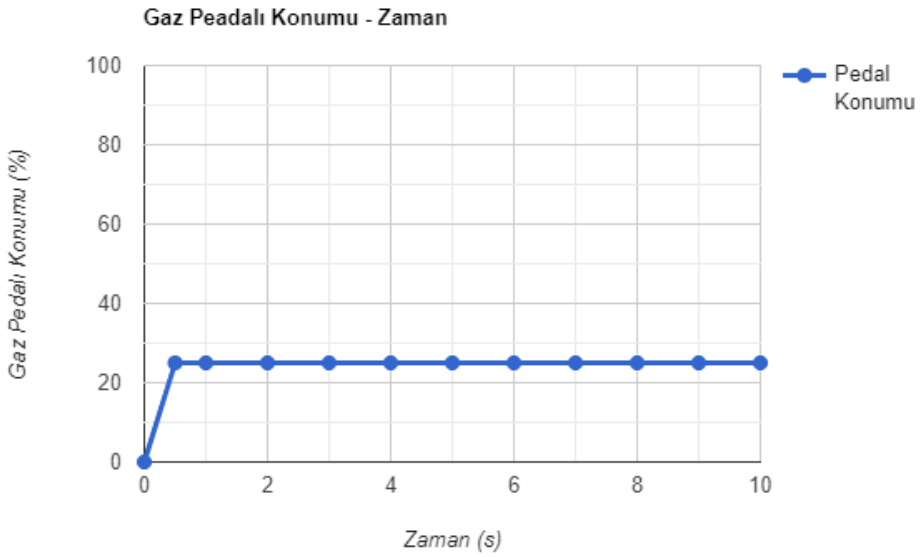
Şekil 7.1.2 Aracımıza Uyguladığımız Dinamik Dayanım Testi sonuçları

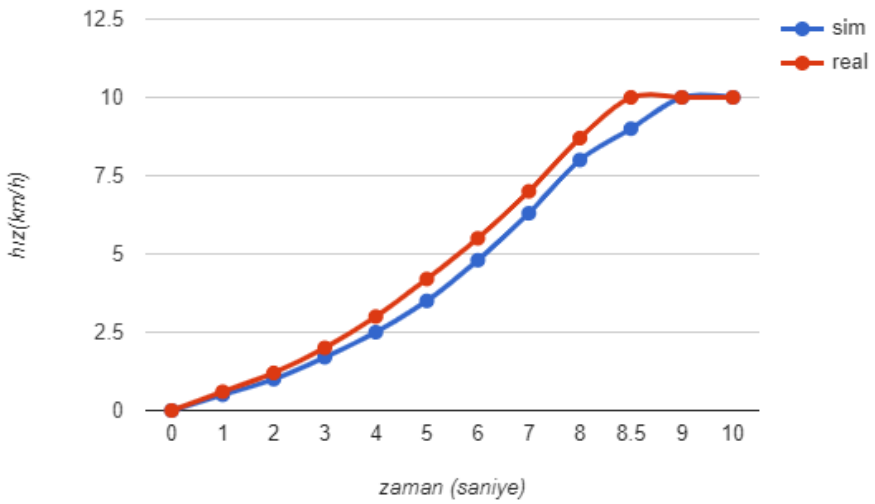
Aracımızın Şasesi 6000 serisi alüminyumdan yapılmıştır. Aracımızın şasesine uyguladığımız dinamik dayanım testlerinde şase yapısında kuvvete karşı kayma miktarının az olduğu görülmüştür. Aracımıza uyguladığımız dinamik dayanım testlerinden sonra aracımızın roll cage ve roll bar tasarımının yeterli olduğu görülmüştür.

7.1.3 İvme - Fren - Yönlendirme Grafikleri

İvmelenme

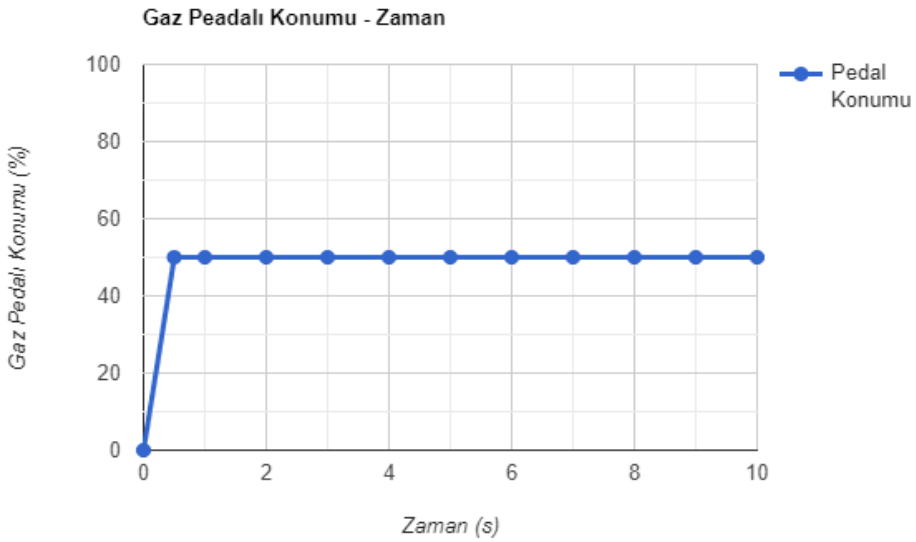
TEKNOFEST
HAVACILIK, UZAY VE TEKNOLOJİ FESTİVALİ

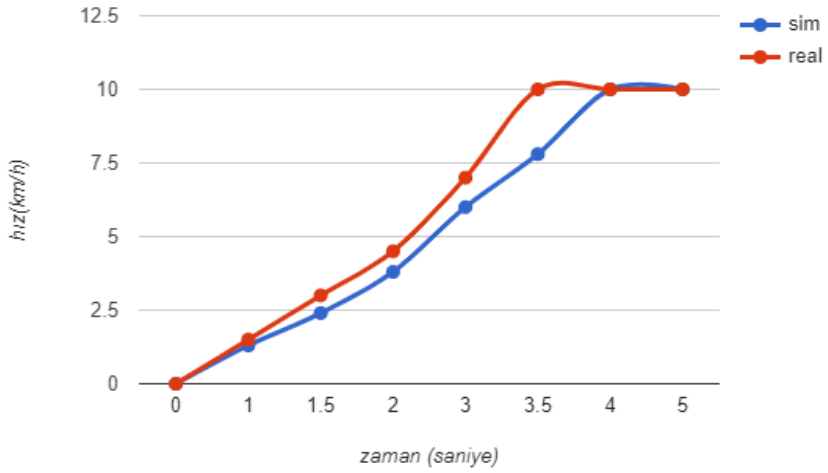




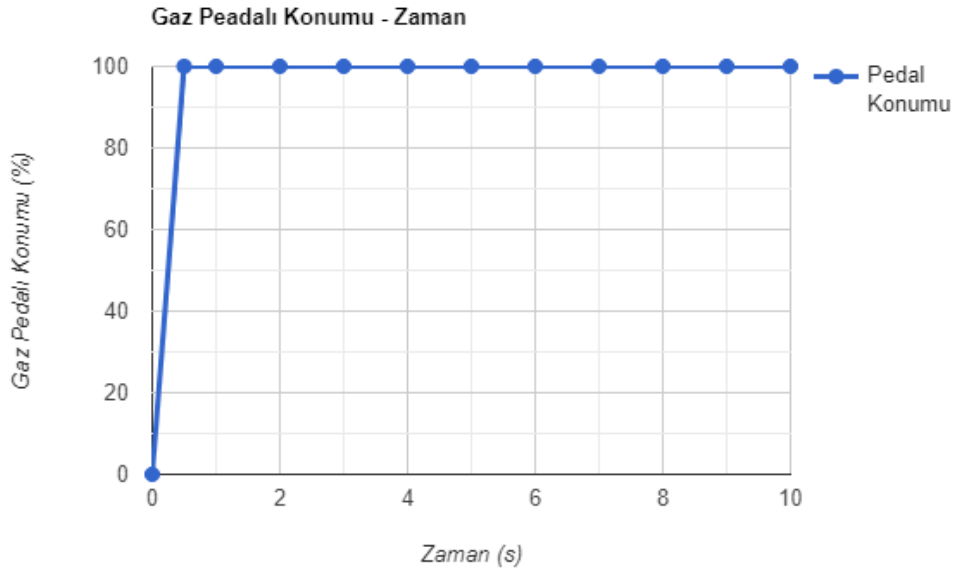
Şekil 7.1.3.1 Aracımızın Gaz Pedalı Değeri %25 iken Oluşan Hız-Zaman Grafikleri

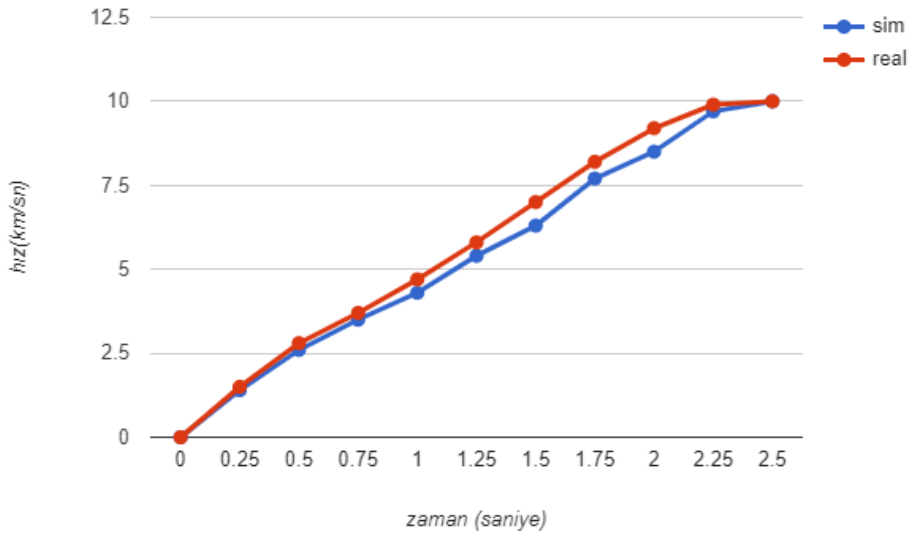
HAVACILIK, UZAY VE TEKNOLOJİ FESTİVALİ





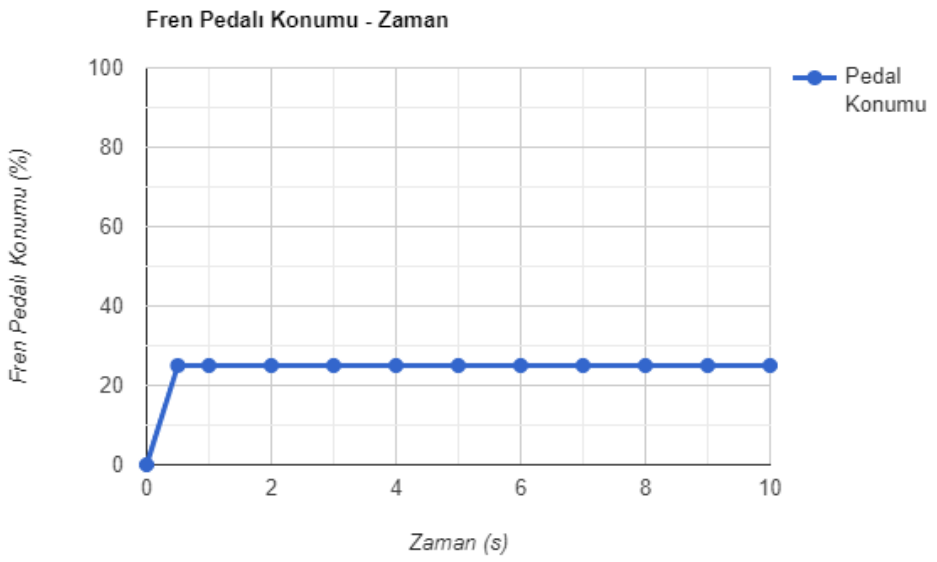
Şekil 7.1.3.2 Aracımızın Gaz Pedalı Değeri %50 iken Oluşan Hız-Zaman Grafikleri



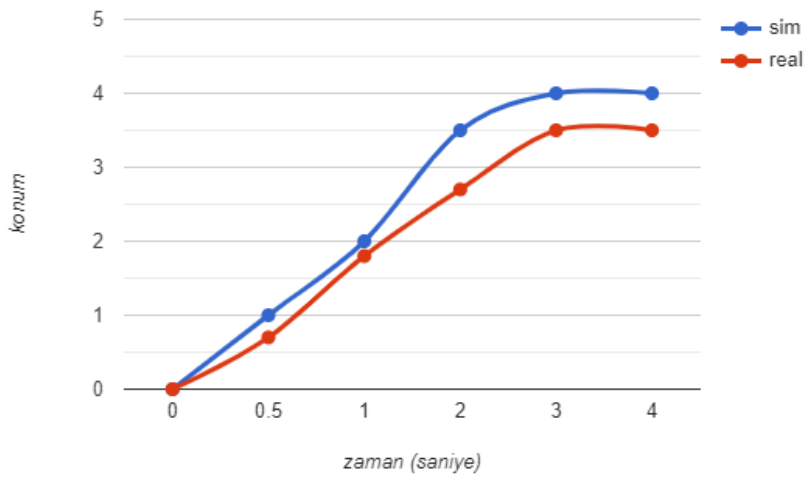


Şekil 7.1.3.3 Aracımızın Gaz Pedalı Değeri %100 iken Oluşan Hız-Zaman Grafikleri

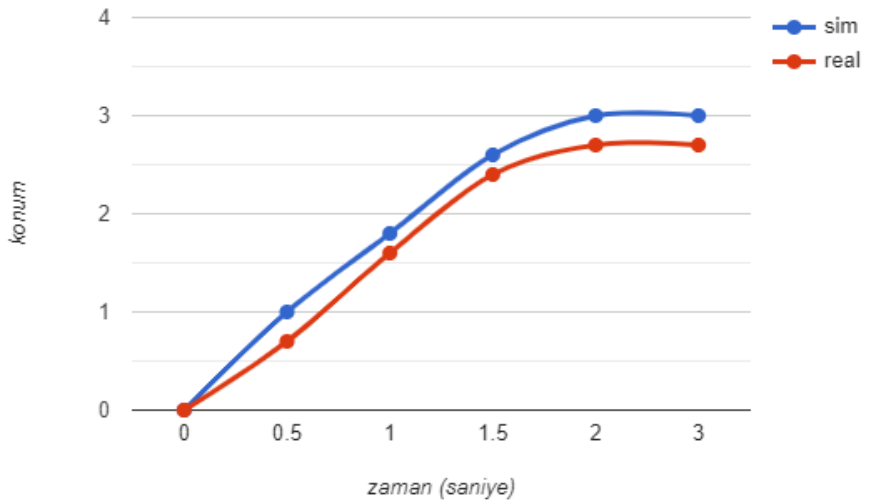
Aracımızın ivmelenme ve frenleme testleri ekspertizde yapılmıştır. Aracımız için yaptığımız ivmelenme simülasyonları ve gerçek hayat testlerinde verilerin aynı doğrultuda olduğu görülmüştür. Bu nedenle aracımızın ivmelenme konusunda bir iyileştirmeye ihtiyacı olmadığı görülmüştür. Aracın ivmelenmesini görebilmek için 0'dan 10 kilometreye hızlanmada düz yolda ne kadar süre aldığını ölçtük. Bu testleri trafiğe kapalı alanda gerçekleştirdik. Ekspertiz sonuçları ve yaptığımız testlerin sonuçlarının benzer olduğunu, meydana gelen ufak farklılıkların; sürtünme, rüzgar gibi dış ortam şartlarından kaynaklandığını gözlemledik.

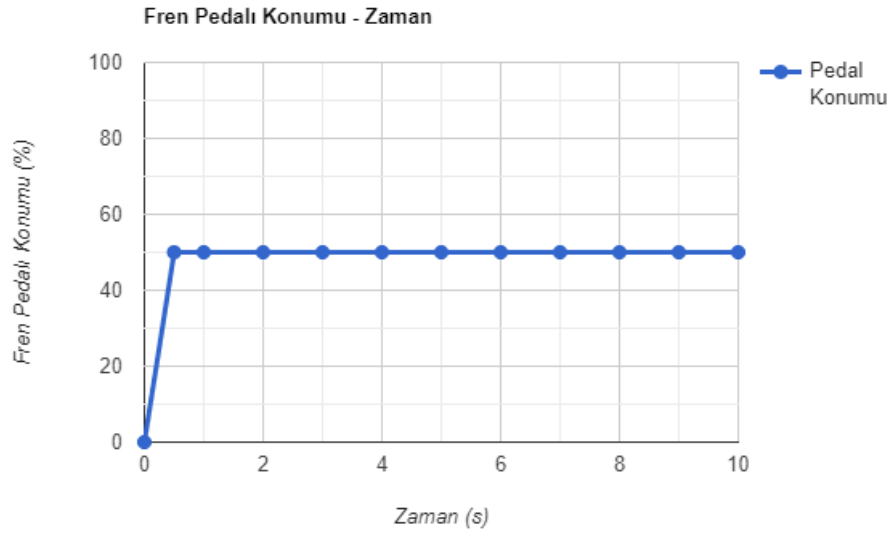


Frenleme

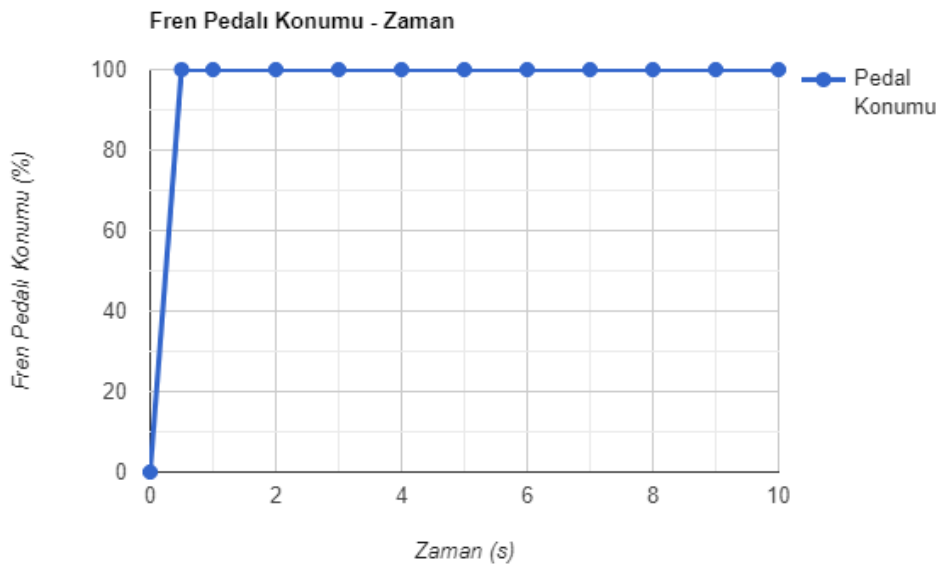


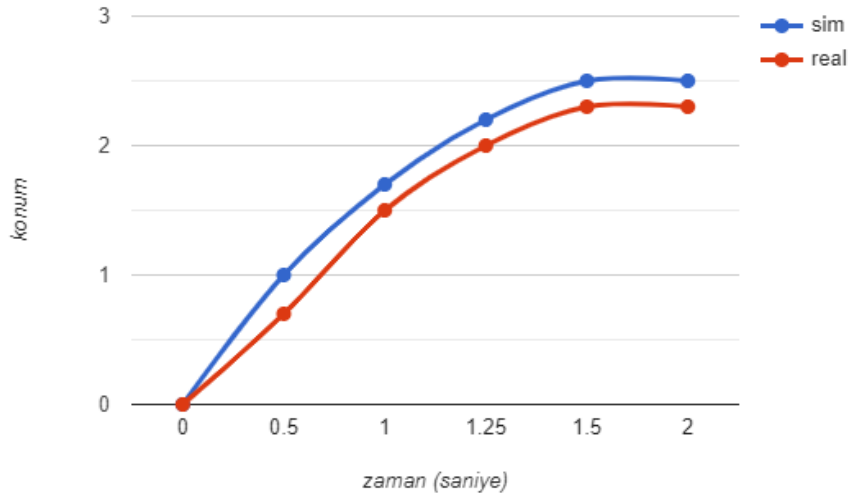
Şekil 7.1.3.4 Aracımızın Fren Pedalı Değeri %25 iken Oluşan Konum-Zaman Grafikleri





Şekil 7.1.3.5 Aracımızın Fren Pedalı Değeri %50 iken Oluşan Konum-Zaman Grafikleri



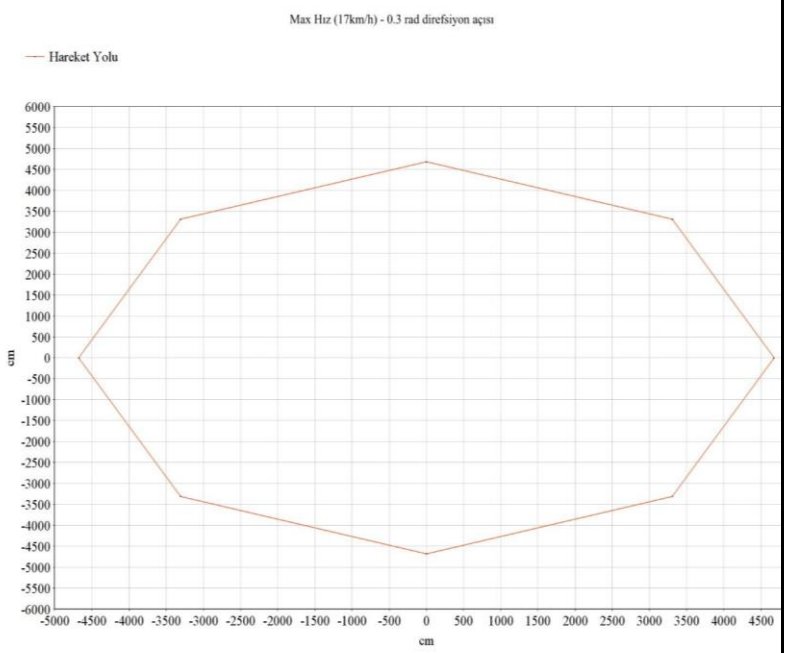


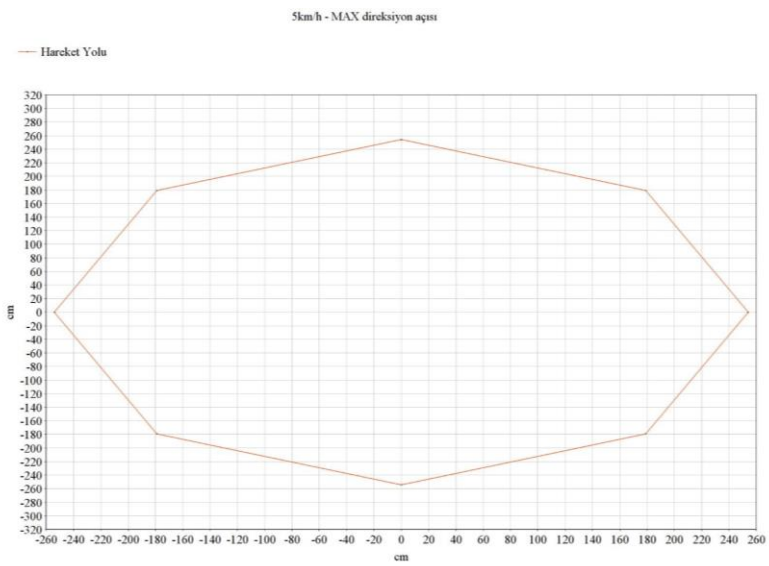
Şekil 7.1.3.6 Aracımızın Fren Pedalı Değeri %100 iken Oluşan Konum-Zaman Grafikleri

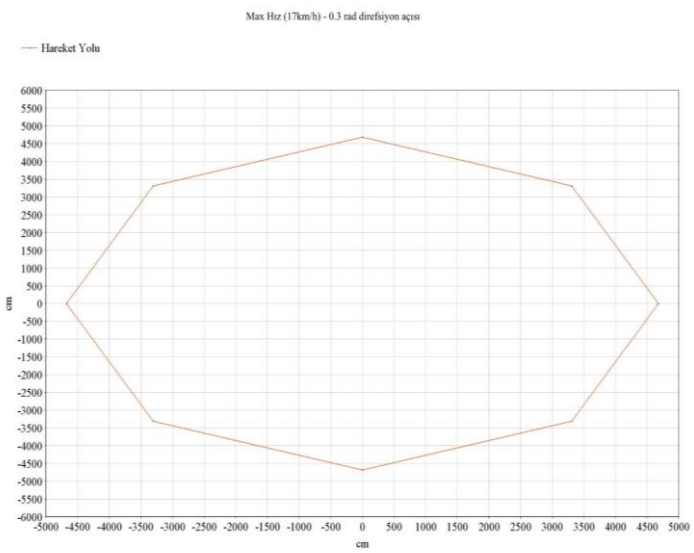
Aracımızın frenleme testi ekspertizde fren test cihazı rampasında yapılmıştır. Zemin üzerinde yer alan rulo disklerden oluşan bu sistemde araç disk üzerine yerleştirilir ve uzman kişiler tarafından sabitliği korunarak dijital ölçüm yapılır.

Aracımız için yaptığımız frenleme simülasyonları ve gerçek hayat testlerinde verilerin birbiriyle aynı doğrultuda olduğu görülmüştür. Yarışma alanında ve günlük kullanımda yeterli frenlemeyi sağladığı için frenleme konusunda aracımızın iyileştirmeye ihtiyacı olmadığı kararlaştırılmıştır.

Yönlendirme

Yönlendirme (Direksiyon) Sistemi Açı Değeri	Araç Hızı	Sonuç
0.3 rad ve -0.3 rad	5 km/s	 Aracın 5 km/s hız ile sürüşü esnasında yönlendirme girişi sabit -0.3 rad ve 0.3 rad değerleri ile aracın yatay yer değişiminin x-y ekseninde grafiğinin oluşturulması
Yönlendirme (Direksiyon) Sistemi Açı Değeri	Araç Hızı	Sonuç

		
Yönlendirme (Direksiyon) Sistemi Maksimum negatif ve pozitif Açı Değerleri	5 km/s	Aracın 5 km/s hız ile sürüşü esnasında aracın maksimum pozitif ve negatif yönlendirme girişlerine göre aracın yatay yer değişiminin x-y ekseninde grafiğinin oluşturulması
Yönlendirme (Direksiyon) Sistemi Açı Değeri	Araç Hızı	Sonuç

0.3 rad ve -0.3 rad	Maksimum araç hızı	 Araç maksimum hızında hareket halinde iken -0.3 rad ve 0.3 rad direksiyon açı değerleri için x-y ekseninde konum değişimi grafiğinin oluşturulması

Aracımızın dönüş grafiklerinin tek bir grafik olmasının sebebi aracımızın sağ ve sola eşit derecede dönüş kabiliyetinin olması ve bu nedenle negatif veya pozitif değer verildiğinde sadece grafikteki gidiş yönünün değişmesidir. Grafiklerde gidiş yönü belirtmesi bulunamadığından istenilen değerler tek bir grafikte gösterilmiştir. Aracımız Double Ackermann prensibi ile dönüşlerini gerçekleştirdiği için 2.6 m yarıçapında dönüşleri rahatlıkla alabilmiştir. Bu nedenle aracımızın yönlendirme sistemlerinde herhangi bir iyileştirme gereği görülmemiştir.

7.2 Elektronik Testler

7.2.1 Elektronik Devre Stres Testi

Değerlendirme Ölçütü: Baskı devre kartlarının tasarımında kullanılan elektronik komponentlerin tasarım limitlerinin (akım ve gerilim dayanım limitleri) aracın optimal çalışma koşullarına uyumlu olması gerekmektedir.

Test Senaryosu: Araç içerisinde kullanılan baskı devre kartlarına 1 saat boyunca kesintisiz akım verilecektir. Osiloskop ve şönt direnç yardımı ile yapılan ölçümlerden elde edilen akım değerlerinin istenen seviyede olup olmadığı ve herhangi bir anda akımda dalgalanmaların yaşanıp yaşanmadığı gözlemlenir.

Test Sonucu: Baskı devre kartlarının 1 saatlik süre sonunda işlevlerini yitirmeden çalıştığı gözlemlenmiştir.

7.2.2 Kablolu Haberleşme Testi

Değerlendirme Ölçütü: Tasarlanan kartların üzerindeki ve kartlar ile ana bilgisayar kartı

arasındaki kablolu haberleşme sinyal seviyeleri belirli değerlerin üzerinde bulunmalı ve veri sorunsuz iletilmelidir.

Test Senaryosu: Kablolu haberleşme protokolünün her bir sinyali osiloskop kullanılarak sinyalin temizliği test edilir. Ayrıca sürekli 1 ve sürekli 0 içeren datalar yollanarak veri bütünlüğünün korunup korunmadığı gözlemlenir.

Test Sonucu: Bu test ile kontrol kartının desteklediği wifi, Bluetooth, UART, CAN Bus, I2C protokollerinin sorunsuz çalıştığı gözlemlenmiştir.

7.2.3 Kablosuz Haberleşme Testi

Değerlendirme Ölçütü: Kablosuz haberleşme için mesafe ve coğrafi engellerin haberleşmeyi engelleyip engellemediğine bakılır.

Test Senaryosu: Alıcı ve vericilerin birbirlerinden uzak mesafelerde ve coğrafi engeller içeren yollanan ve alınan veri paketleri ve veri içerikleri karşılaştırılır. Ayrıca sürekli 1 ve sürekli 0 içeren datalar yollanarak veri bütünlüğünün korunup korunmadığı gözlemlenir.

Test Sonucu: Bu test ile kontrol kartının desteklediği RF, GSM protokollerinin sorunsuz çalıştığı gözlemlenmiştir.

7.2.4 Motor Sürücü Testi

Değerlendirme Ölçütü: Sürüş Kontrol Bilgisayarı kartı ile DC motor sürücüler arasındaki PWM sinyallerinin doğru bir şekilde iletildiği gözlemlenmelidir.

Test Senaryosu: Osiloskop yardımıyla PWM sinyallerinin dalga şekilleri izlenerek bu sinyallerin frekansı ve gerilim seviyeleri ölçülür. Bu testi multimetre ile yapmak mümkün değildir.

Test Sonucu: Sürüş Kontrol Bilgisayarı kartı ile oluşturulan PWM sinyalleri doğru bir şekilde DC motor sürücülere iletildiği gözlemlendi.

7.2.5 Yol Çizgisi Tanıma Testi

Değerlendirme Ölçütü: Değişik yol ve ışık ortamlarında yol çizgilerinin görüntü işleme bilgisayarı tarafından hatasız tespit edildiği gözlemlenmelidir.

Test Senaryosu: Dizüstü bilgisayar ve trafiğe çıkma izni olan bir araca takılan kamera ile ilgili yazılımın gerçek hayatta yol çizgilerini tanıma durumuna bakılır. Bu test sabah, öğle ve akşam tekrarlanır.

Test Sonucu: Yapılan testler sonucu yol çizgilerinin ilgili yazılım tarafından büyük bir başarı ile tanındığı görülmüştür.

7.2.6 Trafik Işığı Tanıma Testi

Değerlendirme Ölçütü: Değişik yol ve ışık ortamlarında trafik ışıklarının görüntü işleme bilgisayarı tarafından hatasız tespit edildiği gözlemlenmelidir.

Test Senaryosu: Dizüstü bilgisayar ve trafiğe çıkma izni olan bir araca takılan kamera ile ilgili yazılımın gerçek hayatta trafik ışıklarının tanıma durumuna bakılır. Bu test sabah, öğle ve akşam

tekrarlanır.

Test Sonucu: Yapılan testler sonucu trafik ışıklarının ilgili yazılım tarafından büyük bir başarı ile tanındığı görülmüştür.

7.2.7 Trafik Tabelası Tanıma Testi

Değerlendirme Ölçütü: Değişik yol ve ışık ortamlarında Trafik Tabelalarının görüntü işleme bilgisayarı tarafından hatasız tespit edildiği gözlemlenmelidir.

Test Senaryosu: Dizüstü bilgisayar ve trafiğe çıkma izni olan bir araca takılan kamera ile ilgili yazılımın gerçek hayatta Trafik Tabelalarını tanıma durumuna bakılır. Bu test sabah, öğle ve akşam tekrarlanır.

Test Sonucu: Yapılan testler sonucu Trafik Tabelalarının ilgili yazılım tarafından büyük bir başarı ile tanındığı görülmüştür.

7.2.8 Ön ve Arka Direksiyon Testi

Değerlendirme Ölçütü: Sürüş kontrol bilgisayarından yollanan komutlara göre ön ve arkada bulunan direksiyonların tekerlekleri istenen yöne uygun zamanda çevirdiği gözlemlenmelidir.

Test Senaryosu: Araç dururken ve güvenlik açısından elle itilirken direksiyon döndürme komutları yollanıp tekerleklerin istenen yöne dönüp dönmediği gözlemlenir.

Test Sonucu: Araç dururken ve güvenlik açısından elle itilirken sürüş kontrol bilgisayarından yollanan direksiyon döndürme komutlarına göre tekerleklerin istenen yöne döndüğü gözlemlendi.

7.2.9 Acil Durdurma Butonu Testi

Değerlendirme Ölçütü: Acil durdurma butonuna basıldığında araçtaki tüm enerji kesilmelidir.

Test Senaryosu: Araç dururken ve yavaş hızda giderken acil durum butonuna basılır.

Test Sonucu: Araç dururken ve yavaş hızda giderken acil durum butonuna basıldığında aracın tüm enerjisinin kesildiği gözlemlenmiştir.

7.2.10 Acil Durma Testi

Değerlendirme Ölçütü: Aracın önüne bir engel çıktığında araç otomatik olarak ani fren sistemini devreye sokmalı ve durmalıdır.

Test Senaryosu: Araç sabit hızında ilerlerken önüne araca hasar vermeyecek bir cisim konulur ve aracın cismi algıladığında çarpmadan durup durmadığı test edilir.

Test Sonucu: Test sonucunda aracın engeli algıladığında ani fren sistemini devreye sokup engele çarpmadan durarak tesi başarılı bir biçimde gerçekleştirdiği gözlemlenmiştir.

7.2.11 Yol Çizgisi Takip Testi

Değerlendirme Ölçütü: Aracın görüntü işleyerek şeritleri doğru bir biçimde tanıyıp tanıyamadığı ölçülür.

Test Senaryosu: Aracın yarışma koşullarında da belirtilen standart şeritli yolda sabit hızda

ilerleyebilme durumu gözlemlenir.

Test Sonucu: Aracın; şeritleri kameradan aldığı görüntüleri bulanıklaştırma, filtreleme vb. gibi işlemler uygulayarak ayırt ettiği ve sabit hızda %100 doğruluk payıyla örnek yolu tamamladığı test edilip onaylanmıştır.

7.2.12 Trafik Tabelasına Uyma Testi

Değerlendirme Ölçütü: Aracın görüntü işleyerek trafik tabelalarını doğru bir biçimde tanıyıp tanıyamadığı ölçülür.

Test Senaryosu: Araç yolda ilerlerken karşısına çıkan trafik tabelalarını doğruluk oranı yüksek bir biçimde algılayabilmelidir.

Test Sonucu: Araç, üstünde konumlanan tabela kamerasından aldığı görüntüleri yarışma şartnamesinde belirtilen trafik tabelalarına ve bunlara ek olarak oldukça yaygın birkaç tabelaya ait kendimiz tarafından oluşturulan ve Google Colab'de eğitilen veri setiyle eşleştirerek tabelaları yüksek doğruluk payıyla algılamayı başarmıştır.

7.2.13 Trafik Işığına Uyma Testi

Değerlendirme Ölçütü: Aracın görüntü işleyerek trafik ışıklarını doğru bir biçimde algılayıp uygun davranışlar gösterip göstermediği ölçülür.

Test Senaryosu: Araç yolda ilerlerken karşısına çıkan trafik ışıklarını algılayabilmeli ve durumlara uygun (kırmızı ışıkta durmalı, yeşil ışıkta geçmeli vb.) tepkiler verebilmelidir.

Test Sonucu: Araç, üstünde bulunan trafik ışıkları kamerası ile görüntüyü görüntü işleme bilgisayarına aktarır ve trafik ışığı tanıma algoritması devreye girerek aracın ışıklarda doğru tepkiler verdiği gözlemlenmiştir.

7.2.14 Durağa Girme ve Çıkma Testi

Değerlendirme Ölçütü: Aracın görüntü işleme tabela algılama ve şerit takibi yapma işlemlerinin hepsini kapsayan durak görevini başarıyla gerçekleştirip gerçekleştiremediği ölçülür.

Test Senaryosu: Araç tabela kamerasıyla algılayıp görüntü işleyerek okuduğu durak tabelasına göre hareket ederek durağa girmeli, şartnamede belirtildiği üzere en az 30 sn beklemeli ardından şerit takibi yaparak duraktan çıkmalıdır.

Test Sonucu: Aracın senaryoda belirtilen tüm aşamaları simülasyon ortamında başarıyla gerçekleştirdiği hakemler tarafından onaylanmıştır.

7.2.15 Viraj Alma Testi

Değerlendirme Ölçütü: Aracın standart yolda şerit takibi yaparak virajı alıp alamadığı ölçülür.

Test Senaryosu: Aracın kıvrılan yolları tespit edip virajı şerit takip ile alabilmesi beklenir.

Test Sonucu: Araç görüntü işleme ile şeritleri takip eder, Double Ackermann prensibine göre kıvrılan şeridin orta noktasını bulur, direksiyondan gelen yön ve açı verileri lineer cetvel aracıyla ölçülür, tekerleklerde bulunan encoderlardan gelen hız verileri ile sensör füzyona girer ve açısal

hıza göre virajı en etkin biçimde alır.

7.2.16 Şerit Değiştirme Testi

Değerlendirme Ölçütü: Görüntü işleme bilgisayarında şerit değiştirme komutu geldiğinde aracın göstereceği tepkiler ölçülür.

Test Senaryosu: Şerit değiştirilmesi gereken ekstrem bir durumda (aracın önüne engel çıkması, yolda göçük olması vb.) aracın bu komuta uyması beklenir.

Test Sonucu: Araç komutlara uyarak Şerit değiştirme algoritması basamaklarına uygun davranmıştır.

7.2.17 Park Etme Testi

Değerlendirme Ölçütü: Aracın görüntü işleme ile Park tabelasını algılayarak park yerinde durması gözlemlenir.

Test Senaryosu: Araç seyir halindeyken park tabelası ile karşılaştığında görüntü işleme bilgisayarı park etme komutunu verecek ve aracın park edip etmediği görülür.

Test Sonucu: Gerek simülasyon ortamı gerekse örnek pistte araç park etme algoritması basamaklarına uymuş ve görevi başarıyla yerine getirmiştir.

7.2.18 Lidar Testi

Değerlendirme Ölçütü: Aracın üzerinde bulunan lidar sensörü yardımıyla lazer ışınlarını kullanarak nesnelerin ölçüm aleti ile arasındaki uzaklığı ölçmesi ve algılaması ölçülür. Çevredeki nesne engelleri algılama ve kontrol bilgisayarına bildirme amaçlı kullanılacaktır.

Test Senaryosu: Araç seyir halindeyken engelle karşılaştığında nesneyi algılayıp ani fren sisteminin devreye girip girmediği gözlemlenecektir.

Test Sonucu: Hem yapılan simülasyonda hem de gerçek hayatta gerçekleştirilen testlerde lidar testinin başarılı bir şekilde sonuçlandığını ve engeller algılanarak donanıma haber verildiği gözlemlenmiştir.

7.2.19 Ultrasonik Sensör Testi

Değerlendirme Ölçütü: Aracın önünde bulunan belirli alanı gözlemleyebilmesi ,tehlike içeren durumlarda ani fren ile durup duramadığı ölçülür.

Test Senaryosu: Aracın ultrasonik sensörlerinin görüş alanına girecek biçimde ortaya bir engel konulabilir ve aracın duruma tepkisi ölçülebilir.

Test Sonucu: Aracın üzerinde konumlanan 3 ultrasonik sensör bulunmaktadır. Bu sensörler lidar ile birlikte aracın çevresini gözlemler ve olası engelleri sürüş kontrol bilgisayarına bildirerek can ve mal güvenliğini korumak amaçlı kullanılmış ve başarılı sonuçlar elde edilmiştir..

7.2.20 Sıcaklık Sensör Testi

Değerlendirme Ölçütü: Aracın olası bir aşırı ısınma durumunda bunu sıcaklık sensörleri ile

algılaması ve acil stop uyarısı vermesi beklenir.

Test Senaryosu: Aracın hayati öneme sahip bölümlerindeki aşırı ısınma kazalara ve patlamalara sebep olabileceğinden bu bölümü test etmek yerine teknik kontrol uygulanması ve bu kontrolün sonucunda aracın piste çıkıp çıkamayacağı kararı verilebilir.

Test Sonucu: Aracın üzerinde batarya paketinde bir adet, direksiyon motoru için iki adet ve Ana itki motoru sürücülere iki adet olmak üzere toplam 5 adet sıcaklık sensörü bulunur. Bu sensörler sayesinde aşırı ısınan yerleri tespit edip tehlike oluşmadan önlenebilecektir. Teknik kontrolden geçmesi beklenir.

Testlerde bahsedilen tüm algoritmalar yazılım mimarisinde detaylıca açıklanmıştır.

7.2.21 Batarya Yeterlilik Testi

Değerlendirme Ölçütü:Araç bataryasının yarış pistini tamamlamak için yeterli kapasiteye sahip olup olmadığı ölçülür.

Test Senaryosu: Araç maksimum tükenme sınırına dek devamlı olarak sürülür ve geçen süreye göre bir ortalama belirlenebilir.

Test Sonucu: Aracın bataryasını pisti bitirebilecek yeterlilikte olduğu yapılan test sonucu görülmüştür.

8. Referanslar

[1]https://ai-pool.com/a/s/yolov3-and-yolov4-in-object-detection

[2]https://arxiv.org/pdf/2004.10934v1.pdf

[3]https://medium.com/deep-learning-turkiye/makineler-g%C3%B6r%C3%BCyor-2-yolov4-ile-kendi-nesne-tespit-modelinizi-olu%C5%9Fturun-c73b81c8a4ea

[4]https://medium.com/deep-learning-turkiye/makineler-g%C3%B6r%C3%BCyor-1-nesne-tespit-modelleri-db036f6e1db6

[5]https://idemarket.com.tr/blog/yazilim/en-cok-kullanilan-yazilim-mimarileri/

[6]https://dergipark.org.tr/tr/download/article-file/614340

[7]https://dergipark.org.tr/tr/download/article-file/504790

[8]https://docplayer.biz.tr/168307480-Sensor-fuzyonu-algoritmalari-ile-acisal-konum-referans-sistemi-tasarimi-attitude-and-heading-reference-system-design-with-sensor-fusion-algorithms.html

[9]https://medium.com/operations-management-t%C3%BCrkiye/rcnn-yolo-opencv-ile-nesne-tan%C4%B1ma-fa3856e69fb3

[10]https://www.geeksforgeeks.org/difference-between-yolo-and-ssd/

[11]https://stanford.edu/~shervine/l/tr/teaching/cs-230/cheatsheet-convolutional-neural-networks

[12]https://www.veribilimiokulu.com/yapay-sinir-agiartificial-neural-network-nedir/

[13] https://github.com/roboflow-ai/darknet.git

[14]https://roboflow.com/

[15]https://colab.research.google.com/