

Derin Öğrenme Tabanlı Robotun İç Ortamlarda Yönlendirilmesi

Özge Çevik, 152120141020

Elif Genç, 152120151004

Ahmet Selim Küçükkara, 152120161059

Eskişehir Osmangazi Üniversitesi

Mühendislik Mimarlık Fakültesi

Lisans Yönetmeliği Uyarınca

Bilgisayar Mühendisliği Bölümü

LİSANS TEZİ

Olarak Hazırlanmıştır

Danışman: Dr. Öğr. Üyesi Şahin IŞIK

Ocak, 2020

İÇİNDEKİLER

A. PLANLAMA	4
A.1. Özet(Abstract) ve Anahtar Kelimeler(Keywords)	4
A.2. Bilgi Gereksinim Belirleme, Problemin Tanımlanması	5
A.2.1. Amaç	6
A.2.2. Konu ve Kapsam	6
A.2.3. Literatür Özeti	6
A.3. Beklenen Fayda	8
A.3.1. Özgün Değer	8
A.3.2. Yaygın Etki / Katma Değer	9
A.4. Yöntem	9
A.5. Araştırma Olanakları	11
A.6. Çalışma Takvimi	11
A.6.1. İş Zaman Çizelgesi	11
A.6.2. Kişi- İş Açıklaması	12
B. ANALİZ	12
B.1. Sistem Gereksinimlerini Ortaya Çıkarma Yöntem ve Teknikleri	12
B.1.1. Yazılı Basılı Belge İnceleme	12
B.1.2. Yüz Yüze Görüşme	14
B.1.3. Anket	14
B.1.4. Gözlem	17
B.1.5. Prototip ve Hızlı Uygulama Tasarım (Rapid Application Design- RAD)	18
B.1.6. Ortak Uygulama Tasarım (Joint Application Design-JAD)	18
B.1.7. Veri Akış Şemaları (Kavramsal, Mantıksal, Fiziksel Veri Akış Şeması)	18
B.1.7.1. Kavramsal Veri Akış Şeması	18
B.1.7.2. Mantıksal Veri Akış Şeması	19
B.1.7.3. Fiziksel Veri Akış Şeması	19
B.1.8. (OPS)Olay Tabloları, Durum Formları, İşlevsel Analiz Raporu	19
B.1.8.1. (OPS) Olay Tabloları	19
B.1.8.2. Durum Formları	20
B.1.8.3. İşlevsel Analiz Raporu	22
B.1.9. İş Akış Şeması	23
B.2. Sistem Gereksinimleri	23
B.2.1. İşlevsel Gereksinimler	23
B.2.2. Sistem ve Kullanıcı Arayüzleri ile İlgili Gereksinimler	24
B.2.3. Veriyle İlgili Gereksinimler	24
B.2.4. Kullanıcılar ve İnsan Faktörü Gereksinimleri, Güvenlik Gereksinimleri	25
B.2.5. Teknik ve Kaynak Gereksinimleri, Fiziksel Gereksinimler	26

C. TASARIM	27
C.1. Sistem Tasarımı	27
C.2. Kullanıcı ve Sistem Arayüzü Tasarımları	27
C.3. Test Tasarımı	29
C.3.1. Gereksinim Analizlerinden Teste Yönelik Hedeflerinin Detaylandırılması	29
C.3.2. Fonksiyonel Test Tasarımı	30
C.3.2.1. Birim(Unit) Test Tasarımı	30
C.3.2.2. Entegrasyon Testleri Tasarımı	30
C.3.2.3. Kapalı Kutu (Black-Box) Testleri Tasarımı	31
C.3.2.4. Beyaz Kutu (White-Box) Testleri Tasarımı	31
C.3.2.5. Kabul Testleri Tasarımı	31
C.3.3. Performans Test Tasarımı	31
C.4. Yazılım Tasarımı	31
C.4.1. Gereksinime Bağlı Tasarım Kalıp(lar)ı Seçimi	32
C.4.2. UML Kullanarak Tasarımı Diyagramları Oluşturma	32
C.5. Veri Tabanı Tasarımı	33
C.6. Donanım Tasarımı	33
C.6.1. Sistem Mekanik Bileşen Tasarımlarının Verilmesi	33
C.6.2. Elektronik Tasarım	34
C.6.3. Gömülü Yazılım Tasarımları	34
D. UYGULAMA	36
D.1. Geliştirilen Sistemin Sistem Tasarımlarını Karşılanmasının Değerlendirilmesi	36
D.2. Kullanıcı ve Sistem Ara yüzü Gerçeklemeleri	36
D.3. Gerçeklenen Testler	40
D.4. Yazılım/Veri Tabanı/Donanım Gerçeklemeleri	43
E. SONUÇ VE ÖNERİLER	43
KAYNAKÇA	44

Şekiller

Şekil 1. Donanım Mimarisi	9
Şekil 2. Yazılım Mimarisi	10
Şekil 3. Faster Region-Based Convolutional Neural Network	14
Şekil 4. Devre Şeması	18
Şekil 5. Kavramsal Veri Akış Şeması	19
Şekil 6. Mantıksal Veri Akış Şeması	19
Şekil 7. Fiziksel Veri Akış Şeması	19
Şekil 8. İşlevsel Analiz Raporu	22
Şekil 9. İş Akış Şeması	23
Şekil 10. Modeli Eğitmek İçin Kullanılan Nesneler	25
Şekil 11. Sistem Tasarımı	27
Şekil 12. IP Webcam Mobil Uygulama Arayüzü	28
Şekil 13. IP Webcam Mobil Uygulamasının Kamera Kullanım Ekranı	28
Şekil 14. Nesne Tanıma Uygulaması Örnek Arayüzü	29
Şekil 15. Use-Case Diyagramı	33
Şekil 16. Sistem Mekanik Bileşen Tasarım Şeması	34
Şekil 17. Arduino IDE Arayüzü	35
Şekil 18. Gömülü Yazılımın Akış Şeması	35
Şekil 19. IP Webcam Mobil Uygulaması Kamera Kullanım Ekranı	37
Şekil 20. Algılanan Nesnenin Kimlik Numarasını Gösteren Arayüz	37
Şekil 21. Nesne Tanıma Uygulaması Örnek Arayüzü-1	38
Şekil 22. Nesne Tanıma Uygulaması Örnek Arayüzü-2	38
Şekil 23. Nesne Tanıma Uygulaması Örnek Arayüzü-3	39
Şekil 24. Nesne Tanıma Uygulaması Örnek Arayüzü-4	39
Şekil 25. Nesne Tanıma Uygulaması Örnek Arayüzü-5	40
Şekil 26. Arduino'ya Aktarılan Veri	41

Tablolar

Tablo 1. Projede Kullanılacak Mevcut Makine-Teçhizat Listesi	11
Tablo 2. Projede Kullanılacak Mevcut Makine-Teçhizat Listesine Ait Bütçe	11
Tablo 3. Proje İş Zaman Çizelgesi	12
Tablo 4. Proje İş Paketleri İçin Görev Dağılımı Tablosu	12
Tablo 5. Olay Tablosu	20
Tablo 6. 1 No'lu İşleme Ait Durum Tablosu	20
Tablo 7. 2 No'lu İşleme Ait Durum Tablosu	20
Tablo 8. 3 No'lu İşleme Ait Durum Tablosu	21
Tablo 9. 4 No'lu İşleme Ait Durum Tablosu	21
Tablo 10. 5 No'lu İşleme Ait Durum Tablosu	21
Tablo 11. 6 No'lu İşleme Ait Durum Tablosu	22

A. PLANLAMA

A.1.Özet(Abstract) ve Anahtar Kelimeler(Keywords)

Teknolojideki gelişmelerle birlikte robotlar, günümüzde savunma ve endüstriyel sistemlerde, eğitim ve araştırma alanlarında, tarım alanlarında, tıp ve sağlık alanlarında olduğu gibi birçok alanda kullanılmaktadır. Derin öğrenme teknolojisinin geliştirilmesi ve kullanılmasıyla birlikte yapılan çalışmalar en sık kullanılan uygulamalardan biri olmuştur. Bu yüzden derin öğrenme ve robotlar yüksek katma değere sahip ileri teknoloji alanlarıdır.

Endüstri 4.0 çalışmaları kapsamında pek çok alanda robotlar kullanılmaktadır. Derin öğrenme kapsamında yapılan çalışmalar ile robotların yaptıkları iş çeşitliliği artacaktır. Bu yüzden derin öğrenme teknolojisi kapsamında yapılan nesne algılama çalışmasıyla robotların gelişmesi önemli bir yere sahip olmaktadır.

Derin öğrenme teknikleri kullanılarak önceden tanımlanmış geometrik şekilleri algılama ve şekillere göre robotun belirli bir iş parçasını yapması hedeflenmektedir. Projenin gerçekleştirilebilmesi için telefon üzerinde çalışan IP Webcam isimli uygulama ile kameradan alınan görüntüler bilgisayara ağ üzerinden aktarılacaktır ve görüntü üzerinde önceden eğitilmiş bir model çalışacaktır. Kullanılan model ve görüntü üzerinde çalışacak olan program Python dili ile yazılacaktır. Bilgisayarda çalışan Faster R-CNN kamera görüntü üzerinde algılanan belirli geometrik şekillere göre robota veri aktarımı yapacaktır. Bu veri aktarımı kablosuz veri aktarımı şeklinde olacaktır. Robot alınan verilere göre önceden belirlenmiş hareketleri yapacak şekilde tasarlanacaktır. Robotun bu hareketleri yapabilmesi için Arduino kartına yüklenen program, gelen veriye göre robottan istenilen işi yapmasını sağlayacaktır. Bu sistemi sağlayacak robot yapılarak gerçek zamanda test edilecektir.

Anahtar Kelimeler: Robot, Nesne Algılama, Endüstri, Derin Öğrenme

ABSTRACT

With the advances in technology, robots are used in many fields such as defense and industrial systems, education and research, agriculture, medicine and health. Studies with the development and use of deep learning technology have become one of the most frequently used applications. That's why deep learning and robots are advanced technology fields with high added value.

Robots are used in many fields within the scope of Industry 4.0 studies. With the work done within the scope of deep learning, the work diversity of the robots will increase. Therefore, the development of robots has an important place with the object detection study carried out within the scope of deep learning technology.

Using deep learning techniques, it is aimed to detect predefined geometric shapes and make the robot a specific workpiece according to the shapes. In order to realize the project, IP Webcam application running on the phone will transfer the images taken from the camera to the computer over the network and a pre-trained model will work on the image. The program that will work on the model and image used will be written in Python language. The Faster-RCNN camera running on the computer will transfer data to the robot according to the specific geometric shapes detected on the image. This data transmission will be in the form of wireless data transmission. The robot will be designed to perform predetermined movements according to the received data. In order for the robot to perform these movements, the program loaded on the Arduino card will enable the robot to perform the desired job according to the incoming data. The robot that will provide this system will be built and tested in real time.

Keywords: Robot, Object Detection, Industry, Deep Learning

A.2. Bilgi Gereksinim Belirleme, Problemin Tanımlanması

Teknolojideki gelişmelerle birlikte nesnelerin algılanması günümüzde yüksek sınıf güvenlik sistemlerinde, savunma sistemlerinde, hava araçlarında olduğu gibi birçok alanda kullanılmaktadır. Nesnelerin algılanması derin öğrenme teknolojilerin kullanılmasıyla birlikte en sık kullanılan uygulamalardan biri olmuştur. Bu yüzden bu alanda gerçekleştireceğimiz çalışmalar gelecek teknolojilerin robotlara entegre edilmesinde kolaylık sağlayacaktır.

Proje danışmanımız Dr. Öğr. Üyesi Şahin IŞIK ile görüşülüp yapmayı planladığımız projeyi geliştirmek ve gerçekleştirmek için proje kapsamında neler yapacağımıza karar verilmiştir. Bu proje için gereken temel bilgiler ve gereksinimler hakkında bilgi sahibi olunmuştur. Daha sonra bu bilgiler ışığında bu projeye benzer projelere, çalışmalara bakılmıştır ve kullanılan teknikler ve yöntemler incelenmiştir. Yapılan çalışmalar projemizde kullanacağımız yöntem ve teknikler hakkında ayrıntılı bilgi edinebilmemiz sağlanmıştır ve projemizi gerçekleştirmek için gereken adımlar belirlenmiştir.

Projemizi parçalara ayırarak izleyeceğimiz adımların neler olduğu belirlenmiştir. Projenin birinci kısmında takip edilmesi gereken nesnenin algılanması gerekmektedir. Nesne algılamak için geliştirilen yöntemler ve algoritmalar incelenmiştir. Daha sonra projenin ikinci kısmında yapılması gereken örnek robotlar ve robotların yapılma amaçları, amaçlarına uygun mekanizmaları, algoritmaları, mekanik ve devre tasarımları ve kullandıkları sensörler incelenmiştir.

Projemizle ilgili isterleri belirlemek ve projenin mekanik kısmı için farklı görüş ve önerileri almak, projeyi daha da geliştirebilmek adına elektrik ve elektronik mühendisliği bölümümüzdeki akademisyenimiz M. Mert ULUPINAR ile görüşülmüştür. Hocalarımızla yapılan görüşmeler

sonucunda projenin mekanik kısmı için nasıl bir mekanik tasarımın yapılması ve hangi sensörlerin kullanılmasının gerekliliği hakkında bilgiler alınmıştır.

A.2.1. Amaç

Projemizin amacı derin öğrenme ve yapay zekayı birleştirerek birçok işlemi gerçekleştirecek bir robot ortaya koymaktır. Birçok işlemden kastımız ise robotun nesneyi tanımasıyla birlikte nesneye yüklediğimiz harekettir (örnek olarak oval sağa dön). Bu robotun kullanılacağı alana göre nesne tanımalarının yeterli olmasıyla birlikte birden fazla alanda rahatlıkla kullanılmasını amaçlanmaktadır. Belirli komutlarla hareket etmesi nedeniyle fabrika gibi alanlarda kullanılması amaçlanmaktadır. Örneğin fabrikalarda belirli birimlerde taşıma işlemi için insan gücü kullanılmaktadır ve yapılan iş sadece malzemelerin oradan alınıp başka bir alana götürülmesi işlemidir. İnsan gücüne gerek duyulmadan yaptığımız robotun belirli nesnelere göre bu işlemi gerçekleştirebilmesi hedeflenmektedir. Bunları amaçlarken ilk olarak küçük bir robotla işe başlanmıştır. Nesneleri tanıması için kullandığımız resimlerle eğitilmesi gerekmektedir. Eğitildikten sonra öğrendiği nesneler sayesinde istediğimiz alanlarda gösterdiğimiz nesnelerle hareket etmesi sağlanacaktır. Bu işlemden sonra belirli bir alanda nesneler sayesinde ilerleyebilecektir. Bunun için ufak bir parkur gerçekleştirilecektir.

A.2.2. Konu ve Kapsam

Projenin konusu önceden tanımlanmış geometrik şekilleri algılama ve şekillere göre robotun belirli bir iş parçasını yapmasını gerçekleştirmesidir. Nesnelerin algılanması günümüzde yüksek sınıf güvenlik sistemlerinde, savunma sistemlerinde, hava araçlarında olduğu gibi birçok alanda kullanılmaktadır. Nesnelerin algılanması en sık kullanılan uygulamalardan biri olmuştur. Bu yüzden bu alanda gerçekleştireceğimiz çalışmalar gelecek teknolojilerin robotlara entegre edilmesinde kolaylık sağlayacaktır.

Projenin amacı da birden fazla alanda insan gücüne gerek duyulmadan belirlenmiş ve sabit işleri robotların nesne algılamasıyla birlikte gerçekleştirmesidir. Projenin amacı doğrultusunda ilgili sektördeki kuruluşların belirli taşıma, depolama vb. işlerin zaman kaybı olmadan daha hızlı bir şekilde ve insan gücüne gerek kalmadan düzgün olarak yapılacağı düşünülmektedir.

A.2.3. Literatür Özeti

Yapay zekâ ve alt dalları ile ilgilenen araştırmacıların hedefi insan düşünce yapısını ve karar verme yetisini modellemektir. Bu amaçla 1943'te McCulloch-Pitts tarafından insan sinir sisteminden esinlenerek beyin fonksiyonlarının işleyişinin mantıksal olarak hesaplayan bir model ortaya konulmuştur. (McCulloch and Pitts 1943)

Günümüzde var olan mevcut sistemler kısaca açıklanacak olursa, iç ve dış ortamlarda konumlandırma ve yönlendirme ile dönüşümlü sinir ağlarının kullanımı problemlerine odaklanılmıştır. Sistem sensörler veya video kameraları yardımıyla görüntüyü işler ve tek veya çok katmanlı dönüşümlü sinir ağları kullanarak verileri analiz eder.

Derin öğrenme tabanlı iç mekân drone yarışları için algılama, rehberlik ve navigasyon, bu projede görüntülerin işlenmesi için NVIDIA Jetson TX2 gömülü bilgisayar kullanılıyor ve algılama sonuçları kullanılarak “line-of-sight” algoritması uygulanıyor. Böylece drone’ların kaza yapmadan geçiş noktalarına varması sağlanıyor. Sunggoo Jung tarafından geliştirilmiştir [1].

Derin öğrenme tabanlı monoküler resimlerin görsel ve metinsel analizini kullanarak perakende incelemesi ve envanter bilgilendirmesi için mobil robot, bu projede iç ortam UWB konumlandırması kullanarak rafları otonom olarak görüntüleyen bir mobil robot kullanılıyor. İki eğitilmiş dönüşümlü sinir ağlarından elde edilen metinsel ve görsel özelliklere dayanarak ürünün türü bir makine öğrenme sınıflandırıcı tarafından tanımlanır. Robot stokta ürün yok (SOOS) durumlarının otomatik olarak algılanmasını kolaylaştırmak için tasarlanmıştır. Marina Paolanti tarafından geliştirilmiştir [2].

Bilinmeyen iç ortamlarda trafik işaretleri kullanan Nrobotic mobil robot navigasyonu, bu projede algoritma, sonardan veya robota monte edilmiş kızılötesi sensörlerden gelen verileri ve robotun donatıldığı video kameradan gelen verileri kullanır. Video kamera yardımıyla trafik işaretlerini algılar ve sınıflar ve hedef noktaya güvenle ve kısa sürede ulaşır. Algoritma, trafik işaretlerini sınıflandırmak için 6 katmanlı bir dönüşümlü sinir ağı kullanır. Böylece robotun bilinmeyen bir ortamda trafik işaretleriyle doğru şekilde hareket etmesi sağlanıyor. Constantin Purcaru tarafından geliştirilmiştir [3].

Önceden eğitilmiş dönüşümlü sinir ağı kullanarak iç ortamlarda nesne tanıma, bu projede iç ortam veri seti ve özel video kareleriyle çevrimdışı bir dönüşümlü sinir ağı eğitiliyor. Sonra girdi olarak verilen video karelere ayrılıp, ilgilenilen bölgeyi (ROI) çıkarmak için seçici bir arama işlemi yapılıyor. Çıkarılan ilgi bölgesi (ROI) önceden eğitilmiş olan dönüşümlü sinir ağı kullanılarak sınıflandırılır ve en yakın görüntüler algılama füzyonu kullanılarak işlenir. Böylece iç ortamda nesnelerin daha verimli tespit edilmesi sağlanıyor. Xintao Ding tarafından geliştirilmiştir [4].

Derin dönüşümlü sinir ağı kullanarak yol çatlağı tespiti, bu projede toplanan görüntülerdeki her çatlağı sınıflandırmak için denetimli bir derin dönüşümlü sinir ağı eğitiliyor. Düşük maliyetli bir akıllı telefonla toplanan, 3264x2448 boyutundaki 500 görüntüden oluşan bir veri kümesinde yapılan nicel değerlendirme, önerilen derin öğrenme çerçevesine sahip öğrenilen derin özelliklerin, mevcut el ile elde edilen özelliklerle karşılaştırıldığında üstün çatlak algılama performansı sağladığını gösteriyor. Böylece çatlakların homojen olmama yoğunluğundan ve arka planın karmaşıklığından kaynaklanan zorluklarla mücadelede önemli bir rol oynuyor. Lei Zhang tarafından geliştirilmiştir [5].

Mobil robotun yeniden konumlandırılması için dönüşümlü sinir ağı tabanlı sensörler, bu projede ilk önce bir monoküler kameradan RGB görüntülerini girdi olarak alan ve robot pozunu için regresyon yapan bir dönüşümlü sinir ağını eğitir. Ardından, eğitilmiş dönüşümlü sinir ağının yer değiştirme çıktısını, robot konumlandırılması için bir Genişletilmiş Kalman Filtresi (EKF) içine dahil

eder. Böylece düşük maliyetli sensörlere sahip ve hesaplama kabiliyeti düşük mobil robotlar için doğru bir yeniden konumlandırma yapısıdır. Harsh Sinha tarafından geliştirilmiştir [6].

Parmak izi-görüntü ve derin öğrenme tabanlı iç mekân konumlandırma, bu projede Wi-Fi ve manyetik alan parmak izlerini kullanarak yeni bir karma konum görüntüsü tasarlanıyor ve ardından parmak izi görüntülerinin konumlarını sınıflandırmak için bir dönüşümlü sinir ağı kullanılıyor. Projede kullanılan konumlandırma yönteminin farklı akıllı telefon yönelimleri, kullanıcıları ve kullanım şekilleri altında yaklaşık 1 metrelik bir kesinliğe ulaştığını ortaya koyuyor. Wenhua Shao tarafından geliştirilmiştir [7].

Literatür taramasında karşılaşılan projelerin birçoğu görüntü işleme tabanlı ve dönüşümlü sinir ağları kullanan projelerdir. Projelerin çoğunluğunda sensörler, video kameralar veya her ikisi de kullanılarak görüntüler elde edilmekte ve elde edilen görüntüler dönüşümlü sinir ağları kullanılarak verilere dönüştürülmektedir.

A.3. Beklenen Fayda

Bu bölümde özgün değer, yaygın değer başlıkları altında beklenen faydadan bahsedilmiştir.

A.3.1.Özgün Değer

Nesne tanıma modeli olarak, istenilen nesneleri hızlı bir şekilde tanımasını sağlamak için Faster R-CNN kullanılacaktır. Nesne tanıma programı Python dilinde yazılacaktır. Eğitim tamamlandıktan sonra artık robotumuz bilgisayardan gönderilecek verilere göre hareket edecektir. Telefon kamerasından alınacak olan anlık görüntü IP Webcam isimli program ile http protokolü kullanılarak bilgisayara aktarılacaktır. Modelin tanınması gereken nesneleri hızlı ve doğru bir şekilde tanınması gerekmektedir. Tanınan nesnelerin kimlik numaraları alınacaktır ve bluetooth modülü ile birlikte Arduino' ya aktarılacaktır. Bluetooth modülü olarak HC 05 kullanılacaktır. Bluetooth modülü MAC adresi ve port numarası üzerinden iletişim sağlamaktadır. Robot gelen verilere göre motor sürücü kartını kontrol edecektir, bunu yapabilmesi için motor sürücü kartı veri kabloları aracılığıyla Arduino' ya bağlanacaktır. Kullanılan motor sürücü kartı ile 6V 250 RPM' dir. Motor sürücü kartı güç kabloları aracılığıyla bir kaynağa bağlanacaktır. Robotun hareketi sağlandıktan sonra robotun önüne çıkacak olan engellere çarpıpması gerektiği için ultrasonik mesafe sensörü kullanılacaktır.

Robotumuzun rahat hareket etmesi amacıyla kablo olmadan bilgisayar ile haberleşmesi sağlanacaktır. Bu işlemi gerçekleştirdiğimizde verilen herhangi bir parkurda kendisi rahatlık ile hareket edecektir. Herhangi bir insan gücüne ihtiyaç duyulmayacaktır. Ama asıl amacımız robotumuzun nesne tanınması ile hareket çeşitliliğinin artmasını sağlamaktır. İlk olarak temel hareketleri yapabilmesi sağlanacaktır.

Arduino ile yapılan benzer çalışmalara bakıldığında bu projeden farklı olarak robot ile iletişim kablolar aracılığıyla yapılmaktadır. Arduino' nun hafızası görüntü işleme için gerekenden oldukça

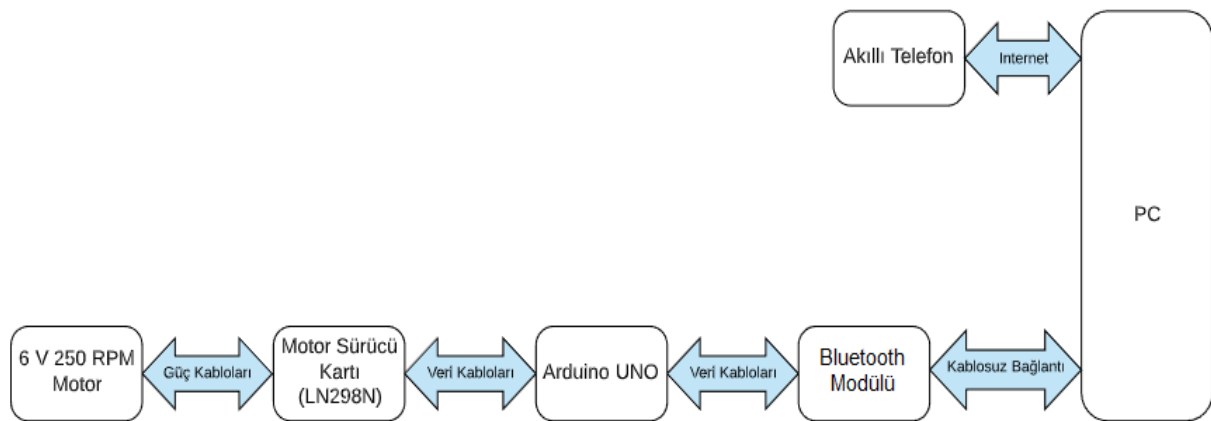
küçük olduğu için benzer projelerde Arduino' ya takılan kameralar veya harici kameralardan görüntüler kablo ile alınmaktadır. Bu da robotu bilgisayara bağımlı hale getirmekte ve hareket alanını kısıtlamaktadır. Yine benzer projelerde görüntü işlemenin sonuçları kablo ile robota aktarılmakta ve bu da robotu bilgisayara bağımlı hale getirmektedir. Bu projede amaçlanan robotun diğer bileşenlerden bağımsız olarak hareket edebilmesi için farklı yöntemler kullanılmıştır. Kullanılan yöntemler ise görüntüler IP Webcam uygulaması ile alınmakta olup, anlık olarak bilgisayara gönderilmektedir. Anlık alınan görüntüler işlendikten sonra sonuçları bluetooth modülü ile robota gönderilmektedir. Böylece hem görüntü almak için kullanılan kabloya hem de robota görüntü işlemenin sonuçlarını göndermek için kullanılan kabloya ihtiyaç duyulmamış ve robot diğer projelere nazaran daha bağımsız hale getirilmiştir.

A.3.2. Yaygın Etki/Katma Değer

Projemizin amacı derin öğrenme tekniğini kullanarak robotun belirli işlevleri gerçekleştirmesidir. Bunu yapabilmek için ilk olarak nesneleri belirlememiz gerekmektedir. Nesneler belirlendikten sonra nesneleri kullanarak modelimizi eğitilmelidir. Eğitim tamamlandıktan sonra robotun üstünde bulunan telefon üzerinde çalışan IP Webcam isimli uygulama ile kameradan alınan anlık görüntüler bilgisayara internet üzerinden aktarılacaktır. Görüntüyü algılayan ise önceden eğittiğimiz modeldir. Bilgisayarda çalışan Faster R-CNN anlık görüntü üzerinde algılanan nesnelere göre robota kablosuz olarak veri aktarımında bulunacaktır.

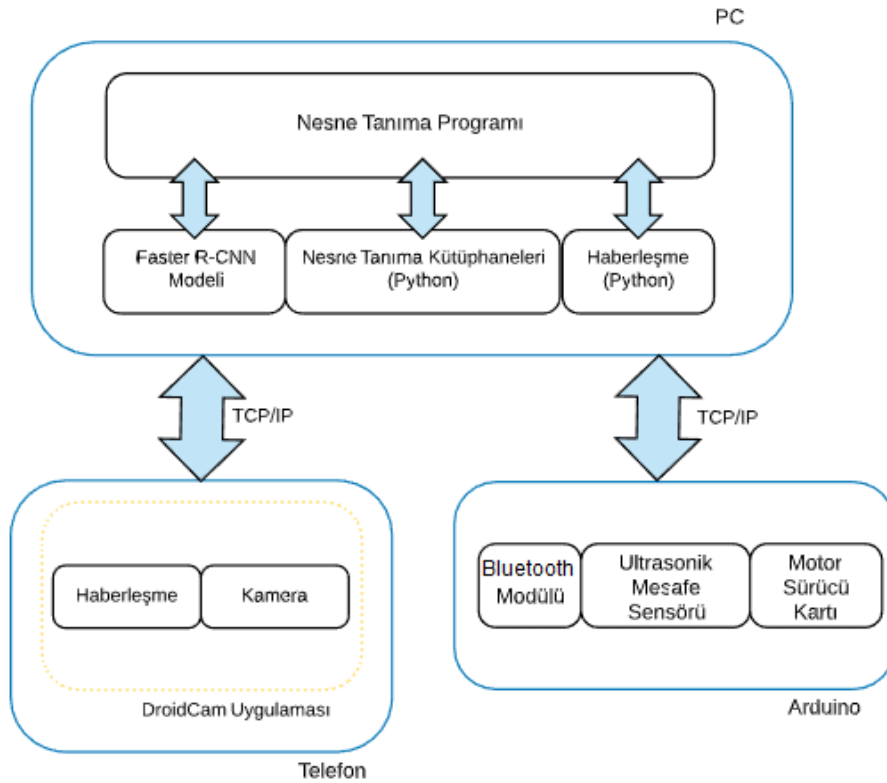
Robotun bu veri aktarımında aldığı hareket bilgisini gerçekleştirebilmesi için Arduino kartına yazdığımız program yüklenecektir. Bu sayede robotların herhangi bir kullanıcıya ihtiyaç duymadan kendine verilen görevleri yerine getirmesi kullanıldığı alanda insan gücünden tasarruf sağlayacaktır. Örnek vermek gerekirse yük taşıyan robotların sürücüsüz hareket etmesi sanayi alanında büyük önem taşımaktadır. Belirli bir rotada yapılan işler insan gücü olmadan yapılabilecektir.

A.4. Yöntem



Şekil 1. Donanım Mimarisi

Projemizin donanım mimarisi Şekil 1’ te gösterilmektedir. Öncelikle kişisel bilgisayar ve akıllı telefon kablosuz bir bağlantı (internet) üzerinden veri alışverişi yapacaktır. Bilgisayar ile bluetooth modülü eşleştirilmiştir ve kablosuz bir şekilde veri alışverişinde bulunacaklardır. Bluetooth modülü olarak HC 05 kullanılacaktır. Bluetooth modülü Arduino’ ya kablolar ile bağlanmaktadır. Veriler bluetooth modülünden Arduino’ ya bu kablolar üzerinden aktarılacaktır. Arduino bluetooth modülünden gelen veriye göre motor sürücü kartına veri aktarımı yapacaktır. Bu veri aktarımı kablolar üzerinden sağlanacaktır. Motor sürücü kartı ile 6V 250 RPM motorlar güç kabloları ile iletişim kurulacaktır. Motorlara gelen güç ile tekerlekler dönmeye başlayacaktır.



Şekil 2. Yazılım Mimarisi

Şekil 2’ te projenin yazılım mimarisi gösterilmektedir. Telefonda koşan IP Webcam isimli mobil uygulama sayesinde telefonun kamerasından alınan anlık görüntüleri bilgisayara aktarmak için bilgisayar ile telefon arasında haberleşmeyi sağlar. Daha sonra bilgisayar üzerinde koşan nesne tanıma programı eğitilen RCNN modelini, nesne tanıma kütüphanelerini ve haberleşme kütüphanesini kullanarak nesneleri tanıma yapar ve robotla haberleşmesini sağlamaktadır. Robota gönderilen veriler ile robot daha önceden yazılan ve Arduino içerisine aktarılan program sayesinde hareketlerini gerçekleştirir. Nesne tanıma programı Python dili ile yazılacaktır. Modelde ve nesne tanıyan program içerisinde kullanılan Python kütüphaneleri; pillow, lxml, jupyter, matplotlib, pandas, opencv-python, scikit-learn, scipy’ dir.

A.5. Araştırma Olanakları

Projede kullanılacak mevcut makine-teçhizat listesi Tablo 1’ de, projede kullanılacak mevcut makine-teçhizat listesine ait bütçe ise Tablo 2’ de gösterilmiştir.

Projede Kullanılacak Mevcut Makine – Teçhizat Listesi	
Adı/Modeli	Projede Kullanım Amacı
Kişisel Bilgisayar(PC) / Casper Nirvana C710 Notebook	Faster R-CNN isimli eğitilen modeli telefonda gelen anlık kamera görüntüleri üzerinde çalıştırmaktır..
Akıllı Telefon / Xiaomi Redmi Note 7	DroidCam adlı uygulama ile anlık kamera görüntülerinin alınmasıdır.
Arduino UNO	Robotun belirli hareketleri yapmasını ve sensörlerden veri toplayarak nesnelere çarpmasını sağlar.

Tablo 1. Projede Kullanılacak Mevcut Makine-Teçhizat Listesi

Malzeme Adı/Modeli	Fiyat (TL)
Robot Şasesi	19,9
Robot Tekerleği ve DC Motor Seti (x4)	103,96
L298N Voltaj Regülatörlü Çift Motor Sürücü Kartı	13,59
HC-SR04 Arduino Ultrasonik Mesafe Sensörü	5,57
HC 05 Bluetooth Modülü	23,62
Kişisel Bilgisayar(PC)/Casper Nirvana C710	2899
Akıllı Telefon/Xiaomi Redmi Note 7	1899,01
Breadbord	5,41
Telefon Tutacağı	12,99
Pil Yuvası	6,13
Pil	8,45
Arduino UNO	133,21
TOPLAM	5130,84

Tablo 2. Projede Kullanılacak Mevcut Makine-Teçhizat Listesine Ait Bütçe

A.6. Çalışma Takvimi

A.6.1. İş zaman Çizelgesi:

İş Paketi Adı/Tanımı	Haftalar													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Problem Tanımlanması														
Fizibilite Raporlarının Hazırlanması														
Proje Zaman Çizelgesinin Hazırlanması														
Görev Paylaşımının Yapılması														
Bilgi Toplanması														
Sistem Gereksinimlerinin Belirlenmesi														
Yüz Yüze Görüşmeler														
Anket Hazırlanması														
Fotoğrafları Etiketlemek														
Modeli Eğitmek														
Modelin Test Edilmesi														
Cihazlar Arası Haberleşme ve Kontrol Testi														
Robota Özel Parkurun Yapılması														
Robot Hareketlerinin Geliştirilmesi ve Testi														

Tablo 3. Proje İş Zaman Çizelgesi

A.6.2. Kişi- İş Açıklaması

Görev Adı	Kişiler
Problem Tanımlanması	E.G, Ö.Ç, A.S.K,
Fizibilite Raporlarının Hazırlanması	E.G, Ö.Ç
Proje Zaman Çizelgesinin Hazırlanması	E.G
Görev Paylaşımının Yapılması	E.G, Ö.Ç
Bilgi Toplanması	E.G, Ö.Ç
Sistem Gereksinimlerinin Belirlenmesi	E.G, Ö.Ç
Yüz Yüze Görüşmeler	E.G, Ö.Ç, A.S.K,
Anket Hazırlanması ve Uygulanması	E.G, Ö.Ç
Fotoğrafları Etiketlemek	E.G, Ö.Ç, A.S.K,
Modeli Eğitmek	E.G
Modelin Test Edilmesi	E.G
Cihazlar Arası Haberleşme ve Kontrol Testi	E.G, Ö.Ç, A.S.K,
Robota Özel Parkurun Yapılması	E.G, Ö.Ç, A.S.K,
Robot Hareketlerinin Geliştirilmesi ve Testi	E.G, Ö.Ç, A.S.K,
Elif GENÇ = E.G	
Özge ÇEVİK = Ö.Ç	
Ahmet Selim KÜÇÜKKARA = A.S.K	

Tablo 4. Proje İş Paketleri için Görev Dağılımı Tablosu

B.ANALİZ

Projemizin sistem gereksinimlerini belirleyebilmek amacıyla yazılı basılı belge inceleme, yüz yüze görüşme, anket formlarının oluşturulması ve anketlerin uygulanması, gözlem gibi yöntem ve teknikler kullanılmıştır. Bahsi geçen yöntemlerin kullanılması ile projemizin sistem gereksinimleri belirlenmiştir.

B.1. Sistem Gereksinimlerini Ortaya Çıkarma Yöntem ve Teknikleri

B.1.1. Yazılı Basılı Belge İncelenmesi

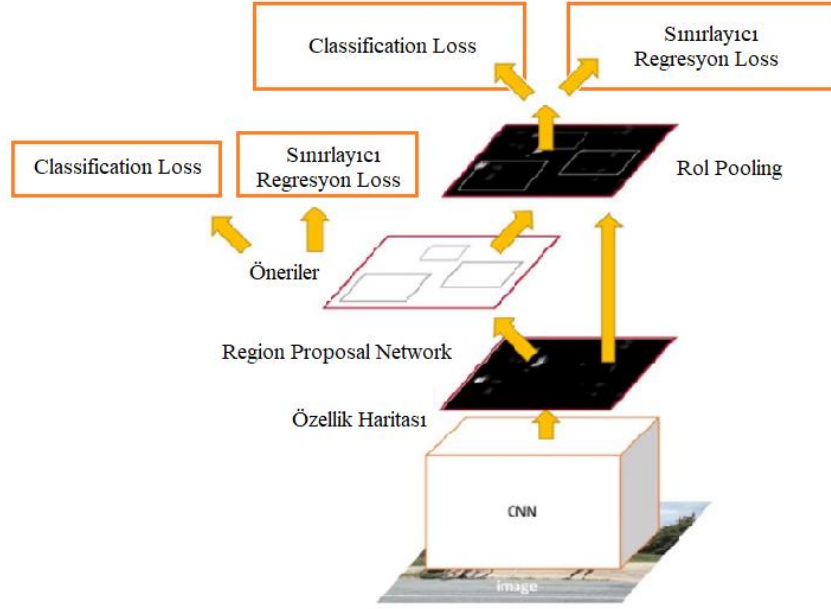
Projemizde kullanacağımız teknolojiler hakkında bilgi sahibi olmak için derin öğrenme, derin öğrenmenin kullanım alanları, derin öğrenme yöntemleri ve teknikleri hakkında bilgi edinilmiştir.

Yapay zeka ve alt dalları ile ilgilenen araştırmacıların hedefi, hayvan zekasını taklit edebilecek sistemler tasarlamaktır. Mühendislik uygulamalarında hayvan zekasının yer alması makine öğrenimi olarak bilinmektedir. Derin öğrenme ise aslında bir makine öğrenimidir. Derin öğrenme de hayvan zekasını taklit edebilmek için yapay sinir ağları kullanılmaktadır. Derin öğrenmenin kullanım alanları oldukça geniştir. Derin öğrenme genelde ses ve görüntü tanıma işlemleri için kullanılmaktadır. Derin öğrenmenin kullanım alanlarını sıralamak gerekirse plaka tanıma sistemleri, yüz tanıma sistemleri, parmak izi okuyucular, iris okuyucular, ses tanımlama sistemleri, fotoğraf renklendirme, sürücüsüz arabalar ve spam (istenmeyen) e-posta tespitinde kullanılmıştır.

Proje kapsamında kameradan alınan görüntüler üzerinde nesne tanıma yapmak için Evrişimsel Sinir Ağlarını (Convolutional Neural Network) kullanılacaktır. Derin öğrenme sınıfında evrişimsel sinir ağları görsel verileri analiz etmek için en yaygın olarak kullanılan derin sinir ağları sınıfı olarakta bilinir. Bu nesne tanıma için çok etkili bir yöntemdir. Evrişimsel sinir ağları, convolutional, non-linearity, pooling, flattening ve fully-connected katmanlarını kullanarak görüntüyü işler.

Katmanların ne amaçla kullanıldığından bahsederek eğer convolution katmanı resim üzerindeki nesnelerin özelliklerini saptamak için kullanılır. Bu katmanda resim üzerinde filtre uygulayarak düşük seviyeli ve yüksek seviyeli özellikler çıkartılır. Pooling katmanı katmanlar arasındaki parametreleri ve hesaplama sayısını azaltmak için kullanılır. Flattening katmanı, fully connected katmanına ait girdileri tek boyutlu bir diziye çevirmek için kullanılır. Fully connected katmanı girdilerini flattening katmanından alarak yapay sinir ağı yoluyla işlemleri gerçekleştirir.

Görüntü üzerinde nesne sınıflandırma yapabilmek için standart sinir ağları kullanılır ama nesnelerin özelliklerini belirlemek için evrişimsel sinir ağlarının temelini oluşturan katmanlar kullanılır. Günümüzde nesne tanıma yapmak için kullanılan mimarilerde katmanların sayısı ve katmanların dizilişleri farklıdır. Evrişimsel sinir ağlarının mimari yapılarına örnek olarak LeNet, AlexNet, GoogLeNet, SPP-Net, VGGNet, R-CNN, Fast R-CNN, Faster R-CNN incelenmiştir. İncelenen mimariler içerisinde proje kapsamında kullanacağımız mimari yapısı Faster R-CNN olarak seçilmiştir.



Şekil 3. *Faster Region-based Convolutional Neural Network*

Şekil 3’ te Faster R-CNN mimarisi gösterilmektedir. İlk olarak bu mimariye girdi olarak bir resim verilir. Bu resim convolutional neural network’ ten geçirilip bir özellik haritası çıkartılır. Daha sonra bu aşamada selection search ile bölge önerisi almak yerine ayrı bir Bölge Önerisi Ağı (Region Proposal Network) oluşturulur. Artık bölge önerilerini bu ağ üzerinden yapılır. Belirlenen bölgeler alınıp yeniden şekillendirilir. Şekillendirdikten sonra fully connected katmanından geçirilir ve sınıflandırılma yapılır. Bu işlemler yapıldıktan sonra ortaya eğitilmesi gereken 4 farklı parametre ortaya çıkartılır. Hem bölge önerisi veren ağ eğitilir hemde normal konvolüsyonel işlemlerin yapıldığı ağ eğitilir. Bölge önerisi ağının temelde iki görevi vardır. Birincisi her bölge için orda nesne var mı yok mu diye karar vermesi gerekir. Aynı zamanda önerilerin pencere büyüklüğünü belirlemesi gerekir. Daha sonra asıl ağa geçildikten sonra yapılması gereken iki görev vardır. Asıl sinir ağıımız sınıflandırma işlemi gerçekleştirerek baktığı bölge içerisinde nesneyi tespit edecektir. Daha sonra bulduğu nesnenin sınırlarını belirleyecektir.

B.1.2. Yüz Yüze Görüşme

Proje danışmanımız Şahin IŞIK ve elektrik elektronik mühendisliği bölümü akademisyenlerimizden Muhammed Mert ULUPINAR ile yüz yüze yapılan görüşmelerde öncelikle proje ana hatlarıyla anlatılmıştır. Daha sonra yüz yüze görüşmelerde sormak için hazırlanan 10 adet anket sorusu sorulmuştur. Anket sorularının cevaplandırılmasının ardından proje hakkındaki görüşleri alınmıştır ve projenin sistem gereksinimleri hakkında daha ayrıntılı bilgiler edinilmiştir.

B.1.3. Anket

Proje ile ilgili görüş, öneri ve değerlendirmeleri almak için akademisyenlerimiz Dr. Öğr. Üyesi Şahin IŞIK ve elektrik elektronik mühendisliği bölümü akademisyenlerimiz Araş. Gör. Muhammed Mert ULUPINAR ile görüşülmüştür. Bu görüşmeler sırasında yapılacak olan anket için 10 soru hazırlanmıştır. Anket katılımcılarının proje hakkında fikir ve görüşlerini daha açık ifade edebilmeleri için sorular açık uçlu olarak hazırlanmıştır. Aşağıda hazırlanan anket soruları ve ankete katılanların verdikleri cevaplar aşağıda verilmiştir.

- 1) Daha önce derin öğrenme ve yapay zekayı bir arada kullandığınız proje oldu mu ? Olduysa projeniz neydi?
- 2) Bilgisayar ve telefon kamerasının arasındaki bağlantıyı kablo olmadan hangi yolla aktarmamızı tercih edersiniz? Bluetooth tercihimiz sizce doğru mu?
- 3) Kullandığımız Faster R-CNN mimarisi sizce projemiz için uygun mu?
- 4) Kullandığımız donanım malzemeleri projemiz için yeterli mi?
- 5) Nesneleri tanıması ile hareket edecek robotun hareket durumları ne olmalıdır? Bu hareketleri düzenlerken nelere dikkat etmeliyiz?
- 6) Hareketleri belirleyecek nesneleri nasıl seçmeliyiz? Kaç adet obje tanımasını uygun görürsünüz?
- 7) Derin öğrenme tekniklerinin endüstriyel robotlarda kullanılması hakkında görüşleriniz nelerdir?
- 8) Robot tasarımı donanımsal olarak eksik gördüğünüz yerler var mı? Varsa nedir?
- 9) Robotun nesne tanıma gibi bir özelliğinin olmasını kullanışlı buluyor musunuz?
- 10) Robotumuzu gerçek hayata taşımak istesek sizce hangi alanlarda kullanabiliriz?

Görüşülen Kişi: Dr. Öğr. Üyesi Şahin IŞIK

Araştırma Alanları: Computer Vision, Machine Learning

İletişim Adresi: Eskişehir Osmangazi Üniversitesi Mühendislik Mimarlık Fakültesi Bilgisayar Mühendisliği Bölümü / Lab 14

E-Posta: isahhin@gmail.com

Telefon: 0 (554) 856 48 76

1) Daha önce yaptığım çalışmalarda derin öğrenme ve yapay zekayı bir arada kullandım. Yaptığım proje ‘Üç Farklı Görüntüleme Tekniği ile Buğday Sınıflandırma’ dır.

2) Kablosuz aktarım için bluetooth, GPS, wireless gibi iletişim teknolojilerini kullanırlar. Bu proje kapsamında kullanacağınız wifi ve bluetooth un uygun olduğunu düşünmekteyim.

3) Evet uygundur. Çünkü çoğu projede de Faster R-CNN kullanılmaktadır. CNN resmin tamamını kullanmaktadır bu da maliyetli olmaktadır. R-CNN ile resim patch olarak alınır hem hızlıdır hem de maliyeti azdır. Faster R-CNN ise en hızlısı olmaktadır.

4) Kullandığınız donanım yeterli olmak zorundadır çünkü çok fazla alternatif bulunmamaktadır.

5) Komut olarak sağa, sola, ileri, geri, hızlan, yavaşla komutları kullanılmalıdır. Hangi hareketi yapacağını bilmesi için resmi çok iyi tanımalıdır. Kamerayı zeminde bulunan resimleri en iyi algılayacak şekilde yerleştirmelisiniz.

6) Nesnelerin birbirinden farklı ve ayırt edilebilir olması gerekmektedir. Bir nesneden en az 100 adet bulunmalıdır. Ama 100' den fazla olması sizin açınızdan daha iyi olur. Kaç nesne olmasını gerektiğine gelirsek hareket sayısı kadar nesne olması yeterlidir yani 6 tane sizin için yeterli olacaktır.

7) Eski yöntem traditional olarak adlandırılmaktadır. Bu yöntem feature extraction tabanlı veya rank tabanlı çalışır. Bu yöntemden feature çıkartılıp bir sınıflandırıcı ile sınıflandırılır. Örneğin SVM ile sınıflandırılır. Fakat bu yöntemlerin performansı düşüktür. Bu yüzden günümüzde derin öğrenme teknikleri özellikle CNN tabanlı metotlar endüstride yaygın olarak kullanılmaktadır. Bütün endüstriyel uygulamalar güncellenerek derin öğrenme tekniklerini kullanmaya başlamıştır.

8) Robotunuz donanımsal olarak yeterlidir.

9) Evet buluyorum. Birden fazla örnek bulunmaktadır. Bunları inceleyebilirsiniz. Örneğin kaynak yapan robot kolunun makine öğrenmesi ile eğitilmesi gibi.

10) Fabrika, sanayi, biyometrik uygulamalarda, sağlık alanlarında kullanılabilir.

Görüşülen Kişi: Araş. Gör. M. Mert ULUPINAR

Araştırma Alanları: Yarı iletkenler ve Elektronik Devreler

İletişim Adresi: Eskişehir Osmangazi Üniversitesi Mühendislik Mimarlık Fakültesi Elektrik - Elektronik Mühendisliği Bölümü /

E-Posta: ulupinar.mert@gmail.com

Telefon: +90 (538) 372 63 31

- 1) Daha önce yaptığım çalışmalarda derin öğrenme veya yapay zekayı kullanmadım. Günümüzde derin öğrenme teknolojilerinin geldikleri noktaya baktığımızda robotik alanında kullanımın

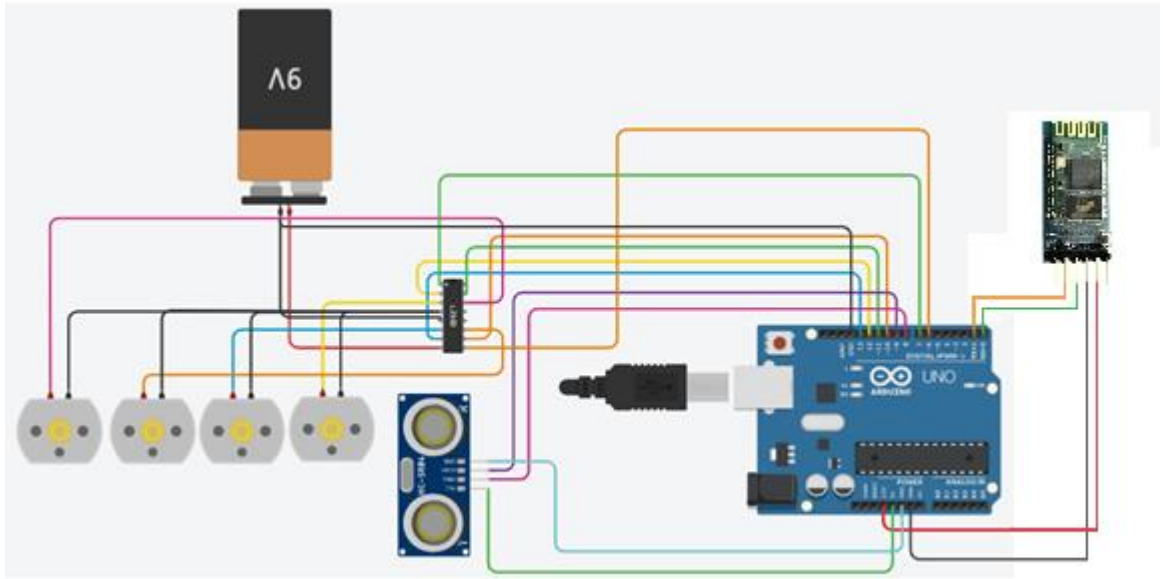
artacağını düşünüyorum. Bu yüzden ilerdeki çalışmalarına derin öğrenmenin ekleneceğini düşünmekteyim.

- 2) Kablosuz aktarım için 3G, wi-fi ve wimax gibi iletişim teknolojilerini kullanırlar. Bu proje kapsamında kullanılabilecek en iyi seçeneğin wi-fi olduğunu düşünmekteyim.
- 3) Faster R-CNN mimarisi hakkında çok fazla bir bilğim yok.
- 4) Arduino UNO bu proje için yeterlidir. Ancak projenin daha da gelişmesi durumunda projede kullanılan Faster R-CNN modelini raspberry pi içerisinde kullanabilirsiniz. Böylece bilgisayarı projeden çıkartabilirsiniz. Tavsiye olarak önerebileceğim yöntem ise NVIDIA Jetson Nano Developer Kit ile kendi kendine öğrenenler ve makerlar görüntü sınıflandırma, nesne tespiti, segmentasyon ve dil işleme gibi yapay zeka yazılım geliştirme ortamlarını ve modellerini çalıştırabilirsiniz.
- 5) Kullanılacak ortamda gereken hareket çeşitliliğine bakmak gerekir. Ama projenin genel hatlarına baktığımda robotun nesneyi tanıması, tanıdıktan sonra da gitmesi gereken güzergahtan veya yoldan çıkmayacak hareketler yapmasını isterim.
- 6) Benzerliği az olan nesneler seçilebilir. Özelliklerinin benzer olmaması modelin nesneyi tanıma oranını arttıracaktır. Örneğin kare, daire, üçgen gibi geometrik şekiller seçilebilir. Proje de robotun tanıyacağı 6 farklı nesne seçilebilir.
- 7) Kullanışlı bir projedir. Derin öğrenme teknolojilerinin gelişme düzeyine baktığımda endüstriyel alanında kullanımı da artacaktır. Bu yüzden bu alanda yapılan güzel bir proje olmuş.
- 8) HC-S04 ultrasonik mesafe sensörü yerine daha hızlı olan TF Mini Lidar veya Garmin Lite V3 kullanılabilir. TF Mini Lidar, time-of-flight (ToF) teknolojisine dayanan tek yönlü bir lazer telemetredir. Lazer algılayıcı engel ile arasındaki mesafeyi tespit etmek için bir mesafe ölçme aracı olarak kullanılabilir. Garmin Lite V3 ise optik mesafe sensörüdür. Drone, robot ve insansız araç uygulamaları için idealdir. Her iki mesafe sensörü de ultrasonik mesafe sensöründen hızlı çalışır.
- 9) Nesne tanıma yaparak robotların kullanım çeşitliliğini artırması bakımından kullanışlı buluyorum.
- 10) Fabrika, sanayi gibi ortamlarda kullanılabileceğini düşünüyorum.

B.1.4. Gözlem

Yaptığımız yüz yüze görüşme ve anketlerin sonucunda projemizin yapılabilirliği onaylanmıştır. Farklı data setlerle projemizin fabrika, sanayi ve benzeri ortamlarda kullanılabileceği düşünülmektedir. Projenin ilk olarak prototip halinde çıkarılması planlanmış olup belli özelliklere göre geliştirilmeler yapılması hedeflenmektedir.

B.1.5. Prototip ve Hızlı Uygulama Tasarım (Rapid Application Design- RAD)



Şekil 4. Devre Şeması

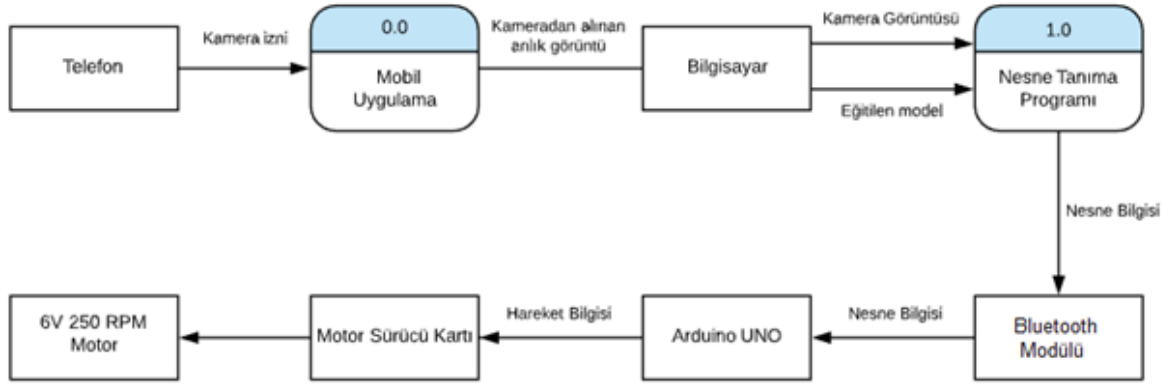
Şekil 4' te robotun devre tasarımı verilmiştir. Devre tasarımında kullanılan L293D kodlu H köprüsü motor tahriği yerine gerçek zamanda LN298N kodlu motor sürücü kartı kullanılacaktır. Tasarımda L293N kodlu H köprülü motor tahriği kullanılma sebebi tasarım yapılan programda motor sürücü kartının tasarımının bulunmamasıdır. Bu yüzden tasarım çiziminde H köprülü motor tahriği kullanılmıştır. İkisi de aynı görevi yapmaktadır. Devre tasarımında arduino UNO, bluetooth modülü, ultrasonik mesafe sensörü, 4 adet DC motor seti ve 9 voltluk pil bulunmaktadır.

B.1.6.Ortak Uygulama Tasarım (Joint Application Design-JAD)

Grup üyeleri ve proje yürütücüleri ile sıkça görüşmeler yapılmış ve bu görüşmelerde projenin kilit taşları üzerinde durulmuştur. Sanayi odaklı bir proje olduğu için endüstriyel bir ihtiyacı karşılamaya yönelik elverişli ve kullanılabilir yöntemler kullanılmaya çalışılmıştır. Son teknoloji ürünleri ile proje gerçekleştirilmeye çalışılmıştır. Grup üyeleri arasında yapılan görev dağılımları ile optimum sürede ve maliyette projenin tamamlanması planlanmıştır.

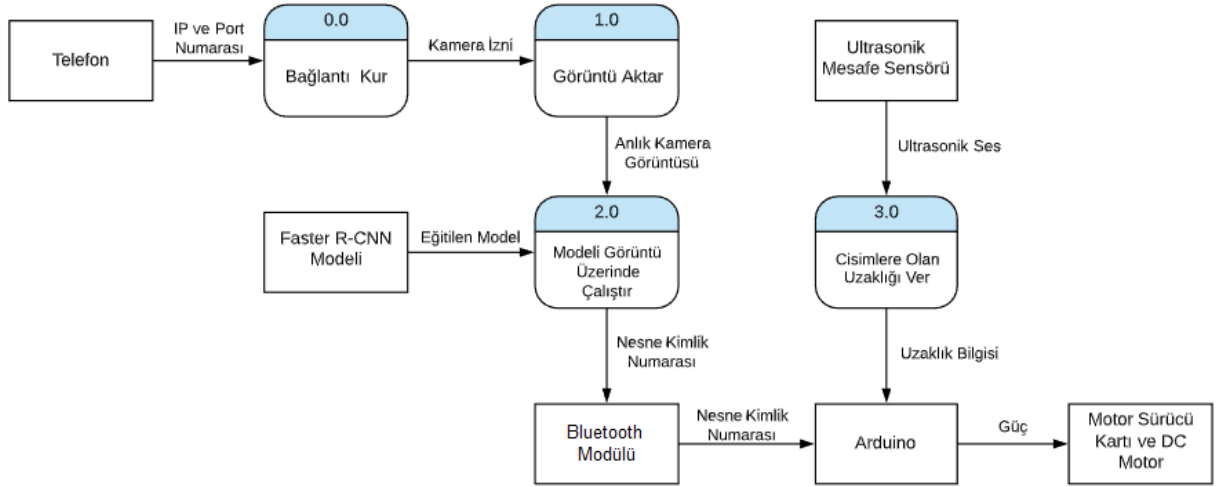
B.1.7. Veri Akış Şemaları (Kavramsal, Mantıksal, Fiziksel Veri Akış Şeması)

B.1.7.1. Kavramsal Veri Akış Şeması



Şekil 5. Kavramsal Veri Akış Şeması

B.1.7.2. Mantıksal Veri Akış Şeması



Şekil 6. Mantıksal Veri Akış Şeması

B.1.7.3. Fiziksel Veri Akış Şeması



Şekil 7. Fiziksel Veri Akış Şeması

B.1.8. (OPS)Olay Tabloları, Durum Formları, İşlevsel Analiz Raporu

B.1.8.1. (OPS) Olay Tabloları

Olay	İstemci	Tetikleyici	İşlem	Yanıt	Hedef
Robotun sağa dönmesi ve ilerlemesi	Robot	solidOval isimli geometrik şekil	Motor sürücü kartına robotun sağa dönebilmesi ve daha sonra düz ilerleyebilmesi için verilerin gönderilmesi	Robot -90° döner ve daha sonra düz bir şekilde ilerler	Nesneyi algılayıp robotun uygun hareketi yapması
Robotun sola dönmesi ve ilerlemesi	Robot	solidDiamond isimli geometrik şekil	Motor sürücü kartına robotun sola dönebilmesi ve daha sonra düz ilerleyebilmesi için verilerin gönderilmesi	Robot 90° döner ve daha sonra düz bir şekilde ilerler	Nesneyi algılayıp robotun uygun hareketi yapması
Robotun düz bir şekilde ilerlemesi	Robot	solidSquiggle isimli geometrik şekil	Motor sürücü kartına robotun düz ilerleyebilmesi için verilerin gönderilmesi	Robot düz bir şekilde ilerler	Nesneyi algılayıp robotun uygun hareketi yapması
Robotun geriye doğru ilerlemesi	Robot	emptySquiggle isimli geometrik şekil	Motor sürücü kartına robotun geri gidebilmesi için verilerin gönderilmesi	Robot geri bir şekilde ilerler	Nesneyi algılayıp robotun uygun hareketi yapması
Robotun durdurulması	Robot	emptyOval isimli geometrik şekil	Motor sürücü kartına robotun durdurulması için verilerin gönderilmesi	Robot durur	Nesneyi algılayıp robotun uygun hareketi yapması
Robotun güzergah boyuca aldığı yolu geri gitmesi	Robot	emptyDiamond isimli geometrik şekil	Sağ, sol, ileri ve geri komutlarının tersini uygulamak için motor sürücü kartına uygun verilerin gönderilmesi	Robot geldiği yol üzerinden geri döner	Nesneyi algılayıp robotun uygun hareketi yapması

Tablo 5. Olay Tablosu

B.1.8.2. Durum Formları

Durum İsmi: Robotun sağa dönmesi ve ilerlemesi			İşlem No: 1
Tanım: solidOval isimli şeklin algılanması ve robotun sağa dönüp ilerlemesi			
Tetikleyici: solidOval isimli şeklin algılanması			
Tetikleyici Türü: Nesne algılama programı			
Girdi İsmi	Kaynak	Çıktı	Hedef
solidOval	Robot	Robot -90° döner ve daha sonra düz bir şekilde ilerler	Nesneyi algılayıp robotun uygun hareketi yapması

Tablo 6: 1 no' lu işleme ait durum tablosu

Durum İsmi: Robotun sola dönmesi ve ilerlemesi			İşlem No: 2
Tanım: solidDiamond isimli şeklin algılanması ve robotun sola dönüp ilerlemesi			
Tetikleyici: solidDiamond isimli şeklin algılanması			
Tetikleyici Türü: Nesne algılama programı			
Girdi İsmi	Kaynak	Çıktı	Hedef
solidDiamond	Robot	Robot 90° döner ve daha sonra düz bir şekilde ilerler	Nesneyi algılayıp robotun uygun hareketi yapması

Tablo 7: 2 no' lu işleme ait durum tablosu

Durum İsmi: Robotun düz bir şekilde ilerlemesi			İşlem No: 3
Tanım: solidSquiggle isimli şeklin algılanması ve robotun düz bir şekilde ilerlemesi			
Tetikleyici: solidSquiggle isimli şeklin algılanması			
Tetikleyici Türü: Nesne algılama programı			
Girdi İsmi	Kaynak	Çıktı	Hedef
solidSquiggle	Robot	Robot düz bir şekilde ilerler	Nesneyi algılayıp robotun uygun hareketi yapması

Tablo 8: 3 no' lu işleme ait durum tablosu

Durum İsmi: Robotun geriye doğru ilerlemesi			İşlem No: 4
Tanım: emptySquiggle isimli şeklin algılanması ve robotun geriye doğru ilerlemesi			
Tetikleyici: emptySquiggle isimli şeklin algılanması			
Tetikleyici Türü: Nesne algılama programı			
Girdi İsmi	Kaynak	Çıktı	Hedef
emptySquiggle	Robot	Robot geri bir şekilde ilerler	Nesneyi algılayıp robotun uygun hareketi yapması

Tablo 9: 4 no' lu işleme ait durum tablosu

Durum İsmi: Robotun durdurulması			İşlem No: 5
Tanım: emptyOval isimli şeklin algılanması ve robotun durdurulması			
Tetikleyici: emptyOval isimli şeklin algılanması			
Tetikleyici Türü: Nesne algılama programı			
Girdi İsmi	Kaynak	Çıktı	Hedef
emptyOval	Robot	Robot durur	Nesneyi algılayıp robotun uygun hareketi yapması

Tablo 10: 5 no' lu işleme ait durum tablosu

Durum İsmi: Robotun güzergah boyuca aldığı yolu geri gitmesi			İşlem No: 6
Tanım: emptyDiamond isimli şeklin algılanması ve geldiği yol üzerinden geri döner			
Tetikleyici: emptyDiamond isimli şeklin algılanması			
Tetikleyici Türü: Nesne algılama programı			
Girdi İsmi	Kaynak	Çıktı	Hedef
emptyDiamond	Robot	Robot geldiği yol üzerinden geri döner	Nesneyi algılayıp robotun uygun hareketi yapması

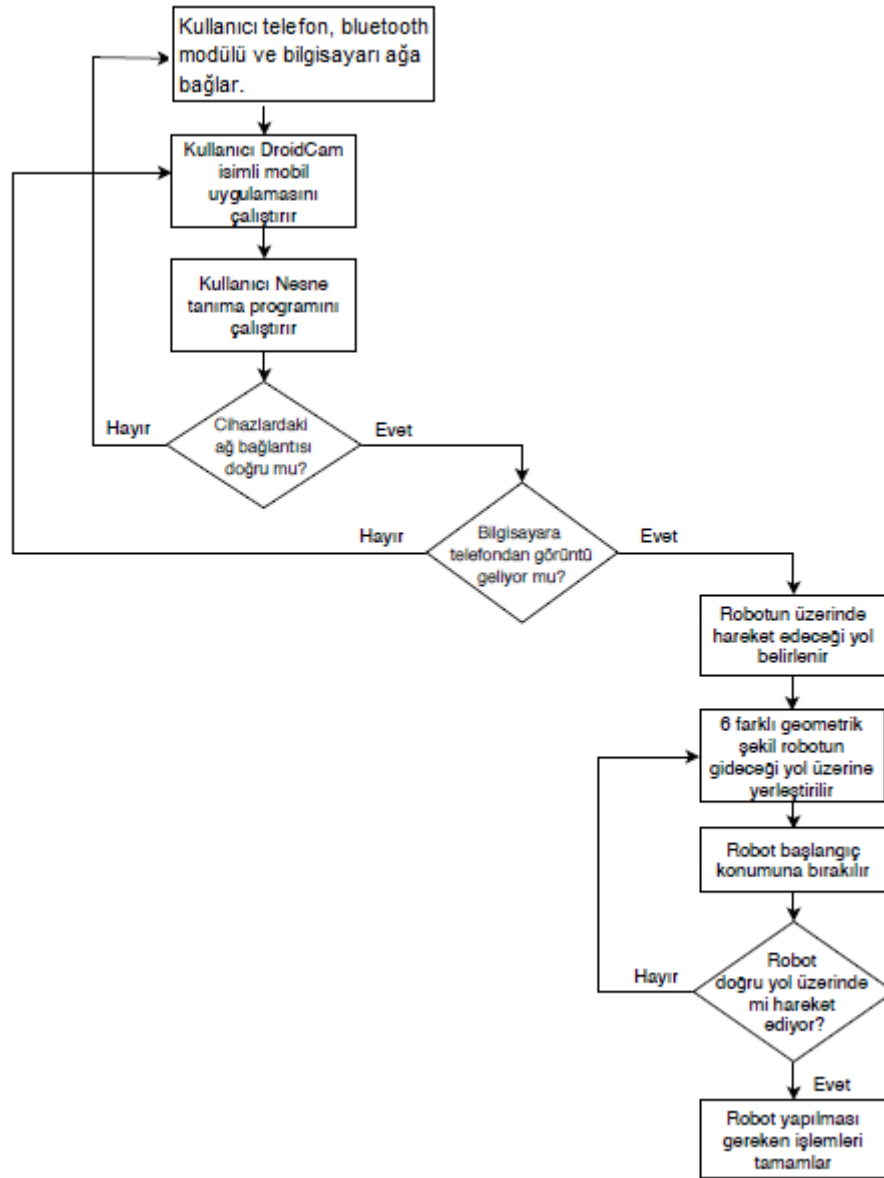
Tablo 11: 6 no' lu işleme ait durum tablosu

B.1.8.3. İşlevsel Analiz Raporu

Proje Kodu	: BP-2019
Proje Adı	: Derin Öğrenme Tabanlı Robotun İç Ortamlarda Yönlendirilmesi
Hazırlayanlar	: Elif GENÇ, Özge ÇEVİK, Ahmet Selim KÜÇÜKKARA
Hazırlanma Başlangıç Tarihi	: 23.09.2019
Son Değiştirme Tarihi	: 13.11.2019
Versiyon	: 1.1
İlgili Kişi	: Dr. Öğr. Üyesi Şahin IŞIK
Görevi	: Proje Danışmanı

Şekil 8. İşlevsel Analiz Raporu

B.1.9. İş Akış Şeması



Şekil 9. İş Akış Şeması

B.2. Sistem Gereksinimleri

Projemizin gereksinimlerini tanımlamak, anlamak ve seçmek için gereksinim analizi yapılmıştır. Bu gereksinim çalışmaları İşlevsel gereksinimler, sistem ve kullanıcı arayüzleri ile ilgili gereksinimler, veriyle ilgili gereksinimler, kullanıcı ve insan faktörü gereksinimleri, güvenlik gereksinimleri, teknik ve kaynak gereksinimleri, fiziksel gereksinimleri başlıkları adı altında ayrıntılı olarak incelenmiştir.

B.2.1. İşlevsel Gereksinimler

- Nesne tanıma modelimizin istenilen nesneleri doğru ve hızlı bir şekilde tanınması gerekmektedir.
- Nesne tanıma yapacak olan program Python dili kullanılarak yazılacaktır.
- Robotumuz bilgisayardan gönderilecek verilere göre hareket etmesi gerekmektedir.
- Bluetooth modülü ile Arduino arasında kablosuz haberleşme yapılacaktır. İletişim bluetooth modülünün MAC adresi ve port numarası kullanılarak yapılacaktır.
- Telefonda çalışacak olan program IP Webcam isimli ücretsiz mobil bir uygulamadır.
- IP Webcam mobil uygulama kameradan alınan anlık görüntüleri bilgisayara aktarırken https protokolünü kullanılacaktır.
- Arduino gelen verilere göre sağa, sola dönme hareketi ve ileri ve geriye doğru gitme hareketlerini yapacaktır.
- Robotun istenilen hareketi yaparken güzergahından çıkmaması gerekecektir.
- Robotun önünde bir nesne varsa nesneye çarpmaması gerekecektir.

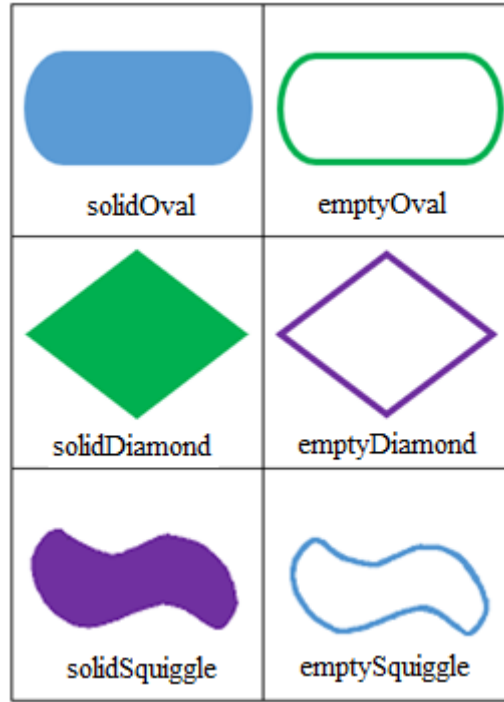
B.2.2.Sistem ve Kullanıcı Arayüzleri ile ilgili Gereksinimler:

Bilgisayar tarafından alınan anlık güncel kamera görüntüleri üzerinde nesne tanıma programı çalışacaktır. Nesne tanıma programı çalıştırıldığında “Video” isimli bir pencere açılarak alınan kamera görüntüsü üzerinde nesnenin ismini, nesnenin konumunu ve tanınan nesne özelliklerinin yüzdesi verilecektir.

B.2.3.Veriyle İlgili Gereksinimler:

- Faster R-CNN modelini eğitmek için 6 farklı nesne sınıfı oluşturulmuştur. Nesnelerin isimleri solidDiamond, solidOval, solidSquiggle, emptyDiamond, emptyOval ve emptySquiggle’ dır.
- Her nesne sınıfında 1461 fotoğraf vardır.
- Fotoğraflar üzerinde eğitim yapmak için fotoğraflar üzerindeki nesneleri bir çerçeve içerisine alarak nesneleri isimlerine göre etiketlememiz gerekmektedir. Etiketleme yapmak için “labelImg” isimli bir program kullanılacaktır. Bu program kullanılarak her bir fotoğrafa özel bir xml türünde bir dosya oluşturulur.
- Fotoğrafların %70’ i modeli eğitmek, %30’ u ise modeli test etmek için kullanılır.
- Telefon ile bilgisayar ortak bir ağa bağlanması gerekecektir. Ortak ağa bağlandığında telefona bir IP numarası verilir. Verinin aktarılacağı port numarası ise “4747” olacak şekilde tanımlanmıştır.
- Bilgisayarın, telefonunun kamerasına bağlanması için bir string değerine ihtiyacı vardır. Bu değer telefonun IP numarası ve kullanılan port numarası ile oluşturulmaktadır. Bu string değer örnek olarak “http://192.168.43.1:8080/shot.jpg” şeklinde olmalıdır.
- Telefonun kamerasının bilgisayar ile olan bağlantısı kontrol edilmelidir. Bağlantının başarısız olması durumunda uyarı mesajı verilmelidir.

- Bağlantının başarılı olması durumunda kamera görüntüsünün alınıp alınmadığının kontrol edilmesi gerekmektedir. Görüntünün alınamaması durumunda bir uyarı mesajı verilir.
- Bilgisayarda eğitilen model ve alınan anlık görüntü kullanılarak görüntü üzerinde nesne tanıma algoritması kullanılır. Bu algoritmanın hızlı, güncel bir şekilde ve doğru çalışması gerekmektedir.
- Robota gönderilen nesnelerin kimlik numaraları integer bir değerdir. Bu integer değerine göre robot önceden belirlenen hareketleri yapacağı için hareketlerin karıştırılmaması gerekmektedir.
- Robot sabit hızlı bir hareket yapacaktır. Hız değişkeni robotun kullanılacağı ortamlara ve güzergahlara göre arduino programı içerisinde değiştirilecektir.
- Projede robotun hareketlerini kontrol eden ve modelin tanınması gereken nesneler şekil 10 'te gösterilmiştir.



Şekil 10. Modeli eğitmek için kullanılan nesneler

B.2.4.Kullanıcılar ve İnsan Faktörü Gereksinimleri, Güvenlik Gereksinimleri:

Robotu kullanacak personel, robotun gitmesi gereken güzergahın üzerine belirlenen nesneleri yerleştirmesi gerekmektedir. Bu işlem yaptıktan sonra robotun istenilen noktaya gidip gitmediğinin kontrolünün yapılması gereklidir.

Robot kullanımının kolay olması nedeniyle yüksek eğitim düzeyi ve bilgisayar bilgisine gerek duyulmamaktadır. Robotun sağlıklı bir şekilde yapması gereken işleri gerçekleştirebilmesi için personellerin nesneler ve robot hakkında bilgilendirilmesi gerekmektedir. Bu aşama şöyle özetlenebilir.

- Personel robot ve nesneler hakkında bilgilendirilmelidir.
- Uygun hareket için nesnelerin doğru şekilde ortama yerleştirilmesi gerekmektedir.
- Robotun hareketleri kontrol edilmelidir.
- Proje daha çok robot ve hareketleri ile ilgili olduğu için insan faktörünün fazla bir etkisi bulunmamaktadır. Kullanıcı sadece güzergâh üzerine yerleştirilmesi gereken nesneleri yerleştirmek için devreye girecektir.

Fiziksel ve yazılımsal olarak güvenlik tedbirlerinin alınması gerekmektedir. Güvenlik gereksinimleri aşağıda listelenmiştir.

- Fiziksel olarak alınacak güvenlik tedbirleri kameranın robot üzerinde sabit durması gerekmektedir.
- Bilgisayar ve kamera aynı ağa bağlı olmalı, ağın dış etkenler yüzünden kopmaması gerekmektedir.
- Bluetooth modülü, Arduino, motor sürücü kartı ve motorlar arasındaki kabloların uzunluğunun gerekenden uzun veya kısa olmaması gerekmektedir.
- Robot çalışır durumda iken herhangi bir dış kaynak tarafından fiziksel temas durumunun olmaması gerekir.
- Robot hareketlerinin doğruluğu nesnelerin doğru algılanmasına bağlıdır. Bu yüzden model eğitimi yüksek seviyede olmalıdır.

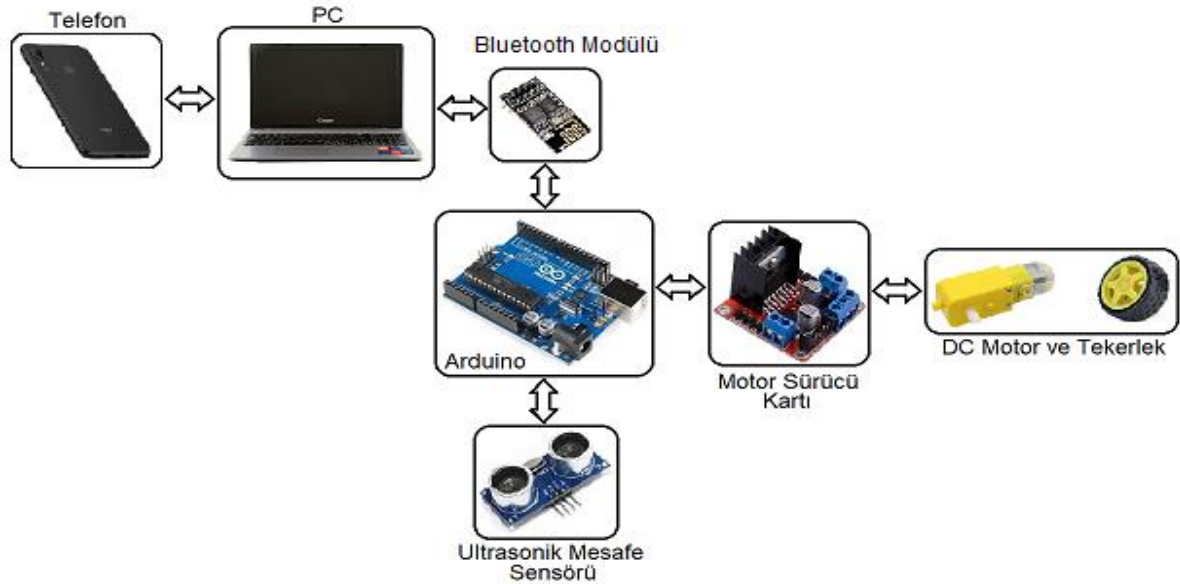
B.2.5.Teknik ve Kaynak Gereksinimleri, Fiziksel Gereksinimler:

- Robot üzerinde bulunan telefon ile bilgisayarın internet üzerinden bağlanması gerekecektir.
- Anlık alınan kamera görüntüleri telefonda bilgisayara internet üzerinden bilgisayara aktarılacaktır. Veri aktarımının hızlı olması gerekecektir.
- Modelin tanınması gereken nesneleri hızlı ve doğru bir şekilde tanınması sağlanacaktır.
- Tanınan nesnelerin kimlik numaralarının bluetooth modülü üzerinden Arduino' ya aktarılacaktır.
- Bilgisayarın ve bluetooth modülü kablosuz haberleşmesi gerekecektir.
- Bluetooth modülü 5 volt ile çalışmaktadır. Bu yüzden Arduino' nun 5 voltluk power pinine bağlanacaktır.
- Bilgisayar ile bluetooth modülü arasında kablosuz iletişim kurulur.
- Robot gelen verilere göre motor sürücü kartını kontrol edecektir bu yüzden Arduino ile motor sürücü kartına veri kabloları aracılığıyla Arduino' ya bağlanacaktır.
- Motor sürücü kartı güç kabloları aracılığıyla bir kaynağa bağlanacaktır ve motor sürücü kartına verilen gerilimin 12 volt olması gerekmektedir.

- Robotun önünde bulunan nesnelere çarpmaması için ultrasonik mesafe sensörü kullanılacaktır. Ultrasonik mesafe sensörünün trig ve echo pinleri arduinonun dijital pinlerinden 0. ve 1. pinlerine bağlanacaktır.
- Ultrasonik mesafe sensörü 5 volt ile çalışmaktadır. Bu yüzden arduinonun 5 volt' luk power pinine bağlanacaktır.
- Ultrasonik mesafe sensörü önünde bulunan nesne ve robot arasındaki mesafe 10 cm olduğunda robotun durmasını sağlayacaktır.
- Ultrasonik mesafe sensörünün, motor sürücü kartının ve bluetooth modülünün ground pinlerinin ortak bağlanması gerekecektir.

C.TASARIM

C.1. Sistem Tasarımı



Şekil 11. Sistem Tasarımı

Sistem, şekil 11’ da gösterilen robot üzerindeki kameradan gelen veriyi anlık olarak alacak ve bu verileri IP Webcam uygulaması ile bilgisayar üzerinde görülebilmesi sağlanacak. Derin öğrenme yöntemleri ile eğitilmiş sisteme bu veri csv dosyası üzerinden okunarak girdi olarak alınacak, çıktı olarak algılanan nesnenin hangi hareketi işaret ettiği robota iletilecektir. Test aşamasında, sistemin sunulan imkanlar dahilinde hazırlanan bir parkurda çalıştırılması planlanmaktadır. Oluşturulan parkurda bazı donanım modüllerinin eklenmesi gerekebilir. Geliştirilebilecek bazı durumlar analiz edilmiş olur. Kullanılan ortama göre sistemin gerçek zamanlı çalışması değişiklik gösterebilir.

C.2. Kullanıcı ve Sistem Arayüzü Tasarımları

Şekil 12’ te görüldüğü üzere IP Webcam isimli mobil uygulamaya ilk girildiğinde karşılaşılan arayüz verilmiştir. Bu arayüz daha çok kamera ayarlarını içeren bir menüdür. IP Webcam uygulaması çalıştığında şekil 13’ te görüldüğü gibi telefonun bağlandığı ağdan aldığı bir IP numarası gözükmektedir. Bilgisayar ile telefon arasında iletişimi kuracak port numarası da gösterilmektedir. Tarayıcı üzerinden kamerayı kullanabilmemiz için bir adres vermektedir. Proje içerisinde bu adres üzerinden birer saniye arayla resim alınmaktadır.

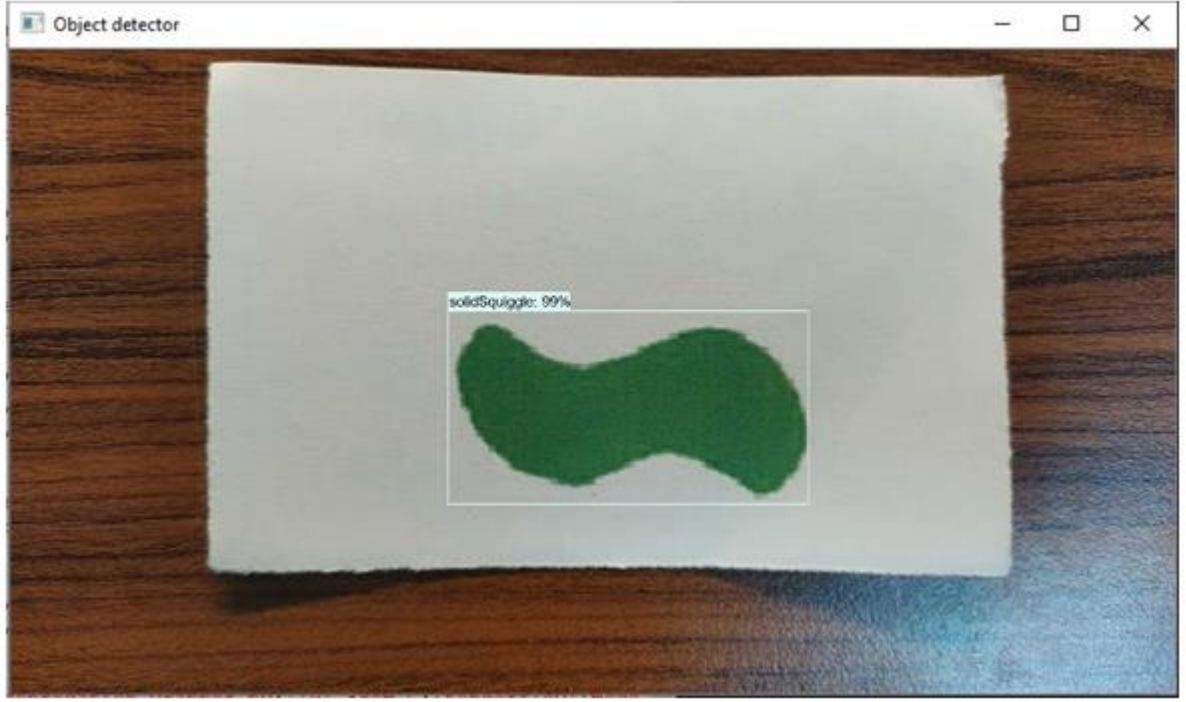


Şekil 12. IP Webcam Mobil Uygulama
Giriş Arayüzü



Şekil 13. IP Webcam Mobil Uygulaması
Kamera Kullanım Ekranı

Tarayıcı üzerinde görüntülenen kamera verileri ile nesne tanınması yapabilmek için eğitilen model çalıştırılır. Şekil 14’ te örnek bir nesne tanıma işlemi yapılmaktadır.



Şekil 14. Nesne Tanıma Uygulaması Örnek Arayüzü

C.3. Test Tasarımı

Yazılım testlerinin amacı oluşturulan sistemin proje başında belirlenen gereksinim ve isterleri karşılayıp karşılamadığını görme amacı taşımaktadır. Testlerin, sistemin istenildiği gibi hareket yeteneğine sahip olup olmadığını göstermesi sebebi ile test tasarımının da önemini arttırmaktadır. Sistem fonksiyonlarının doğru bir şekilde çalıştığını doğru test yöntemleri ve tasarımı kullanılarak sağlanabilir.

C.3.1. Gereksinim Analizlerinden Teste Yönelik Hedeflerinin Detaylandırılması

Test yapılmadan önce testin amacının ve beklenen çıktının ne olacağına karar verilmesi gerekmektedir aksi halde yapılan testlerin amaçları geliştiricileri yanıltabilir. Bu projede ki test hedeflerini ifade etmek gerekir ise;

- Nesne algılama işlemlerinin kontrolü, nesne tanıma makul olsa dahi ileri seviyede hataya yol açacak unsurların testi.
- Nesneleri tanıma makul olsa bile hareketin doğru şekilde gerçekleştirilebilmesi testi.
- İlk nesneyi tanıdıktan sonra hareketin doğru bir şekilde gerçekleştiğinin gözlemlenmesi ardından diğer nesneye kadar karşılaşılabilecek hataların testi.
- İlk nesneyi algılayıp ikinci nesneye kadar hareketin doğru bir şekilde gerçekleştiğinin testi.
- İkinci nesneye ulaştığında nesne algılama işleminin kontrolü testi.

- İkinci nesneyi tandıktan sonra hareketin doğru bir şekilde gerçekleştirilebilmesi testi.
- Oluşturulan yolda nesnenin algılanma kontrolü ve hareketin doğru gerçekleştirilebilmesi testi.
- Hareket sırasında olası durumların dışında gerçekleşen olaylara robotumuzun tepki süresi testi.
- Hareket sırasında oluşan olası durumların dışındaki olayların gerçekleştirilebilmesi testi.
- Sistem çalışmasında oluşabilecek koddan kaynaklı hataların testi
- Robotun hızı ve hıza göre nesne algılama yetisi testi.
- Robotun nesneyi algılamadan ilerlemesiyle karşılaşılabilecek hatalar.
- Robotun sensör yardımıyla etrafındaki eşyalara çarpmadan hareket edebilmesi testi.
- Birbiri ile entegre edilmiş şekilde çalışan sistemlerin testi
- Robotun sağa, sola, ileri ve durma hareketlerini doğru bir şekilde gerçekleştirebilmesi testi.
- Sağa ve sola dönme durumlarında 90 derece dönebilmesini gerçekleştirme testi.
- Oluşturduğumuz path dışında nesne algılaması testi.
- Farklı nesneleri algıladığında robotun tepkisinin testi.

C.3.2. Fonksiyonel Test Tasarımı

Bu bölüm 5 parçaya ayrılmaktadır. Birim test tasarımında geliştirme sırasında hatalar giderilmek istenmektedir. Entegrasyon test tasarımında uygulama geliştirilirken birim testlerle her aşamada kontrol yapılacaktır. Kapalı kutu test tasarımında kullanılacak mekanizmaya karar verilecektir. Beyaz kutu test tasarımında projenin doğru çalışıp çalışmadığı kontrol edilecektir. En son olarak kabul test tasarımında projenin gerçek ortamda test edilerek hataların gözlenmesi sağlanacaktır.

C.3.2.1.Birim (Unit) Testleri Tasarımı

Birim test tasarımının amacı projenin geliştirme aşamasındaki hataları hızlı bir şekilde gidermektir. Bu projede geliştirme sırasında kat edilecek adımlar sırasıyla anlık görüntünün kameradan bilgisayara aktarılması, aktarılan verinin bilgisayar tarafından tanınması ve ardından robota hareket için veri aktarılması aşamalarından oluşur. Elimizde var olan nesneler ile sistemin eğitilmesi ve eğitilen sistemden sonra anlık alınan verinin sayesinde eğitimin doğruluğunun test edilmesi amaçlanmıştır. Her aşama geliştirildikten sonra anlık görüntü alınabiliyor mu, bilgisayara aktarılabilir mi, aktarılan veri bilgisayar tarafından tanınıyor mu, robotun hareketi için veri aktarımı yapılabilir mi, bu testlerin sonucu olumlu ise bir sonraki aşamaya geçilecektir.

C.3.2.2.Entegrasyon Testleri Tasarımı

Uygulama geliştirilirken birim testleri ile her aşamada gerekli kontroller yaparak uygulanacaktır. Fakat sistem birim testlerinden geçse bile aşamalar arasında belli uyumsuzluklar çıkabilmektedir. Bu sebeple her aşama gerçekleştirilirken birim testleri ile birlikte daha önceki aşamalarda geliştirilen kısımları birbiriyle uyumlu çalıştıklarını kontrol etmek için, entegrasyon

testleri de ikinci aşamadan itibaren yapılacaktır. Entegrasyon testlerinin gerçek ortam üzerinde uygulanması ile sistemin ek gereksinim ihtiyacı belirlenebilecek ve bu doğrultuda düzenlemeler yapılabilecektir.

C.3.2.3.Kapalı Kutu (Black-Box) Testleri Tasarımı

Projedeki uygulama uzun bir eğitim sürecinden geçecek olup, eğitim için kendi oluşturduğumuz veri setindeki veriler kullanılacaktır. Sistemi eğitmek amacıyla kullanılacak veriler ile test aşamasında kullanılacak verilerin birbirinden ayrılmasının en önemli sebebi doğru bir karar mekanizmasının ortaya çıkıp çıkmadığını test etmektir.

C.3.2.4.Beyaz Kutu (White-Box) Testleri Tasarımı

Bu testlerin asıl amacı ise kabul testlerinden önce geliştirmenin son aşamasındaki projenin doğru çalışıp çalışmadığını kontrol etmektir. Bu testi gerçekleştirebilmek için kameradan anlık alınan veri bilgisayara anlık olarak aktarılmasına ihtiyaç duyulacaktır. Gerçek zamanlı testler ile sistemin doğruluk değeri sunulacaktır.

C.3.2.5.Kabul Testleri Tasarımı

Sistemin gerçek kullanım alanına yerleştirilmesinden sonra çalıştırılır ve robotun hareketleri gözlenir hata yaptığı zaman not alınıp düzeltilecektir. Bu testi yaptıktan sonra aynı testi bir kere de bu sistemi kullanılabileceği uygun bir ortamda denenmesi düşünülmektedir ve bir sorun çıkması durumunda geliştirmelere yapılabilecektir.

C.3.3. Performans Test Tasarımı

Daha önceden hazırlanan yol üzerinde resimleri tanıyarak bu resimlerin kimlik numaralarına göre doğru hareketi yapan robotun performansı test edilir. Algılanan nesnenin kimlik numarası altı ise robotumuz bu resimden sonra sağa dönüyorsa, ilerleyip başka bir resim ile karşılaştığında ise ona atanan uygun hareketi yapması beklenmektedir. Bu aşamada kameradan birer saniye aralıklarla alınan görüntünün bilgisayara aktarımının, modelin görüntü üzerinde çalışma süresi, robotun veriyi alma süresi, robotun hızı ve bu aşamalarındaki performansı test edilir.

Bu sistemin birim zamanda nesneleri algılayıp hareket etmesi performans test tasarımı olarak düşünülebilir.

C.4. Yazılım Tasarımı

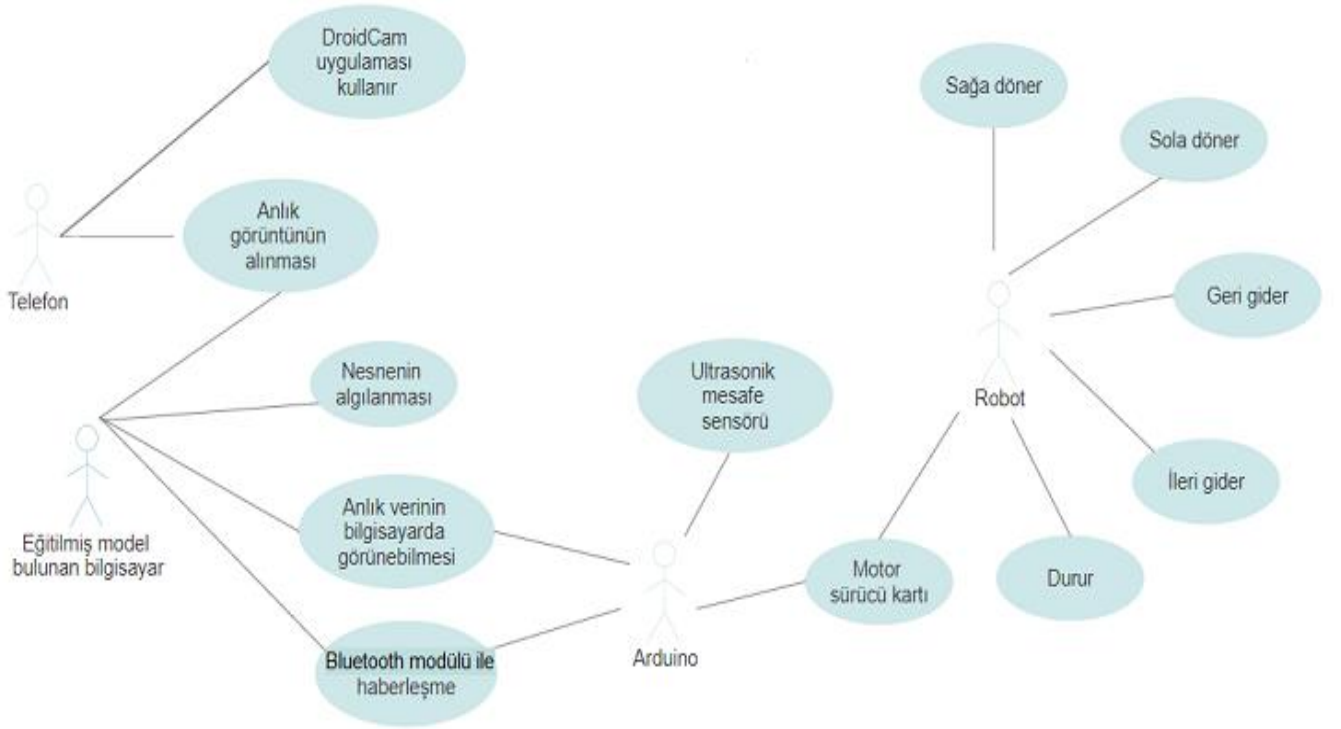
Bu bölüm 2 kısımdan oluşmaktadır. Gereksinime Bağlı Tasarım Kalıpları Seçimi ve Uml Kullanarak Tasarımı Diyagramları Oluşturma kısımlarından aşağıda bahsedilmiştir.

C.4.1. Gereksinime Bağlı Tasarım Kalıp(lar)ı Seçimi

Gerçekleştirilecek projede “MVC (Model-View-Controller), Pattern” kullanılması planlanmaktadır. MVC Model, Controller, View üçlüsünden oluşan bir mimari desenidir. Kısaca bahsetmek gerekirse Model uygulamada kullanılan verileri temsil eder ve verilerin işleme mantığının saklandığı kısımdır. View; Basitçe, uygulamanızın kullanıcılarınızın gözüyle gördüğü kısımdır. Controller; Kullanıcıların View üzerinden gerçekleştirdiği işlemlerle alınan veriyi Model’e taşır, Model’den aldığı veriyi View üzerinden kullanıcıya gösterir. Uygulamamızda nesneleri tanıması için Faster R-CNN modelini kullanıldı. Bu modele train ve test verileri gönderilerek, nesnelerin özelliklerinin tanınması sağlandı. Bu bölüm MVC de Model bölümüne denk gelmektedir. Nesnelerin tanınmış bir şekilde kendilerine özel isimlerini kullandığımız bilgisayar ekranında görmekteyiz. Bu bölüm MVC de View bölümüne denk gelmektedir. MVC de Model ve View arasında iletişimi sağlayan Controller ise kamera görüntüsünü modele taşır, modelden aldığı veriyi ise Arduino ya gönderir. Bu işlemler sayesinde robotumuz kameradan aldığı verinin bilgisayarda Faster R-CNN tarafından tanınması sonucu bu nesneye karşılık gelen hareketin Arduino’ ya iletilmesiyle hareketini gerçekleştirmektedir.

C.4.2. UML Kullanarak Tasarımı Diyagramları Oluşturma

Projenin işleyişine göre tasarlanan Use Case Diagram Şekil 15’ te ki gibidir. Use case diyagramın amacı kullanım senaryolarını görselleştirmektir. Şekil 15’te gösterilen sistemde telefon kamerasından alınan görüntü sistemi başlatacak ve daha sonrasında anlık veri eğitilmiş modeli baz alarak nesne hakkında tahminde bulunacaktır ve bundan sonra robot hareketlerini gerçekleştirecektir.



Şekil 15. Use-Case Diyagramı

C.5. Veri Tabanı Tasarımı

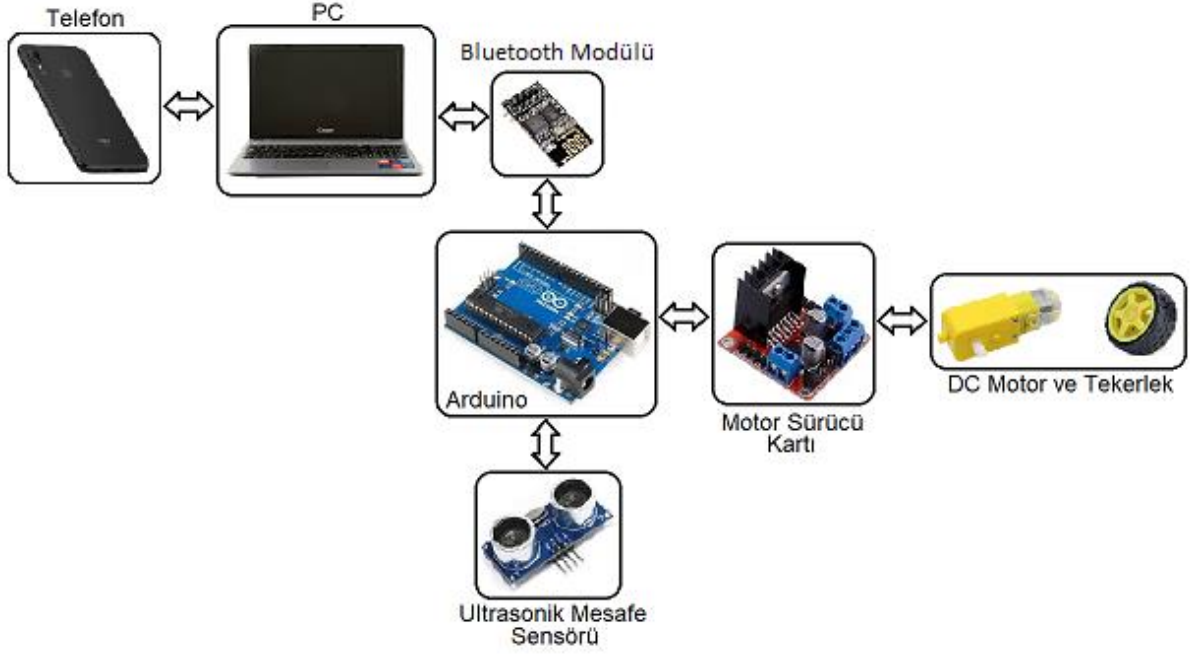
Yapılan görüşmeler sonucunda projede veri tabanına ihtiyaç duyulmamıştır.

C.6. Donanım Tasarımı

Projenin donanımsal tasarımları ve donanımsal parçaları sistem mekanik bileşen tasarımları, elektronik tasarımları ve gömülü sistem tasarımları başlıkları adı altında ayrıntılı olarak ele alınmıştır.

C.6.1. Sistem Mekanik Bileşen Tasarımlarının Verilmesi

Planlanan sistem 3 ana aşamadan oluşturulacaktır. Bunlardan ilki robotun üstüne ve kamerası yerdeki nesneleri görecektir şekilde yerleştirilen telefon üzerinde IP Webcam isimli mobil uygulama çalıştırılacak telefon kamerasından elde edilen anlık görüntüler bilgisayara aktarılacaktır. İkinci aşamada ise bilgisayarda daha önce eğitilen Faster R-CNN modelini alınan kamera görüntüsü üzerinde çalıştırılacak ve görüntüdeki nesnenin sınıflandırılması gerekecektir. Nesnenin kimlik numarası bulunduğundan sonra ise bu veri üçüncü aşamada robota bağlı bluetooth modülüne gönderilecektir. Bluetooth modülünden alınan veri ile robot yapması gereken hareketleri yapacaktır.



Şekil 16. Sistem Mekanik Bileşen Tasarımı Şeması

C.6.2. Elektronik Tasarım

Bilgisayar ile bluetooth modülü arasında veri aktarılırken kablosuz haberleşme kullanılır. Bluetooth modülü Arduino' ya seri bağlıdır ve gelen verileri Arduino' ya veri iletim kablolarıyla aktarır. Sensörle ilgili işlemin gerçekleştirildiği kısımda ise ultrasonik mesafe sensörü Arduino' nun pinlerine bağlantı kabloları kullanılarak fiziksel bağlantısı sağlanacaktır.

Tüm bunların sonucunda, elektronik tasarıma ihtiyaç duyulmadığından herhangi bir elektronik tasarım yapılmamıştır.

C.6.3. Gömülü Yazılım Tasarımları

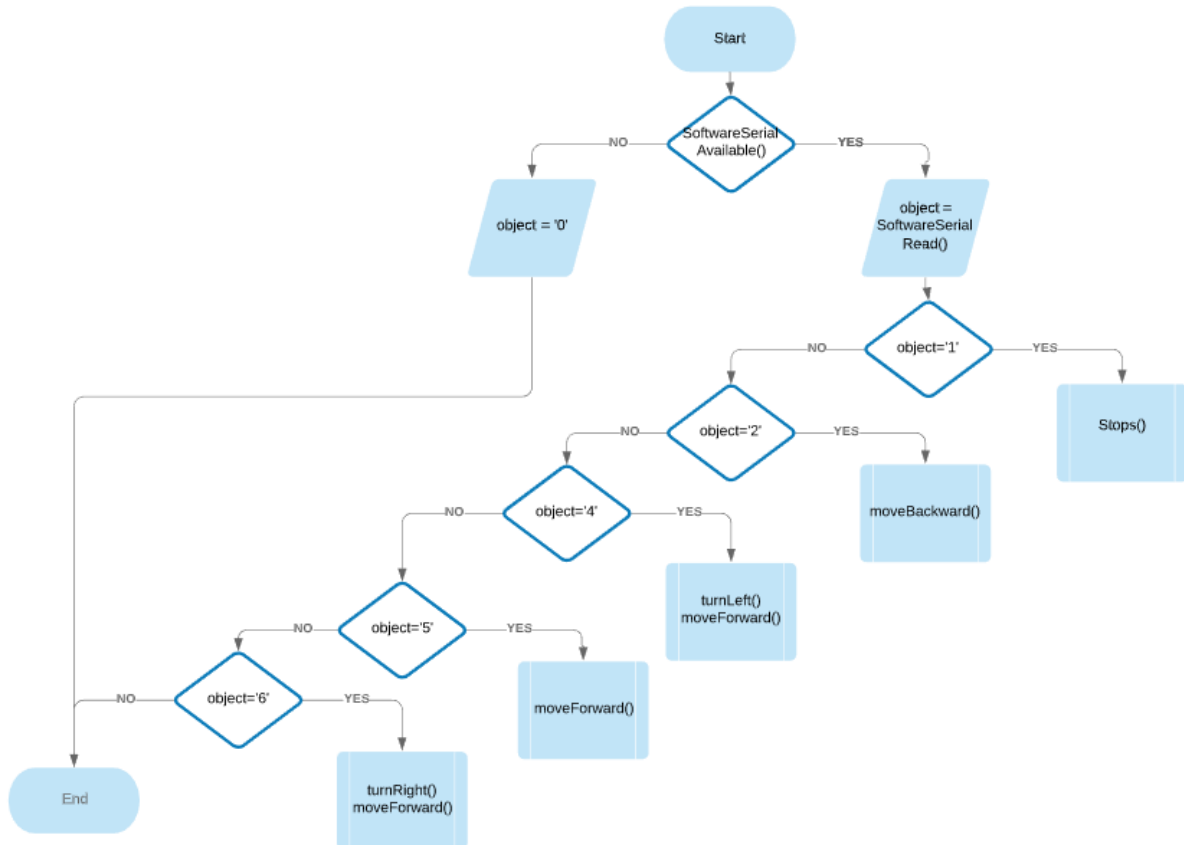
Proje kapsamında donanımsal bileşenleri kontrol edebilmek için Arduino kullanılacaktır. Arduino, elektrik donanım ve yazılım temelli bir geliştirme platformudur. Yani aslında donanım bileşenlerini kontrol eden yazılımları çalıştıran bir mikrodenetleyici kartıdır. Bu platform kullanılarak robotun hareket etmesini sağlayacak donanımlar ve mesafe sensörü kontrol edilir. Robotun nesnelere göre hareket etmesini sağlayan programın Arduino içerisine gömülmesi için Arduino ile bilgisayar arasında seri haberleşme kullanılır. Robotun verilere uygun hareket etmesini sağlayan program, bilgisayardan Arduino' ya USB üzerinden gönderilir. Programı aktarmak için Arduino IDE kullanılır. Bu Arduino' ya program yazmak ve yüklemek için kullanılan bir uygulamadır. Şekil 17' de Arduino' ya gömülen programı yazmak ve Arduino içerisine gömmek için kullanılan uygulamanın arayüzü bulunmaktadır.

```
Bitime_Codes | Arduino 1.8.9
File Edit Sketch Tools Help

Bitime_Codes $
#include <SoftwareSerial.h>
SoftwareSerial BTSerial(10, 11);
#define echoPin 9
#define trigPin 8
#define enA 7
#define in1 6
#define in2 5
#define in3 4
#define in4 3
#define enB 2
#define omuc 13
long times;
int object_code;
char object;
char x;
int y;
void setup() {
  Serial.begin(9600);
  BTSerial.begin(9600);
  pinMode(in1, OUTPUT);
  pinMode(in2, OUTPUT);
  pinMode(in3, OUTPUT);
  pinMode(in4, OUTPUT);
  pinMode(enA, OUTPUT);
  pinMode(enB, OUTPUT);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
}
1
```

Şekil 17. Arduino IDE Arayüzü

Gömülen programa baktığımızda robot bilgisayardan gelen verilere göre hareket edeceği için bilgisayardan veri gelmesini bekler. Bilgisayardan veri gelmediği durumda da robot durur vaziyette beklemektedir. Gelen veri nesnelerin kimlik numarasına göre farklı hareketler yapacaktır. Şekil 18’ de programın akış şeması gösterilmiştir.



D. UYGULAMA

D.1. Geliştirilen Sistemin Sistem Tasarımlarını Karşılanmasının Değerlendirilmesi

Robot üzerindeki kameradan gelen veri birer saniye arayla resim olarak alındı ve bu verileri IP Webcam uygulaması ile bilgisayar üzerinde görülebilmesi sağlandı. Derin öğrenme yöntemleri ile eğitilmiş sisteme bu veril csv dosyası üzerinden okunarak girdi olarak alındı, çıktı olarak algılanan nesnenin hangi hareketi işaret ettiği robota iletili. Robota iletilen hareket sayesinde robot hareketini gerçekleştirdi. Hareketinin sürekliliği veri gelene kadar devam etti ve kameranın yeni veriyi algılamasıyla hareketini gerçekleştirdi. Hareketin iletilmesi bluetooth yoluyla olmaktadır ve veri başarılı bir şekilde iletilmiştir.

D.2. Kullanıcı ve Sistem Ara yüzü Gerçeklemeleri

IP Webcam isimli mobil uygulamaya ilk girildiğinde karşılaşılan arayüz daha çok kamera ayarlarını içeren bir menüdür. IP Webcam uygulaması çalıştığında telefonun bağlandığı ağdan aldığı bir IP numarası ve bilgisayar ile telefon arasında iletişimi kuracak port numarası uygulamanın arayüzünde gösterilmiştir. Aynı ağda olan bilgisayar ile telefon arasındaki görüntü aktarımı yapabilmek için bir http url'si verilmiştir. Bu url yani adres kullanılarak birer saniye aralıklarla telefondan bilgisayara görüntüler gönderilmektedir. Görüntüler üzerinde eğitilen model çalıştırılır ve nesne tanıma işlemi gerçekleştirilir. Bunu yapmak için Python dili ile yazılmış programı çalıştırdığımızda nesnelerin kimlik numaralarını veren bir ekran çıkmaktadır.Şekil 20' de görüntü üzerinde algılanan nesnenin kimlik numarası gösterilmiştir.

Proje kapsamında asıl ilgilenilen arayüz, modelin görüntü üzerinde çalıştırılmasından sonra elde edilen görüntüdür. Şekil 21' de buna örnek bir arayüz görüntüsü vardır. Bu arayüzün önemli olmasındaki etken telefondan bilgisayara aktarılan verinin aktarılma süresi ve görüntünün kalitesidir.

IP Webcam isimli mobil uygulamadan alınan görüntü video şeklinde olduğunda bilgisayara aktarılırken görüntülerde donma ve gecikmeler yaşanmaktadır. Bu istenmeyen durumu ortadan kaldırmak için birer saniye aralıklarla anlık resim alınması görüntüdeki donmaları ve gecikmeleri engellemiştir.



Şekil 19. IP Webcam Mobil Uygulaması Kamera Kullanım Ekranı

```
Python 3.7.3 Shell
File Edit Shell Debug Options Window Help

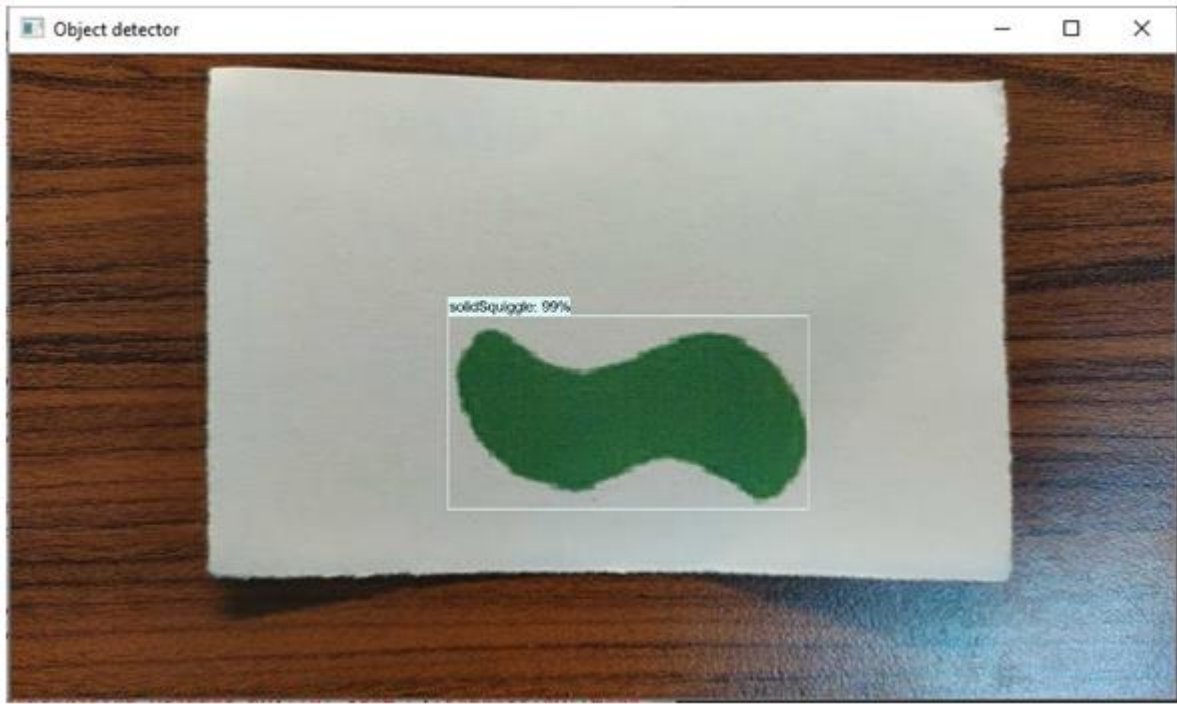
Warning (from warnings module):
  File "C:\Users\elfgnc\Anaconda3\lib\site-packages\tensorboard\compat\tensorflow_stub\dtypes.py", line 544
    _np_quint16 = np.dtype(["quint16", np.uint16, 1])
FutureWarning: Passing (type, 1) or 'ltype' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'
.
WARNING: Logging before flag parsing goes to stderr.
W0108 22:50:40.815937 9788 deprecation_wrapper.py:119] From C:\tensorflow1\models\research\object_detection\utils\label_map_util.py:137: The name tf.gfile.GFile is deprecated. Please use tf.io.gfile.GFile instead.

W0108 22:50:40.831527 9788 deprecation_wrapper.py:119] From C:\tensorflow1\models\research\object_detection\bbb.py:32: The name tf.GraphDef is deprecated. Please use tf.compat.v1.GraphDef instead.

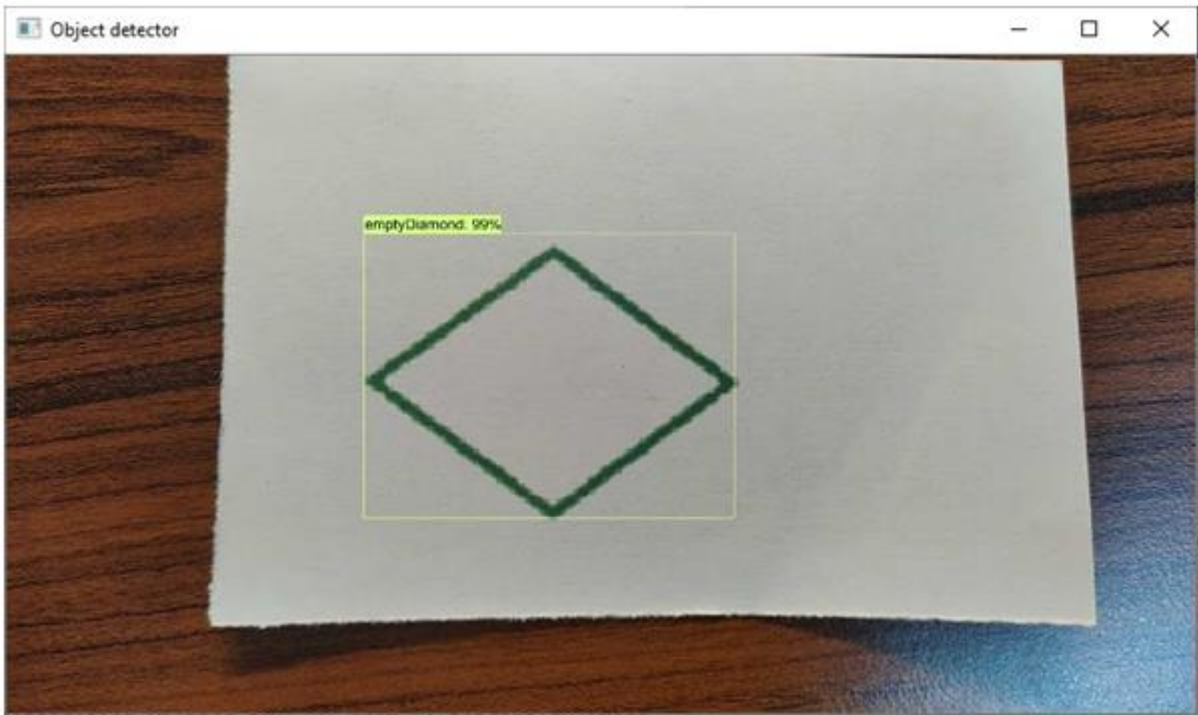
W0108 22:50:41.472001 9788 deprecation_wrapper.py:119] From C:\tensorflow1\models\research\object_detection\bbb.py:38: The name tf.Session is deprecated. Please use tf.compat.v1.Session instead.

5
>>> |
```

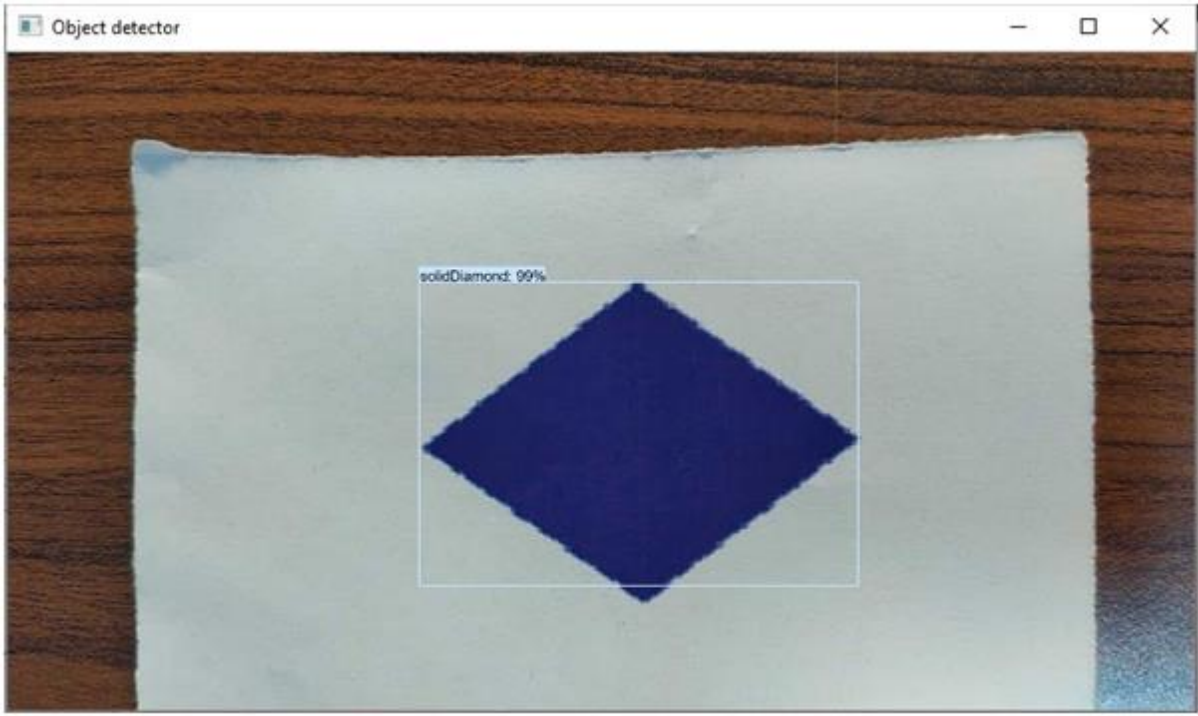
Şekil 20. Algılanan Nesnenin Kimlik Numarasını Gösteren Arayüz



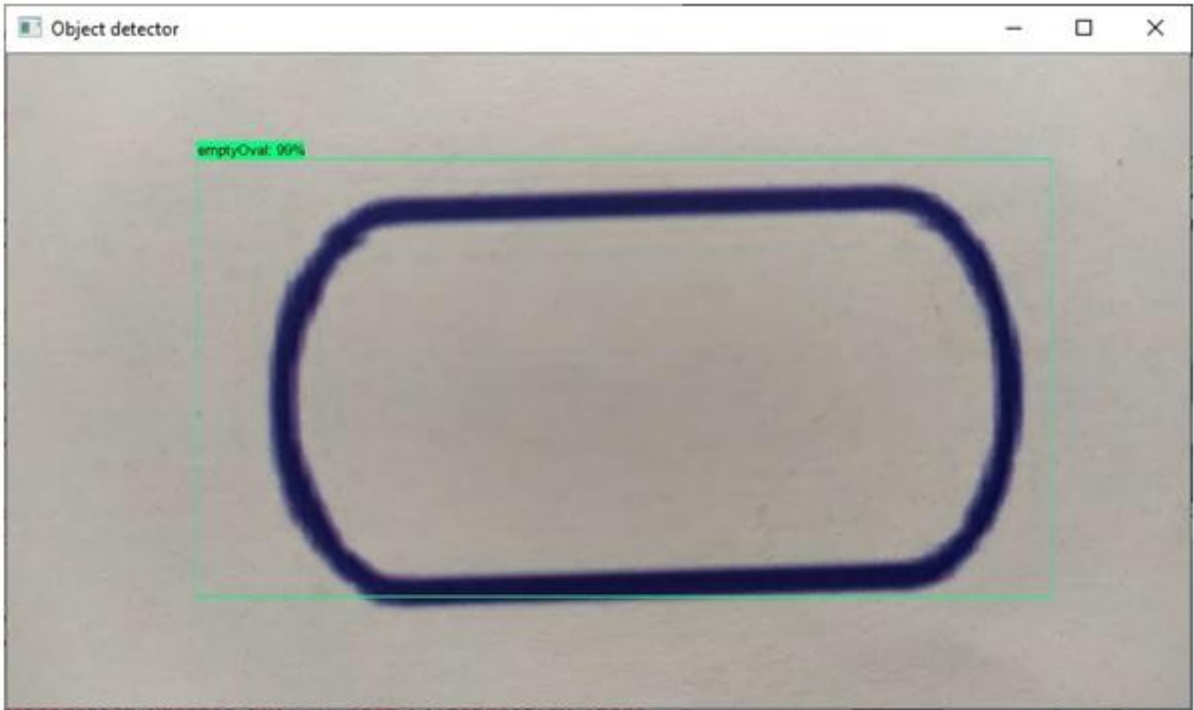
Şekil 21. Nesne Tanıma Uygulaması Örnek Arayüzü-1



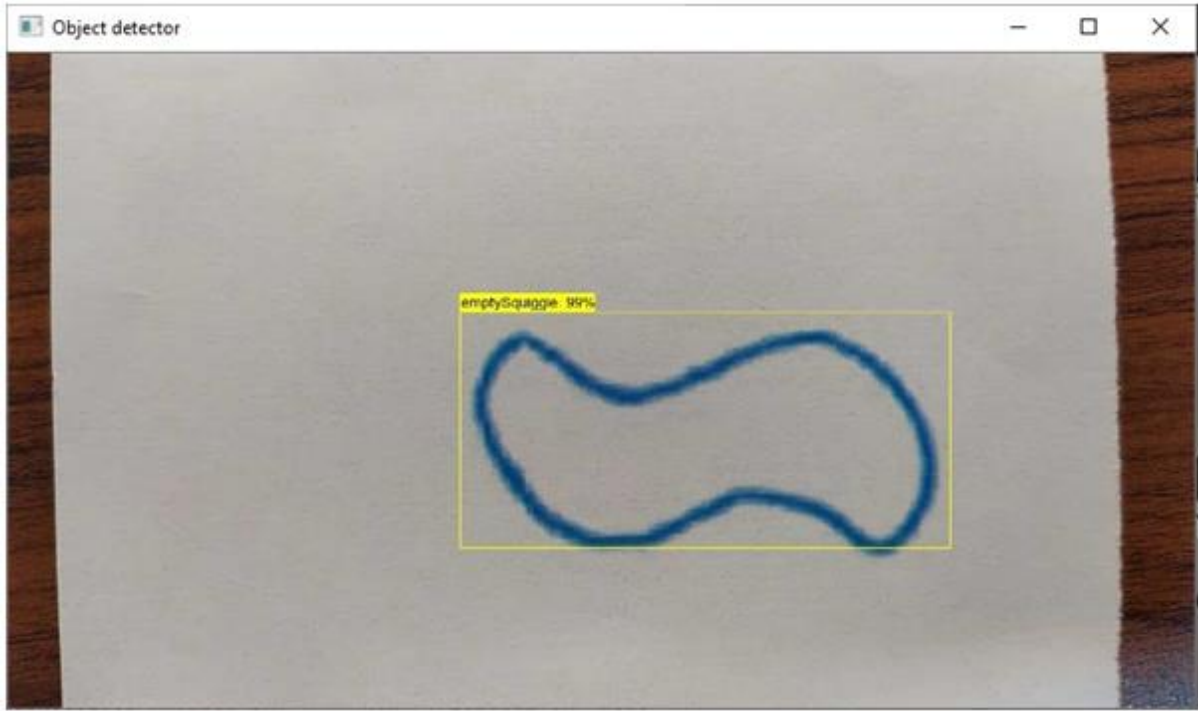
Şekil 22. Nesne Tanıma Uygulaması Örnek Arayüzü-2



Şekil 23. Nesne Tanıma Uygulaması Örnek Arayüzü-3



Şekil 24. Nesne Tanıma Uygulaması Örnek Arayüzü-4



Şekil 25. Nesne Tanıma Uygulaması Örnek Arayüzü-5

D.3. Gerçeklenen Testler

Geliştirilen arayüz için fonksiyonel test türlerinden olan White Box (Beyaz Kutu), Black Box (Siyah Kutu), Unit (Birim), Integration (Entegrasyon) ve Acceptance (Kabul) testleri gerçekleştirilmiştir. Bu testler sayesinde olası program hataları ve programın beklenmedik şekilde sonlanması gibi durumların önüne geçilmiştir. Gerçekleştirilen testler ve bu test sonuçlarının değerlendirilmesi aşağıda detaylandırılarak anlatılmıştır.

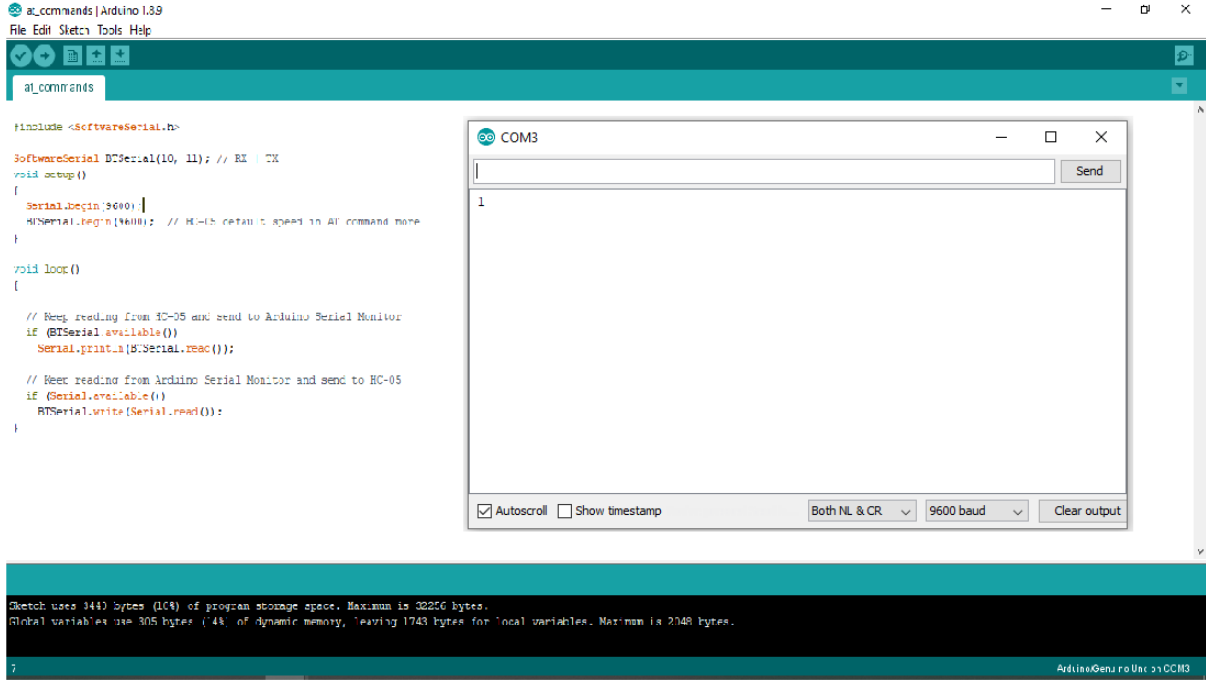
Proje kapsamında gerçekleştirilen testlere baktığımızda IP Webcam isimli mobil uygulamadan bilgisayara aktarılan görüntülerin kalitesi ve aktarılma süresi, modelin alınan görüntüler üzerinde çalışma süresi, verinin bilgisayardan Arduino'ya aktarılma süresi ve robotun gelen veriye göre gerçekleştirdiği hareketlerin performansı test edilmiştir.

Görüntü üzerinde yapılan testler ve test sonucuna göre yapılan değişiklikler “Kullanıcı ve Sistem Arayüzü Gerçeklemeleri” başlığı altında açıklanmıştır.

Nesnelerin algılanması ve robotun hareketleri için gerçekleştirilen testler aşağıda belirtilmiştir.

- **Nesne algılama ve tanıma işlemlerinin kontrolü:** Kameramızdan gelen görüntüler algılanmaktadır. Nesnenin bulanık veya karanlık olması nesne algılama için önemsizdir. Algılanan nesneler bilgisayar üzerinde Faster R-CNN modeli ile tanınmaktadır. Nesnelerin karanlık veya bulanık olması tanıma açısından sorun çıkartmamaktadır.

- **Nesnenin kimlik numarasının doğru aktarılması:** Yapılan ilk testte gönderilen veri ile okunan veri aynı değildi. Bu gönderilen sayı int tipinde iken Arduino veriyi okurken char tipinde okumaktadır. Char tipinden integer tipine dönüşüm yapsakta veri Arduino' ya gönderilen veri ile Arduino' dan okunan veriler aynı değildi. Daha sonra program üzerinde yapılan değişiklikler ile verilerin eşitliği sağlanmıştır. Şekil 26'da okunan veri gösterilmiştir.



Şekil 25. Arduino' ya Aktarılan Veri

- **Nesneleri tanıma mümkün olsa bile hareketin doğru şekilde gerçekleştirilebilmesi:** Nesnelerin tanınması yukarıdaki testte de söylediğimiz gibi makuldur. Tanıma işleminde herhangi bir sorun çıkmamaktadır. Tanınan nesne sahip olduğu harekete göre robotumuz hareket etmektedir. Komutlara doğru bir şekilde uymaktadır.
- **İlk nesneyi tanıdıktan sonra hareketin doğru bir şekilde gerçekleştirildiğinin gözlemlenmesi ardından diğer nesneye kadar karşılaşılan hatalar:** Oluşturduğumuz path üzerinde deneme testleri yaptık. İlk resmimiz ileri hareketiydi. Robotumuz ileri gitti fakat donanım tarafından kaynaklanan bir kaç sorun ile karşılaştık. Biz tamamen düz gitmesini beklerken robotumuz düz gitmedi. Bu sebeple diğer resmimizi algılayamadı ve sürekli ileri komutunda kaldı. Bir diğer karşılaşılan hata ise bulunduğumuz zeminin renginin ve deseninin tek olmamasından kaynaklıydı. Bizim resimlerimizi andıran ufak şekiller bulunmaktadı. Nesne algılama kısmında bunlar da tanındı ve robotumuz hangi hareket komutu geldiyse ona göre hareket etti ve 2. resme ulaşamadık.

- **İlk nesneyi algılayıp ikinci nesneye kadar hareketin doğru bir şekilde gerçekleşmesi:** Birden fazla deneme sonucu 2. resme herhangi bir sorunla karşılaşmadan ulaşıldı.
- **İkinci nesneye ulaştığında nesne algılama işleminin kontrolü:** 2. resme ulaşıldığında resmi algıladığını yaptığı hareketten test ettik. Nesneyi algıladığı testi başarı ile sonuçlandı.
- **İkinci nesneyi tanıdıktan sonra hareketin doğru bir şekilde gerçekleştirilebilmesi:** Yukarıda yaşadığımız bazı sorunları anlatmıştık. Bir de robotun hızından kaynaklı nesneyi okuyup ama nesneyi geçtikten sonra hareket etmesi gibi bir durumla karşılaştık. Resmi algılamadan geçmediği için resmi geçmiş olsa bile doğru hareketi yapmaktadır.
- **Oluşturulan yolda nesnenin algılanma kontrolü ve hareketin doğru gerçekleştirilebilmesi:** Diğer testlerde de bahsettiğim gibi oluşturduğumuz yolu tamamen doğru şekilde sadece birkaç kez gerçekleyebildik. Birden fazla deneme sonucu buna ulaştık. Resimler arası uzaklık, yerin uygunsuzluğu, robotun hızı gibi sebepler sonuca ulaşmamıza olanak sağlamadı.
- **Hareket sırasında olası durumların dışında gerçekleşen olaylara robotumuzun tepki süresi:** Zemin yüzünden yaşanan olaylar için hemen tepki verdi. Hızından kaynaklı nesne tanımadan sonra tepkiyi biraz geç verdi.
- **Sistem çalışmasında oluşabilecek koddan kaynaklı hataların testi:** İlk hatamız bu oldu. Robota gönderdiğimiz veriyi char değerinde gönderdik fakat Arduino kısmında onu int değerine çevirdik. Bu da hataya yol açtı. Robotun komutlara uymadığını ve verinin robota gönderilemediğini gördük. Bunu fark edilince kodu düzenledik ve robotumuz hareket etti. Robotun sağa ve sola dönmesi durumunda robotun 90 derece dönmediğini ve kodda bir hata olduğunu anladık.
- **Robotun hızı ve hıza göre nesne algılama yetisi:** Robotumuzun hızı fazladır ve bu yüzden test ederken zorlandık. Hızlı olmasına rağmen nesne algılamada bir sorun yaşanmamaktadır.
- **Birbiri ile entegre edilmiş şekilde çalışan sistem:** Herhangi bir sorun çıkmamıştır.
- **Robotun sağa, sola, ileri ve durma hareketlerini doğru bir şekilde gerçekleyebilmesi:** Hareketleri doğru bir şekilde gerçekleştirmektedir.
- **Sağa ve sola dönme durumlarında 90 derece dönebilmesini gerçeklenmesi:** Hata ile karşılaştık ilk başta 90 derece dönmemekteydi. Bu sorunu kodda düzelttik ve artık 90 derece dönmektedir.
- **Oluşturduğumuz path dışında nesne algılanması:** Zemindeki diğer nesneleri algılama işlemi olmuştur.
- **Farklı nesneleri algıladığında robotun tepkisi:** Robot hangi resmi algılıyorsa onun hareketine göre devam ediyor.

Karşılaşılan hatalar:

Robotun hareket ettiđi durumlarda karşılaşılan bir diđer şey ise çevre faktörlerinin robotu etkilemesidir. Bunlar zeminin robotun hareketlerini etkilemesi, robot üzerinde bulunan motor sürücü kartının tekerleklerle aynı seviyede enerji verememesi ve robot üzerindeki bileşenlerin ağırlığının robotun hareketinde sapmalara yol açmasıdır. Bunlar gerçekleştirilen testler sonucunda ortaya çıkmıştır.

D.4. Yazılım/Veri Tabanı/Donanım Gerçeklemeleri

Modelin alınan görüntüler üzerinde çalışma süresi, görüntünün kalitesine ve modelin eğitime bağılıdır. Modelin eğitimi proje için oldukça başarılıdır. Kalitesi kötü, donuk bir resimde bile nesneleri tanıyabilmektedir. Modelin iyi eğitilmesi kullanılan veri setinin büyük olmasından ve modeli eğitime süremizden kaynaklanmıştır. Yapılan bir diđer test ise verinin bilgisayardan Arduino' ya aktarılma aşamasıdır. Bu aşamada nesnelerin kimlik numarasını Arduino' ya doğru bir şekilde aktarılması oldukça önemlidir. Aktarılan verinin doğruluđu Arduino ve Python dili ile yazılan test programları ile test edilmiştir. Robotun hareket hızı gelen verinin hızına uygun olmalıdır. Bu, robot çok hızlı olursa verinin asıl gelmesi gereken yerde gelemediğini gösterir. Bu da istenmeyen bir durumdur. Motorların hareket etmesi için motor sürücü kartı ve enerji kaynağı olarakta lityum piller kullanılmıştır. Ancak kullanılan pillerin enerji seviyesinin en yüksek olduđu durumlarda da robotun çalıştığı minimum hız değeri proje kapsamında istenilen bir hız değeri değildir.

E. SONUÇ VE ÖNERİLER

Bu proje kapsamında kamera görüntüsünün aktarılması, nesneleri tanıma, bilgisayardan robota veri aktarma ve robotun uygun hareketleri yapması sağlanmıştır. Ancak robotun şu an ki donanımsal bileşenlerinin yetersiz olması sebebi ile yapılması istenen hareketlerin bazıları doğru olarak gerçekleştirilmemiştir. Bu bakımdan robot üzerinde donanımsal çalışmaların devam etmesi gerekmektedir. Robotun hareketleri sırasında fark edilen bir diđer eksik ise zeminde bulunan şekillerin, seçilen nesnelerin şekillerinden ayırt edilmesi gerektiğidir. Bu sorunu da zemin şekillerini modele negatif sınıf olarak göstererek çözülmesi düşünülmektedir. Bir de nesneler arası uzaklığın belli olup robotun oraya kadar hiçbir şekilde nesne algılaması yapmaması düşünülmektedir. Uzaklığın tamamlanmasıyla nesne okuma işlemi devam etmeli ve hareket gerçekleşmesi düşünülmektedir.

KAYNAKÇA

- [1]Jung, Sunggoo, Sunyou Hwang, Heemin Shin, and David Hyunchul Shim.”Perception, Guidance, and Navigation for Indoor Autonomous Drone Racing Using Deep Learning” IEEE Robotics and Automation Letters (Volume: 3 , Issue: 3 , July 2018) : 2539 - 2544
- [2]Marina Paolanti, Mirco Sturari, Adriano Mancini, Primo Zingaretti, Emanuele Frontoni “Mobile robot for retail surveying and inventory using visual and textual analysis of monocular pictures based on deep learning” 2017 European Conference on Mobile Robots (ECMR)
- [3]Constantin Purcaru, Radu-Emil Precup, Daniel Ierican, Lucian-Ovidiu Fedorovici, Bogdan Dohangie “Nrobotic mobile robot navigation using traffic signs in unknown indoor environments” 2013 IEEE 8th International Symposium on Applied Computational Intelligence and Informatics (SACI)
- [4]Xintao Ding, Yonglong Luo, Qingying Yu, Qingde Li, Yongqiang Cheng, Robert Munnoch, Dongfei Xue “Indoor object recognition using pre-trained convolutional neural network“ 2017 23rd International Conference on Automation and Computing (ICAC)
- [5]Lei Zhang, Fan Yang, Yimin Daniel Zhang, Ying Julie Zhu “Road crack detection using deep convolutional neural network” 2016 IEEE International Conference on Image Processing (ICIP)
- [6]Harsh Sinha ; Jay Patrikar ; Eeshan Gunesh Dhekane ; Gaurav Pandey ; Mangal Kothari “Convolutional Neural Network Based Sensors for Mobile Robot Relocalization” 2018 23rd International Conference on Methods & Models in Automation & Robotics (MMAR)
- [7]Wenhua Shao, Haiyong Luo, Fang Zhao, Yan Ma, Zhongliang Zhao, Antonino Crivello “Indoor Positioning Based on Fingerprint-Image and Deep Learning” IEEE Access (Volume: 6) :74699 - 74712