

T.C.
KÜTAHYA Dumlupınar Üniversitesi
Mühendislik Fakültesi
Elektrik-Elektronik Mühendisliği Bölümü



PLAKA TANIMLI OTOMATİK GARAJ KAPISI LİSANS BİTİRME TEZİ

HAZIRLAYAN

AHMETCAN TOPÇUOĞLU

201513151015

TEZ DANIŞMANI

PROF. DR. HAMDİ MELİH SARAOĞLU

KÜTAHYA, 2021

ÖZET

Plaka Tanımlı Otomatik Garaj Kapısı tezim,sürücüye konfor sağlamak amacı ile yapılmış,manuel sistemin önüne geçmek için hazırlanmıştır.Gelişen teknoloji ile birlikte araç sahiplerinin kendi evlerinin garajlarına geldikleri zaman araçlarından inerek garajlarını manuel olarak açmaları veyahut araçlarının içlerinde garaj kapılarının kumandalarını aramaları dönemin teknolojisinin oldukça gerisinde kalmıştır.Bu sebeple geliştirdiğim projede araç sahiplerinin araçları plaka okuyucu sistem sayesinde algılanarak araç sahiplerine kapılarını açarak zaman ve işten tasarruf sağlamaktadır.

Plakaların garaj sistemine aktarılacak ,sürücülerin ekstra çaba harcamadan garajlarına geldikleri zaman,plaka okuyucu sayesinde algılanarak kapının sürücünün herhangi bir efor sarfetmek sizin rahatça açıldığı sistemdir.Bu çalışma kapsamında sürücünün kişisel araç plakası sistem tarafından algılanarak eğer sisteme kayıtlı bir araçsa yeşil ışık yakarak ve oled ekrana plakanın adını yansıtılarak sürücüye garaj kapısının açıldığı sistemdir.Yabancı bir plakalı araç durumunda ise sistem alarm vererek kırmızı ledi yakar ve aracın sisteme kayıtlı olmadığını oled ekranda yansıtır.

ABSTRACT

The License Plate Defined Automatic Garage Door thesis was made to provide comfort to the driver and was prepared to prevent the manual system. With the developing technology, when the vehicle owners get out of their cars and open their garages manually or call the controls of the garage doors inside their vehicles, it is far behind the technology of the period. For this reason, in the project I have developed, the vehicles of the vehicle owners are detected by the plate reader system and the vehicle owners save time and work by opening their doors.

It is a system where the license plates are transferred to the garage system and when the drivers come to their garages without extra effort, it is detected by the plate reader and the door is opened easily without any effort. It is the system in which the garage door is opened to the driver by reflecting its name. In the case of a foreign vehicle, the system gives an alarm and turns on the red led and it reflects on the OLED screen that the vehicle is not registered in the system.

TEŞEKKÜR

Bu tezin çalışmasında sağladığı katkılardan ötürü tez danışmanım Prof. Dr. Hamdi Melih Saraoğlu'na ve Arş.Gör. Evin Şahin Sadık'a sonsuz teşekkürlerimi sunarım.

Tez çalışması süresi boyunca beni sürekli destekleyerek yanımda olan çok değerli aileme ve sevgili arkadaşlarıma teşekkür ederim.

İÇİNDEKİLER

	<u>Sayfa</u>
ÖZET.....	I.
SUMMARY.....	II.
TEŞEKKÜR	III.
ŞEKİLLERİN DİZİNİ	VI.
1.GİRİŞ.....	1
2.PLAKA ALGILAYAN SİSTEM.....	2
2.1.Plaka Nedir ?	2
2.2.Plaka Tanımlama Sistemi Nedir ?.....	3
2.3.Görüntü İşleyerek Plaka Tanımlama.....	3
2.3.1.Alınan Görüntüde Plaka Bölgesinin Bulunması	4
3. GÖRÜNTÜ İŞLEME NEDİR ?	5
3.1.Görüntü İşleme.....	5
3.2.Görüntü İşleme ile Yapılabilecek İşlemler.....	5
3.2.1.Görüntü İşleme Kullanım Alanları.....	6
3.3 Görüntü İşlemenin Kullanıldığı Platformlar.....	8
3.3.1.Open CV Nedir ?	8
3.3.2.Open CV Bileşenleri.....	9
4. YOLO NEDİR ?	10
4.1.Yolo Çalışma Prensipleri.....	10

İÇİNDEKİLER(Devam)

5.PYTHON.....	12
5.1.Spyder Arayüzü.....	12
5.2 Trafik Levhalarını Algılayan Sistemin Spyder4 Arayüzü ile Pythom Dilinde Kodlanması.....	13
5.2.1. Kodların İçeriklerinin Aktarılması.....	13
6.ARDUİNO.....	17
6.1.Arduino Çeşitleri.....	17
6.2.Arduino Uno.....	18
6.2.1. Arduino Programındaki Başlıca Kodlar.....	20
6.3. Plaka Tanımlı Otomatik Garaj Sistemi Arduino Program Kodları.....	21
7.RGB LED.....	25
8.OLED EKRAN.....	26
9.BUZZER.....	27
9.1.Çalışma Şekli.....	27
10.SERVO MOTOR.....	28
10.SONUÇ.....	29
11. KAYNAKLAR.....	30

ŞEKİLLER DİZİNİ

	<u>Sayfa</u>
Şekil 2.1(a) Plaka Görüntüsü.....	2
Şekil 2.2 Araç Görüntüsü.....	3
Şekil 2.3(a) Kenar Bulma Algoritması Uygulanmış Resim.....	4
Şekil 2.3(b) Plaka Bölgesi Bulunmuş Resim.....	4
Şekil 3.1 Görüntü İyileştirme.....	6
Şekil 3.2 Cisim Tanıma.....	6
Şekil 3.3 Kanser Tanısı.....	7
Şekil 3.4 Open CV Logo.....	9
Şekil 4.1Boundin Box.....	10
Şekil 4.2 Nesnelerin Tanımlanması.....	11
Şekil 5.1 Spyder Arayüzü.....	12
Şekil 5.2 Python komutlarının Girdisi.....	13
Şekil 5.3 Kütüphane Girdisi.....	13
Şekil 5.4 Port Haberleşme.....	14
Şekil 5.5 Layers.....	14
Şekil 5.6 Python Kodlar.....	14
Şekil 5.7 Bounding Box.....	15

ŞEKİLLER DİZİNİ(Devam)

Şekil 5.8 Değerlerin Çıktısı.....	16
Şekil 6.1 Arduino Uno.....	18
Şekil 6.2 Arduino Uno Pin Değerleri.....	19
Şekil 6.3 Arduino Uno Sistem Bilgisi.....	20
Şekil 6.4 Kütüphane.....	21
Şekil 6.5 Şablon.....	22
Şekil 6.6 Pin Ataması.....	23
Şekil 6.7 Pin Değerleri.....	23
Şekil 6.8 Çıktılar.....	24
Şekil 7.1 RGB Led Yapısı.....	25
Şekil 8.1 128*64 Oled Datasheet.....	26
Şekil 9.1 Buzzer.....	27
Şekil 10.1 Servo Motor.....	28
Şekil 11(a) Araç Plaka Tanımlama.....	29
Şekil 11(b) Araç Plaka Tanımlama.....	29

BÖLÜM 1

GİRİŞ

Teknolojinin gelişmesiyle birlikte insan hayatının yaşamsal olaylarında doğru orantıda kolaylık sağlanmıştır,pek çok üretici yaşamı kolaylaştırma adına argeler oluşturup piyasaya yaşamsal faaliyetleri kolaylaştıracak teknolojik ürünler sunmaktadır.Benim çalışmalarına benzeyen bir çok plaka algılama sistemide mevcuttur.Bende bu çalışmamda görüntü işleme ile plaka okumayı birleştirerek , kişiye özel garaj kapısı projesi geliştirerek insanların hayatlarını kolaylaştırmayı hedefledim.Görüntü işleme yöntemi ile sisteme tanıtılan kişisel plaka okunarak,kişisel olan evin garaj kapısını devreye geçiren sistem yapmayı hedeflemekteyim.

Bu sistem sayesinde insanların araçlarının daha rahat giriş çıkışının sağlanması,garajlarında daha güvende korunması sağlanmaktadır.Görüntü işleme ile oluşturulan bu sistem hali hazırda olan bütün garaj kapılarına entegre edilebilir.Kullanıcı siparişlerine göre plaka sisteme tanımlanıp garaj kapıları plaka tanımlı otomatik garaj kapısı haline gelebilir.

BÖLÜM 2

PLAKA ALGILAYAN SİSTEM

2.1. Plaka Nedir ?

Plaka tüm Türkiye ve dünyada, otomobil, otobüs, kamyonet, kamyon, taksi ve motosiklet gibi taşıtlara verilen numara ve harflerdir. [İlk plaka](#) 1893 yılında Fransa'da kullanılmaya başlanmıştır. 30 km'den fazla hız yapan tüm araçlara sol tarafta olmak ve her zaman görünür durumda olması şartıyla konulmuş ve zamanla otomobil sayısının tüm dünyada artması ile yaygınlaşmıştır. Her ülkenin kendine ait bir plaka sıralaması vardır. Türkiye'de bu sıralamada ilk iki sayı şehrin kodu dur. Plakalar aracın bir nevi kimliğidir ve plaka sayesinde birçok işleminizi kolayca yapabilirsiniz.



Şekil 2.1 Plaka Görüntüsü

2.2. Plaka Tanımlama Sistemi Nedir ?

Otomatik plaka tanıma sistemi; kameraları, veri işleyicilerini ve harf/rakam yani karakter okuma yazılımını entegre eden bir bütüncül sistemdir. Günümüzde yapay zeka (AI) ve özdevimli öğrenme (machine learning) teknolojileri ile, otomatik okunan plaka verileri daha hızlı ve daha verimli şekilde hem kullanılabilmekte hem de depolanabilmektedir. Ayrıca plaka bilgilerinin yanı sıra, plakaya bağlı GPS verilerinin de kaydedilmesi depolanan dosya boyutunda fazla bir artışa sebep olmazken, otomatik plaka tanıma teknolojisinin kullanım performansı açısından ciddi katkılar sağlamaktadır.

Geçmişte ortaya çıkan araç hızı, ışık oranı, görüntü alma derecesi ve farklı plaka karakter yazı tipleri gibi engeller, günümüzde algoritma teknolojinin hızla ilerlemesi ile çok yüksek oranda ortadan kalkmıştır. Ayrıca [IP kamera](#) teknolojisinin gelişimi ile de otomatik plaka tanıma sistemlerinin kullanım alanları yaygınlaşmıştır. Görüntü sensörlerinin daha yüksek çözünürlükte bilgi sunması ve plaka okuma kamera sistemlerinin daha hızlı görüntü işlemesi ile performansta ciddi bir artış olmuştur. İlk ortaya çıktığında yaklaşık %60 oranında doğru tespit sağlarken, bugün bu oran %96'ya ulaşmıştır.

2.3. Görüntü İşleyerek Plaka Tanımlama

Şekil 2.2' de Kamradan alınan renkli araç görüntüsü ve bu görüntünün gri tonlamalı hale dönüştürülmüş hali görülmektedir.



Şekil 2.2 Araç Görüntüsü

2.3.1 Alınan görüntüde plaka bölgesinin bulunması

Türkiye trafik denetleme kurumlarına kayıtlı yasal plakalar; sivil, resmi, askeri, diplomatik vs. gibi değişik tiplerde ve her tip değişik renk ve formatta olmaktadır (Trafik Denetleme, 2010). Bu çalışmada da, Türk plaka standartlarına uyan sivil plakaların tanınması amaçlanmıştır. Bu plakaların genel özelliği; beyaz zemin üzerine siyah karakterlerden oluşması, ilk iki karakterde şehir kodunu belirten rakamların olması ve ondan sonra gelen karakterlerin rastgele harf ve rakam dizisinden oluşmasıdır. Gri tonlamalı hale getirilmiş resimde plakanın bulunması için ilk olarak resme kenar buldurma (Edge Detection) algoritması uygulanmıştır. Kenar bulma algoritmasında; sağdan sola ve yukarıdan aşağıya taramalar yapılarak iki piksel arasındaki renk farkına bakılmış ve bu fark belli bir eşik değerden fazla ise o bölge beyaz olarak belirlenmiştir. Eğer belli bir değerden az bir renk geçişi var ise o bölge siyah olarak belirlenmiştir. Bu sayede kenarlar belirlenmiştir.



Şekil 2.3(a)' da kenarları bulunmuş bir görüntü gösterilmektedir.

Çalışmada kenar bulma algoritması yapılan resimde en yoğun olan bölge (beyaz noktaların en fazla olduğu bölge) plaka bölgesi olarak belirlenmiştir ve bu noktaların koordinatları belirlenerek resimden kesilip alınmıştır. Bu durumda plaka bölgesi bulunmuş bir resim Şekil 2.3(b) de gösterilmiştir.



Şekil 2.3 (a) Kenar bulma algoritması uygulanmış resim

Şekil 2.3(b) Plaka bölgesi bulunmuş resim

Farklı açılardan ve günün değişik zamanlarında plaka bölgesi bulunmuş bu görüntülerde ışığın yoğunluğuna, yönüne ve bir takım etkenlere bağlı olarak resimlerin çözünürlüklerinde bir farklılık görüldüğünden dolayı görüntülere netleştirme algoritmaları uygulanmıştır. Görüntüleri netleştirmek için Kontrast Genişletme ve Ortanca (Medyan) Filtresi algoritmaları uygulanmıştır.

BÖLÜM 3

GÖRÜNTÜ İŞLEME NEDİR ?

3.1 Görüntü İşleme

Görüntü işleme, elimizde bulunan görüntüden anlamlı ifadeler çıkarmamıza yarayan işlemler bütünüdür. Bu işlemler, görüntüyü oluşturan pikseller üzerinde gerçekleştirilecek matematiksel işlemler sayesinde gerçekleştirilir. Görüntü elde edildikten sonra, yapılması istenen göreve göre bir algoritma tasarlanır ve görüntü bu aşamalardan geçerek istenen görevi yerine getirir.

3.2 Görüntü İşleme ile Yapılabilecek İşlemler

Görüntü işleme teknikleri kullanılarak yapılabilecek işlemlere göz atacak olursak şunları görebiliriz;

- Görüntü üzerinde bulunan gürültülerin arındırılması ve temiz bir görüntü elde edilmesi
- Görüntü üzerinde bulunan ve insan algısının görmekte zorlandığı nesnelerin tespiti
- Görüntünün daha kaliteli bir hale getirilmesi
- Nesne takibi yapılması
- Görüntü üzerindeki farklı nesnelerin birbirinden ayırt edilmesi

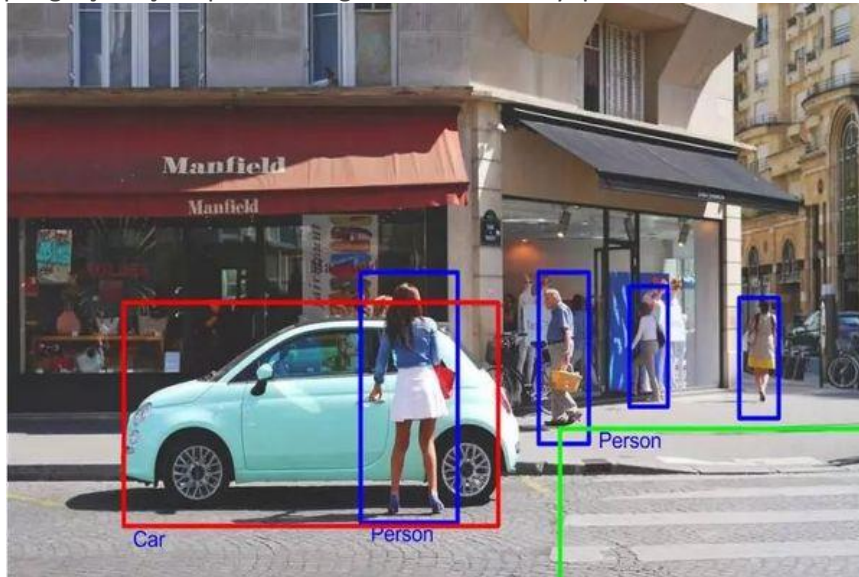
3.2.1 Görüntü İşleme Kullanım Alanları

- **Görüntü İyileştirme:** Alınan görüntülerde gürültü olarak adlandırılan ve görüntü üzerinde bozulmalara neden olan bazı istenmeyen yapılar bulunabilir. Bu gürültülere örnek olarak tuz-biber gürültüsü, gauss gürültüsü, shot gürültüsü verilebilir. Görüntü işleme teknikleri içerisinde bulunan mean (ortalama) filtre, medium (ortanca) filtre gibi tekniklerle görüntü daha kaliteli ve gürültüsüz hale getirilebilir. Bu sayede görüntü üzerinde daha doğru sonuçlar elde edilecektir.



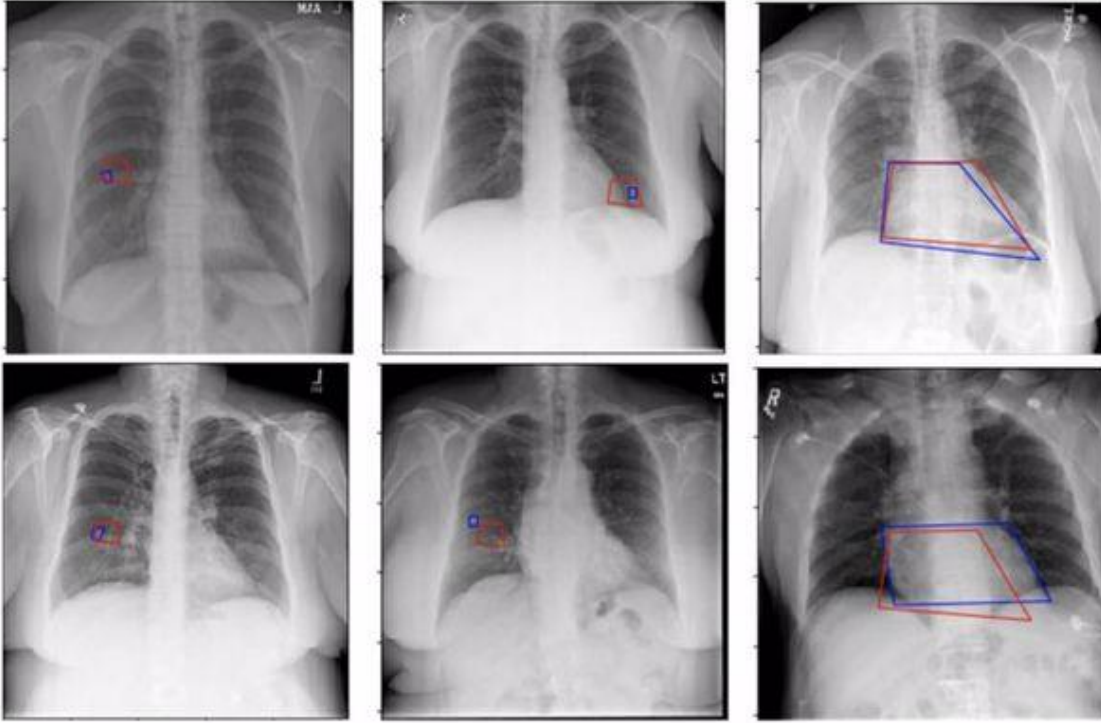
Şekil 3.1 Görüntü İyileştirme

- **Cisim Tanıma:** Tespit edilecek cisme göre gerekli yöntemler ve algoritmalar kullanılarak görüntü üzerinden herhangi bir cismin tespiti ve takibi gerçekleştirilebilir. Örneğin yurt dışında bir çok ülkede suçluların tespiti bu yöntem ile gerçekleştirilmektedir. Mevcut olan kamera düzeneklerinden alınan görüntüler üzerinden her hangi bir insanın tespiti sağlanabilir. Bunun dışında trafik alanında da kullanımı mevcuttur. Trafik içerisinde bulunan araçları sayabilir ve araçların hızı ölçülebilir. Bu sayede trafik yoğunluğu olma ve ya aşırı hız yapma gibi durumların tespiti gerçekleştirilip merkeze gerekli bildirimler yapılabilir.



Şekil 3.2 Cisim Tanıma

- **Sağlık Sektörü:** Görüntü işleme teknikleri sayesinde bir çok hastalığın teşhisi gerçekleştirilebilmektedir. Doğum öncesi fetüsün oluşumu ve takibi, tıbbi görüntülerin incelenmesi ,şüpheli dokuların belirgin hale getirilip uzmanlara doğru tanı koyabilme olanağı tanınması, meme kanserinin erken teşhisi gibi alanlarda görüntü işleme teknikleri kullanılmaktadır. Bunların yanı sıra beyin görüntüleme, kemik şeklinin ve yapısının analizi, kanser tanısı koyma ve tümörü fark etme gibi işlemlerde tıp biliminde kullanılabilmektedir.



Şekil 3.3 Kanser Tanısı

- **Savunma Sanayi:** İnsansız hava araçları, görüntü ile hedef takibi yapan roketler gibi araçların bünyesinde bulunan donanımlar, görüntü işleme sonucu elde edilen veriler doğrultusunda hareket gerçekleştirir.

Diğer alanlardan bazıları ise şu şekildedir:

- Uydu görüntüleri üzerinden nüfus yoğunluğu, çevre kirliliği gibi çevresel durumların tespiti
- Hava Gözlem Ve Tahmin
- Güvenlik Sistemleri
- Kriminal Laboratuvarlar
- Uzaktan Algılama Sistemleri

3.3 Görüntü İşlemenin Kullanıldığı Platformlar

MATLAB: Mühendislik uygulamalarında pek çok farklı alanda kullanılan MATLAB, bünyesinde temel görüntü işleme fonksiyonlarına da sahiptir. Kapsam olarak, direkt görüntü işleme yazılımı olmaması sebebiyle fazla kapsamlı değildir.

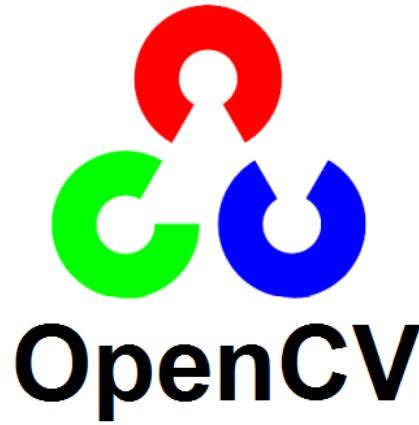
OpenCV: Günümüzde görüntü işleme çalışmalarında en aktif kullanılan kütüphanelerin başında gelmektedir. İçerisinde bir çok hazır fonksiyon bulundurması sebebiyle hızlı bir şekilde proje gerçekleştirilebilir. Aynı zamanda C++, Java, Python, C#, MATLAB gibi farklı programlama dilleri içerisinde kullanılabilmesi en büyük avantajlarından biridir. Aynı zamanda açık kaynak kodlu olup tamamen ücretsizdir.

Halcon: Makine görmesi odaklı ticari bir yazılımdır. İçerisinde bulunan hazır fonksiyonlar ile kolayca proje gerçekleştirilebilir.

Fiji: Java platformu için geliştirilmiş açık kaynak kodlu bir görüntü işleme kütüphanesidir. Bilimsel görüntü analizi için geliştirilmiştir. Aynı zamanda genetik, hücre biyolojisi, nöro-bilim gibi alanlar için özelleştirilmiş algoritmalara sahiptir.

3.3.1 Open Cv Nedir ?

OpenCV (Open Source Computer Vision) açık kaynak kodlu görüntü işleme kütüphanesidir. 1999 yılında Intel tarafından geliştirilmeye başlanmış daha sonra Itseez, Willow, Nvidia, AMD, Google gibi şirket ve toplulukların desteği ile gelişim süreci devam etmektedir. İlk sürüm olan OpenCV alfa 2000 yılında piyasaya çıkmıştır. İlk etapta C programlama dili ile geliştirilmeye başlanmış ve daha sonra birçok algoritması C++ dili ile geliştirilmiştir. Open source yani açık kaynak kodlu bir kütüphanedir ve BSD lisansı ile altında geliştirilmektedir. BSD lisansına sahip olması bu kütüphaneyi istediğiniz projede ücretsiz olarak kullanabileceğiniz anlamına gelmektedir. OpenCV platform bağımsız bir kütüphanedir, bu sayede Windows, Linux, FreeBSD, Android, Mac OS ve iOS platformlarında çalışabilmektedir. C++, C, Python, Java, Matlab, EmguCV kütüphanesi aracılığıyla da Visual Basic.Net, C# ve Visual C++ dilleri ile topluluklar tarafından geliştirilen farklı wrapperlar aracılığıyla Perl ve Ruby programlama dilleri ile kolaylıkla OpenCV uygulamaları geliştirilebilir.



Şekil 3.4 Open CV Logo

OpenCV kütüphanesi içerisinde görüntü işlemeye (image processing) ve makine öğrenmesine (machine learning) yönelik 2500’den fazla algoritma bulunmaktadır. Bu algoritmalar ile yüz tanıma, nesneleri ayırt etme, insan hareketlerini tespit edebilme, nesne sınıflandırma, plaka tanıma, üç boyutlu görüntü üzerinde işlem yapabilme, görüntü karşılaştırma, optik karakter tanımlama OCR (Optical Character Recognition) gibi işlemler rahatlıkla yapılabilmektedir.

3.3.2 Open Cv Bileşenleri

- **Core:** OpenCV’nin temel fonksiyonları ve matris, point, size gibi veri yapılarını bulundurur. Ayrıca görüntü üzerine çizim yapabilmek için kullanılabilecek metotları ve XML işlemleri için gerekli bileşenleri barındırır.
- **HighGui:** Resim görüntüleme, pencereleri yönetme ve grafiksel kullanıcı arabirimleri için gerekli olabilecek metotları barındırır. 3.0 öncesi sürümlerde dosya sistemi üzerinden resim dosyası okuma ve yazma işlemlerini yerine getiren metotları barındırmaktaydı.
- **Imgproc:** Filtreleme operatörleri, kenar bulma, nesne belirleme, renk uzayı yönetimi, renk yönetimi ve eşikleme gibi neredeyse tüm fonksiyonları içine alan bir pakettir. 3 ve sonra sürümlerde bazı fonksiyonlar değişmiş olsada 2 ve 3 sürümünde de bir çok fonksiyon aynıdır.
- **Imgcodecs:** Dosya sistemi üzerinden resim ve video okuma/yazma işlemlerini yerine getiren metotları barındırmaktadır.
- **Videoio:** Kameralara ve video cihazlarına erişmek ve görüntü almak ve görüntü yazmak için gerekli metotları barındırır. OpenCV 3 sürümü öncesinde bu paketteki birçok metot video paketi içerisindeydi.

BÖLÜM 4

YOLO NEDİR ?

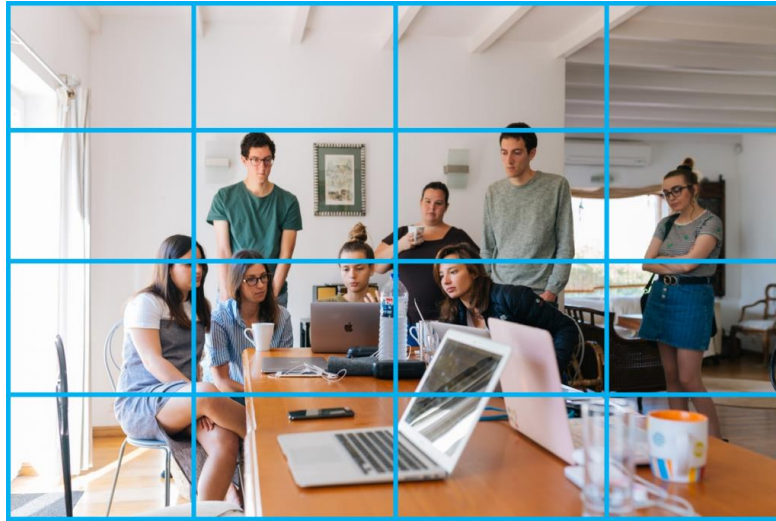
YOLO, konvolüsyonel sinir ağlarını (CNN) kullanarak nesne tespiti yapan bir algoritmadır. Açılımı “**You Only Look Once**”, yani “**Sadece Bir Kez Bak**”. Bu adın seçilmesinin nedeni algoritmanın nesne tespitini tek seferde yapabilecek kadar hızlı olmasıdır. YOLO algoritması çalışmaya başladığında görüntülerdeki veya videolardaki nesneleri ve bu nesnelerin koordinatlarını aynı anda tespit eder.

4.1 Yolo Çalışma Prensibi

Video ve resim işleme arasında tek fark resimlerin tek bir kareden (frame), videoların ise birçok kareden oluşmasıdır. Resimlerde algoritma tek bir kare için çalışırken, videolarda tüm kareler için tekrar tekrar çalışır. . YOLO algoritması, öncelikle görüntüyü bölgelere ayırır. Daha sonra her bir bölgedeki nesneleri çevreleyen kutuları (bounding box) çizer ve her bir bölgede nesne bulunma olasılığı ile ilgili bir hesaba yapar.

Ayrıca her bir bounding box için bir güven skoru hesaplar. Bu skor bize o nesnenin yüzde kaç olasılıkla tahmin edilen nesne olduğunu söyle. Örneğin, bulunan bir araba için güven skoru 0,3 ise bunun anlamı o nesnenin araba olma olasılığının oldukça düşük olduğudur. Diğer bir deyişle, YOLO yaptığı tahminin güvenilirmez olduğunu bize söyler.

Bounding box’ların içindeki nesnelere **non-maximum suppression** denen bir teknik uygulanır. Bu teknik güven skoru düşük olan nesneleri değerlendirmeden çıkarır ve aynı bölgede güven skoru daha yüksek bir **bounding box**’ın varlığını kontrol eder.



Şekil 4.1Boundin Box

Her bir bölgede nesne olup olmadığı araştırılır. Eğer bir nesne bulunursa o nesnenin orta noktası, yüksekliği ve genişliği bulunur daha sonra bounding box çizilir. Bunun yapılabilmesi için bir takım alt işlemlerin yapılması gerekir. Her bir bölge için bir tahmin vektörü oluşturulur, bu vektörlerin içinde güven skoru yer alır.

Eğer güven skoru 0 ise orada nesne yok, 1 ise orada nesne var demektir. Aynı içerisindeki aynı nesne için birden fazla bounding box çizdirilebilir. İşte bu sorundan kurtulmak için de, daha önceden sözünü ettiğim non-maximum suppression tekniği kullanılır. Bu teknik ile yapılan şey basitçe, en yüksek güven skoru olan bounding box'ın kalması diğerlerinin ise görüntüden atılmasıdır.

Tüm işlemlerden sonra aşağıdaki çıktıya erişilir(Şekil 4.2):



Şekil 4.2 Nesnelerin Tanımlanması

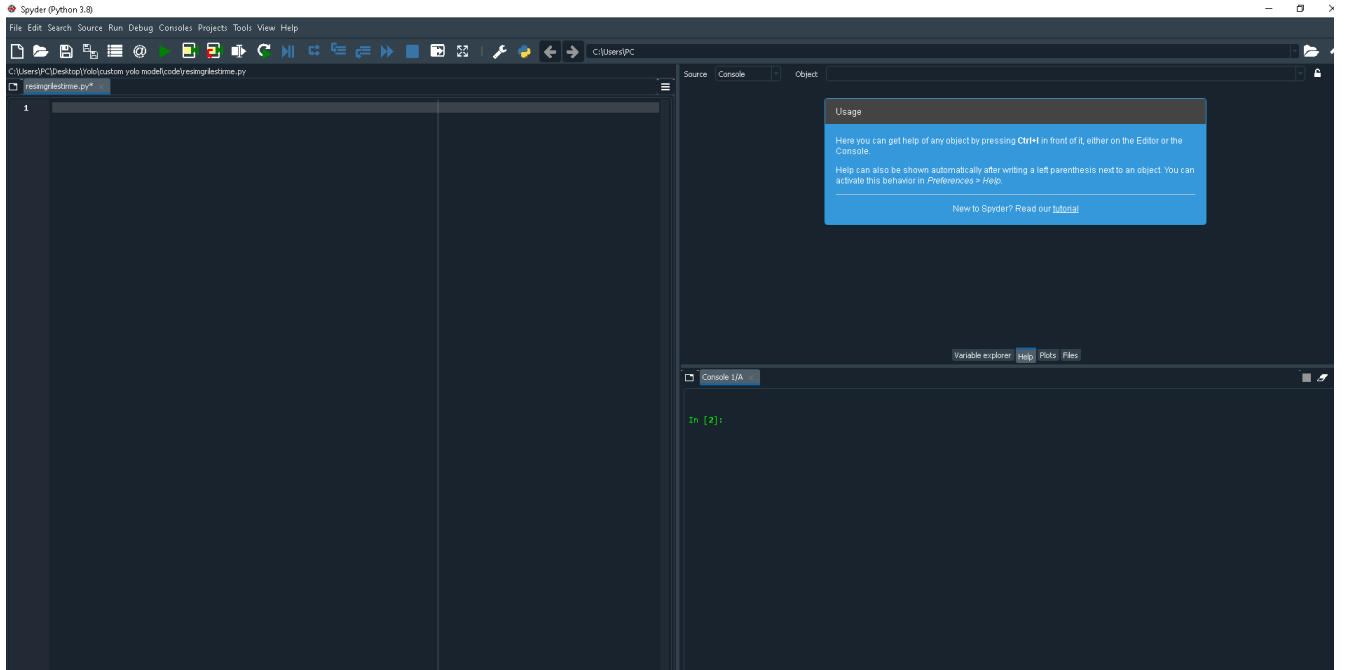
BÖLÜM 5

Python

Python 90'lı yılların başında Amsterdam'da Guido Van rossum tarafından geliştirilmeye başlanan bir programlama dilidir. nesne yönelimli, yorumsal, modüler ve etkileşimli, yüksek seviyeli bir dildir. Programlama dilleri makine mantığı ile insan mantığı arasında köprüdür.

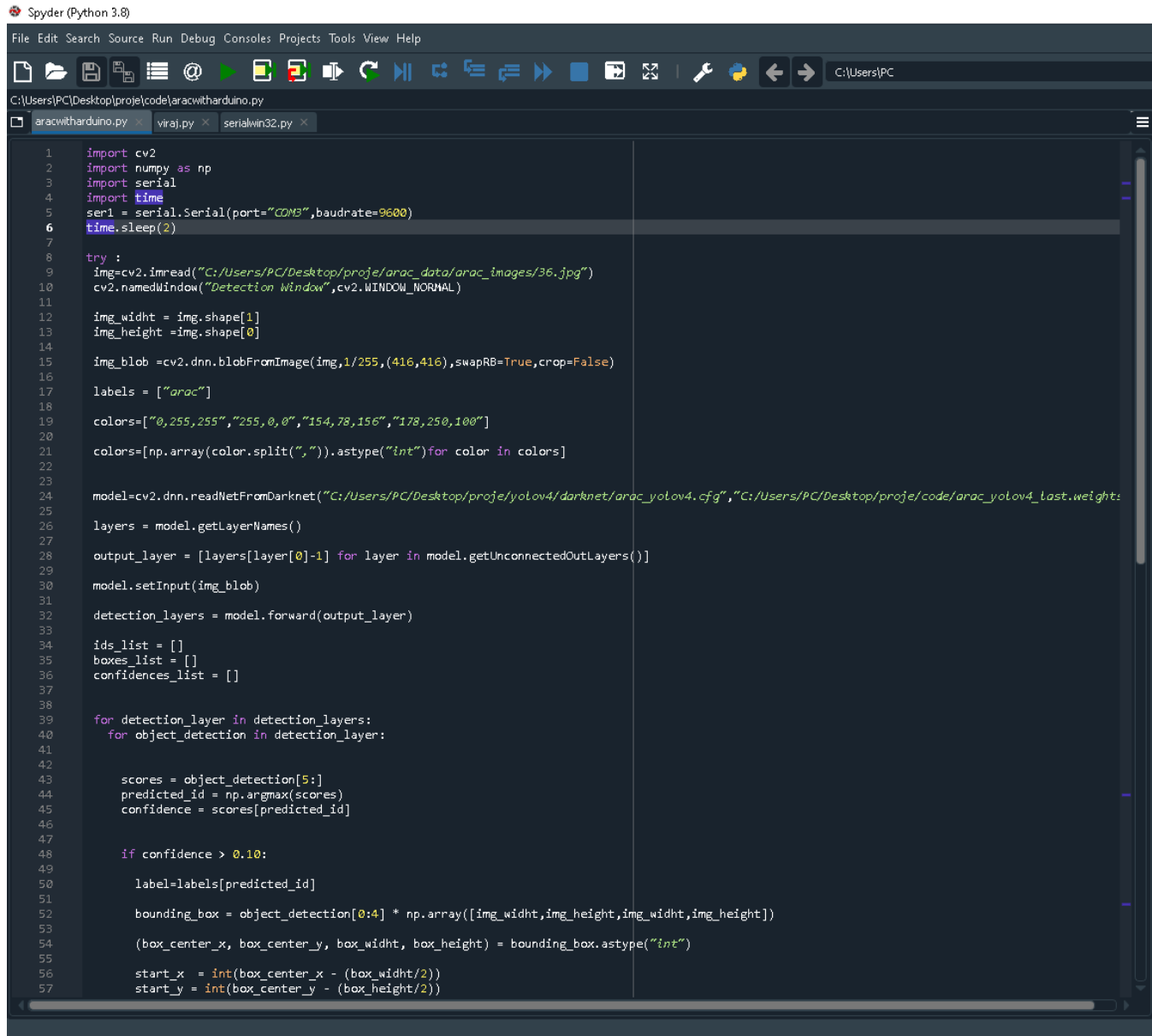
5.1. Spyder Arayüzü

Python ile yazılmış ve Python geliştirme için kullanabileceğiniz, açık kaynak bir IDE'dir. ... Ek olarak, **Spyder**, IPython ve NumPy, SciPy veya matplotlib gibi popüler Python kitaplıklarının desteği sayesinde sayısal bir hesaplama ortamıdır.



Şekil 5.1 Spyder Arayüzü

5.2 Plaka Tanımlı Garaj Kapısı Sistemin Spyder4 Arayüzü ile Pythom Dilinde Kodlanması



```

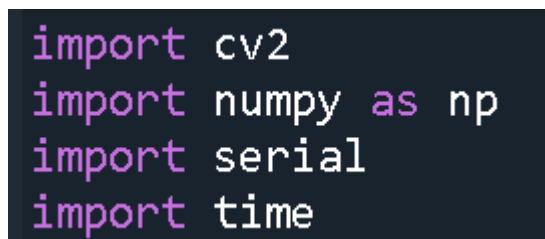
1  import cv2
2  import numpy as np
3  import serial
4  import time
5  ser1 = serial.Serial(port="COM3",baudrate=9600)
6  time.sleep(2)
7
8  try :
9      img=cv2.imread("C:/Users/PC/Desktop/proje/code/arac_data/arac_images/36.jpg")
10     cv2.namedWindow("Detection Window",cv2.WINDOW_NORMAL)
11
12     img_widht = img.shape[1]
13     img_height =img.shape[0]
14
15     img_blob =cv2.dnn.blobFromImage(img,1/255,(416,416),swapRB=True,crop=False)
16
17     labels = ["arac"]
18
19     colors=["0,255,255","255,0,0","154,78,156","178,250,100"]
20
21     colors=[np.array(color.split(",")).astype("int")for color in colors]
22
23
24     model=cv2.dnn.readNetFromDarknet("C:/Users/PC/Desktop/proje/yolov4/darknet/arac_yolov4.cfg","C:/Users/PC/Desktop/proje/code/arac_yolov4_last.weights")
25
26     layers = model.getLayerNames()
27
28     output_layer = [layers[layer[0]-1] for layer in model.getUnconnectedOutLayers()]
29
30     model.setInput(img_blob)
31
32     detection_layers = model.forward(output_layer)
33
34     ids_list = []
35     boxes_list = []
36     confidences_list = []
37
38
39     for detection_layer in detection_layers:
40         for object_detection in detection_layer:
41
42
43             scores = object_detection[5:]
44             predicted_id = np.argmax(scores)
45             confidence = scores[predicted_id]
46
47
48             if confidence > 0.10:
49
50                 label=labels[predicted_id]
51
52                 bounding_box = object_detection[0:4] * np.array([img_widht,img_height,img_widht,img_height])
53
54                 (box_center_x, box_center_y, box_widht, box_height) = bounding_box.astype("int")
55
56                 start_x = int(box_center_x - (box_widht/2))
57                 start_y = int(box_center_y - (box_height/2))

```

Şekil 5.2 Python komutlarının Girdisi

5.2.1 Kodların İçeriklerinin Aktarılması

Kısım- 1



```

import cv2
import numpy as np
import serial
import time

```

Şekil 5.3 Kütüphane Girdisi

Kütüphanelerin import edilmesi sisteme yüklenmesi.(şekil 5.3)

Kısım-2

```
ser1 = serial.Serial(port="COM3",baudrate=9600)
time.sleep(2)
```

Şekil 5.4 Port Haberleşme

Arduino ile Seri port üzerinden sistemin haberleştirimesi(şekil 5.4)

Kısım-3

```
cap = cv2.VideoCapture(0)
model=cv2.dnn.readNetFromDarknet("C:/Users/PC/Desktop/Yolo/custom_yolo_model/yolov
layers = model.getLayerNames()
output_layer = [layers[layer[0]-1] for layer in model.getUnconnectedOutLayers()]
```

Şekil 5.5 Layers

Video üzerinden Görüntü alınması Yolo kütüphanesi oluşturulmuş dosya üzerinden alınan görüntünün layerlerinin incelenmesi(Şekil 5.5)

Kısım-4

```
try :
    img=cv2.imread("C:/Users/PC/Desktop/proje/arac_data/arac_images/36.jpg")
    cv2.namedWindow("Detection Window",cv2.WINDOW_NORMAL)

    img_widht = img.shape[1]
    img_height =img.shape[0]

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

    labels = ["arac"]

    colors=["0,255,255","255,0,0","154,78,156","178,250,100"]

    colors=[np.array(color.split(",")).astype("int")for color in colors]
```

Şekil 5.6Python Kodlar

İncelenen layerlar doğru iken çalıştırılan programda çerçevelerin analiz edilmesi çerçeve isimlerinin renk ve isimlerinin sisteme tanıtılması(şekil 5.6)

Kısım-5

```

for detection_layer in detection_layers:
    for object_detection in detection_layer:

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

        if confidence > 0.10:

            label=labels[predicted_id]

            bounding_box = object_detection[0:4] * np.array([img_widht,img_height,img_widht,img_height])

            (box_center_x, box_center_y, box_widht, box_height) = bounding_box.astype("int")

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

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

max_ids = cv2.dnn.NMSBoxes(boxes_list, confidences_list, 0.5, 0.4)

```

Şekil 5.7 Bounding Box

Sisteme çıktı alınacak id lerin kutuların oranlarının listelerinin tanıtılması, For dönüğü altında Yolo üzerinden işlediğimiz görüntünün çerçevelerini bounding boxlarla (sınırlayıcı çerçevelerle) koordinatlarının bulunarak çizdirilmesi, Çizilen boxa id listesi kutu listesi ve ve güvenilirlik değerinin (oranının) atanmasının yazdırılması, oluşan güvenilirlik değerinin eşitliği >0.10 dan büyük olduğu durumlarda boxların çizdirilmesinin bulunduğu kodlardır. (Şekil 5.7)

Kısım-6

```

if(confidence > 0.20):
    ser1.write("1".encode())
    time.sleep(1)

label = "{}: {:.2f}%".format(label, confidence*100)
print("predicted object {}".format(label))
cv2.rectangle(img,(start_x,start_y),(end_x,end_y),box_color,2)
cv2.putText(img,label,(start_x,start_y-10), cv2.FONT_HERSHEY_SIMPLEX, 0.5, box_color, 2)

cv2.imshow("Detection Window",img)

except:
    print('yabancı araç')
    ser1.write("2".encode())
    time.sleep(1)

ser1.close()

cv2.waitKey(0)
cv2.destroyAllWindows()

```

Şekil 5.8 Değerlerin Çıktısı

Çizdirilen bound boxların güvenilirlik değerlerinin 0.2 den büyük olması durumunda plakanın belirtilecek şekilde arduinoya komut göndererek arduino ile haberleştirilmesi sağlanmıştır, eğer plakamız tanımlı ise arduinoya 1 çıktısı göndererek arduino 1 parametresi altındaki verileri işleyerek uygulamalı olarak çıktı almamızı sağlamaktadır. Eğer sisteme gösterilen plaka algılanamaz ise sistemden yabancı araç çıktısı yazdırılmaktadır. Komut dosyalarının tamamlanması çıktıların bize gösterilmesi ve pencerelerin kapatılması komutlarıylada python dilindeki kodlamalarımız son bulmaktadır.(Şekil 5.8)

BÖLÜM 6

ARDUİNO

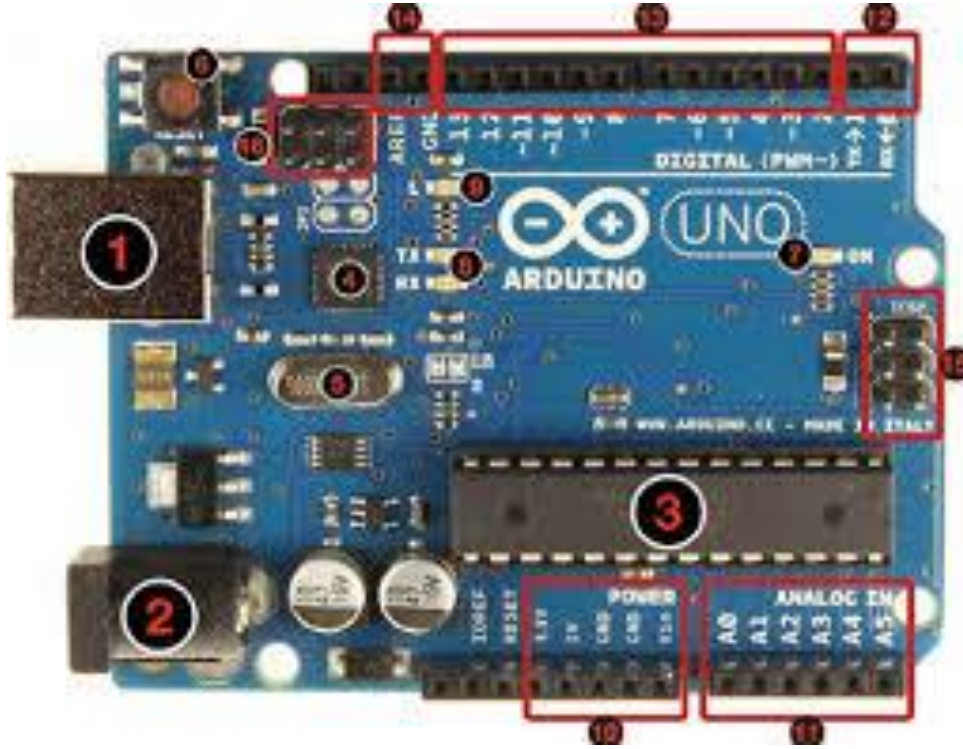
Arduino açık kaynaklı bir fiziksel programlama platformudur. Tek başına bir bilgisayar gibi düşünülebilir. Arduino sadece mikrodenetleyici kartlardan meydana gelmeyip geniş kütüphanesi, örnek projeleri ve öğrenimi kolay diliyle en çok tercih edilen proje geliştirme platformudur. Arduino' nun programlama dili olarak en yaygın C/C++ dili kullanılmaktadır. Arduino kartlar çok kolay temin edilebilmekte ve programlama da daha eğlenceli bir hal almaktadır. Arduino tanımında da bahsedildiği gibi açık kaynaklı olduğundan elde edilmesi de oldukça kolaydır. Arduino orijinal sayfasından ingilizce indirilebildiği gibi farklı kaynaklardan farklı dillerde indirmek de mümkündür. Arduino elektronik mikrodenetleyici bir karttan ve bir bilgisayaruygulamasından oluşmaktadır. Programda yazılan kodlar karta yüklendikten sonra istenilen komut yerine getirilmektedir.

6.1.Arduino Çeşitleri

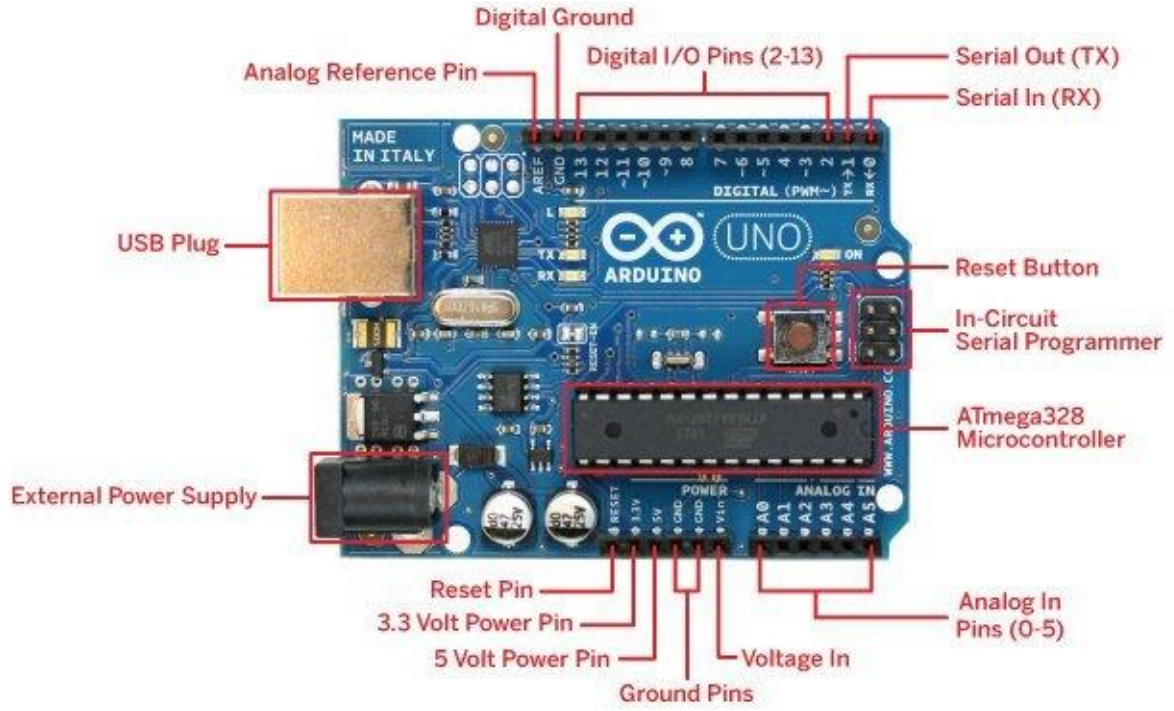
Arduino kartları kullanım alanlarına göre değişiklik göstermektedir. Arduino UNO; Arduino MEGA; Arduino LILYPAD; Arduino ADK, Arduino Leonardo, Arduino Mini, ArduinoNano, Arduino Yun bunlardan bazılarıdır. Yapılan çalışmada Atmega 328 mikroişlemciye sahip olan ArduinoUno kullanılmıştır.

6.2.Arduino Uno

Genel olarak Arduino modelleri arasında piyasada en çok kullanılan, bizim de çalışmamızda kullandığımız ArduinoUno modelini tanıyalım. ATmega 328 tabanlı bir mikroişlemciye sahip olan geliştirme kartının üzerinde, 14 adet dijital giriş/çıkış ve 6 analog giriş bulunmaktadır.16 Mhz kristal osilatöre, USB bağlantısına, güç bağlantısına, ICSP bağlantısına ve reset tuşuna sahiptir. Kartın çalışabilmesi için harici bir güç kaynağı yada bilgisayarın Usb portu ile enerji verilmesi gerekmektedir.



Şekil 6.1 Arduino Uno



Şekil 6.2 Arduino Uno Pin Değerleri

1. USB jakı
2. Power jakı (7-12 V DC)
3. Mikrodenetleyici ATmega328
4. Haberleşme çipi
5. 16 MHz kristal
6. Reset butonu
7. Power ledi
8. TX / NX ledleri
9. Led
10. Power pinleri
11. Analog girişler
12. TX / RX pinleri
13. Dijital giriş / çıkış pinleri (yanında ~ işareti olan pinler PWM çıkışı olarak kullanılabilir.)
14. Ground ve AREF pinleri
15. ATmega328 için ICSP USB arayüzü için ICSP

Mikrodenetleyici	ATmega328P
Çalışma Gerilimi	5V
Önerilen Adaptör Giriş Gerilimi	7-12V
Adaptör Giriş Gerilim Sınırı	6-20V
Dijital Pin Sayısı	14 (Bunlardan 6 sı PWM özellikli)
PWM Pin Sayısı	6
Analog Pin Sayısı	6
Giriş Çıkış Pinleri İçin Akım	20 mA
3.3V Pin İçin Akım	50 mA
Flash Hafıza	32 KB (ATmega328P) 0.5 KB bootloader tarafından kullanılır
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
Saat Hızı	16 MHz
Uzunluk	68.6 mm
Genişlik	53.4 mm
Ağırlık	25 g

Şekil 6.3 Arduino Uno Sistem Bilgisi

6.2.1. Arduino Programındaki Başlıca Kodlar

Programda yer alan başlıca kullanılan kodlar aşağıda verilmiştir.

- voidsetup(): Bu kodun ardından yazılacak komutlar karta yüklendikten sonra sadece bir kereliğine çalışmaktadır.
- voidloop(): Bu fonksiyonun arasına yazılacak komutlar program akışı boyunca sürekli tekrar etmektedir. Fonksiyonlar en üst komuttan en alt komuta doğru akar ve bittiğinde tekrar başa döner.
- analogRead(): Analog pin değerlerini okur.
- delay(): Bir kodun çalışma süresinin belirlendiği bölümdür.

6.3. Plaka Tanımlı Otomatik Garaj Sistemi Arduino Program Kodları

```
#include <SPI.h>
#include <Wire.h>
#include <Servo.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#include <OLED_I2C.h>
#define echoPin 4
#define trigPin 3
```

Şekil 6.4 Kütüphane

- Kütüphanelerin programa tanıtıldığı kısım(Şekil 6.4)


```

void setup()
{
    motor.attach(2);
    display.begin(SSD1306_SWITCHCAPVCC, 0x3C);
    Wire.begin();

    // Tampon Bellek sıfırlama
    display.clearDisplay();

    pinMode(8, OUTPUT);
    pinMode(9, OUTPUT);
    pinMode(10, OUTPUT);
    pinMode(11, OUTPUT);
    pinMode(12, OUTPUT);
    pinMode(trigPin, OUTPUT);
    pinMode(echoPin, INPUT);
    Serial.begin(9600);
}

```

Şekil 6.6 Pin Ataması

- Oled ekranının başlatılması ve temizlenmesi, arduino üzerindeki Pinlerin giriş ve çıkışının atanması. (Şekil 6.6)

```

void loop()
{

    digitalWrite(8, HIGH);
    digitalWrite(9, HIGH);
    digitalWrite(10, HIGH);
    digitalWrite(11, LOW);
}

```

Şekil 6.7 Pin Değerleri

- 8,9,10. Pinlere bağlı rgb ledin başlangıçta yanmaması için pinlere 5 v değer girilmesi .(Şekil 6.7)

```

if (Serial.available() > 0)
{
    arac = Serial.read();
    Serial.print(arac);

    if (arac == '1') {

        digitalWrite(8, LOW);
        digitalWrite(9, HIGH);
        digitalWrite(10, HIGH);
        digitalWrite(11, HIGH);
        delay(200);
        digitalWrite(11, LOW);
        delay(100);
        digitalWrite(11, HIGH);
        delay(250);
        digitalWrite(11, LOW);

        display.setTextSize(1, 3);
        display.setTextColor(WHITE);
        display.setCursor(27, 13);
        display.println("TANIMLI ARAC");
        display.setCursor(35, 42);
        display.println("16 UN 190");
        display.display();

        for (Derece = 90; Derece > 0; Derece--)
        {
            motor.write(Derece);
            delay(35);
        }
        delay(3000);
        geri:
        digitalWrite(trigPin, HIGH);
        delayMicroseconds(10);
        digitalWrite(trigPin, LOW);
        sure = pulseIn(echoPin, HIGH);
        mesafe = (sure / 2) / 29.1;

        if(mesafe>10){
        for (Derece = 0; Derece < 90; Derece++)
        {
            motor.write(Derece);
            delay(35);
        }

        digitalWrite(8, HIGH);
        digitalWrite(9, LOW);
        digitalWrite(10, LOW);
        digitalWrite(11, HIGH);
        delay(500);
        digitalWrite(8, HIGH);
        digitalWrite(9, HIGH);
        digitalWrite(10, HIGH);
        digitalWrite(11, LOW);
        delay(300);
        digitalWrite(8, HIGH);
        digitalWrite(9, LOW);
        digitalWrite(10, LOW);
        digitalWrite(11, HIGH);
        delay(500);
        digitalWrite(8, HIGH);
        digitalWrite(9, HIGH);
        digitalWrite(10, HIGH);
        digitalWrite(11, LOW);
        display.setTextSize(1, 4);
        display.setTextColor(WHITE);
        display.setCursor(25, 17);
        display.println("YABANCI ARAC");

        display.display();
        delay(5000);
        display.clearDisplay();
    }

    else (arac == '0');
    {
        digitalWrite(8, HIGH);
        digitalWrite(9, HIGH);
        digitalWrite(10, HIGH);

    }
}
display.drawBitmap(15, -5, araba, 100, 50, WHITE);
display.setTextSize(1);
display.setTextColor(WHITE);
display.setCursor(45, 47);
display.println("OTOPARK");
display.setCursor(45, 55);
display.println("Sistemi");
display.display();
delay(1000);
display.clearDisplay();

```

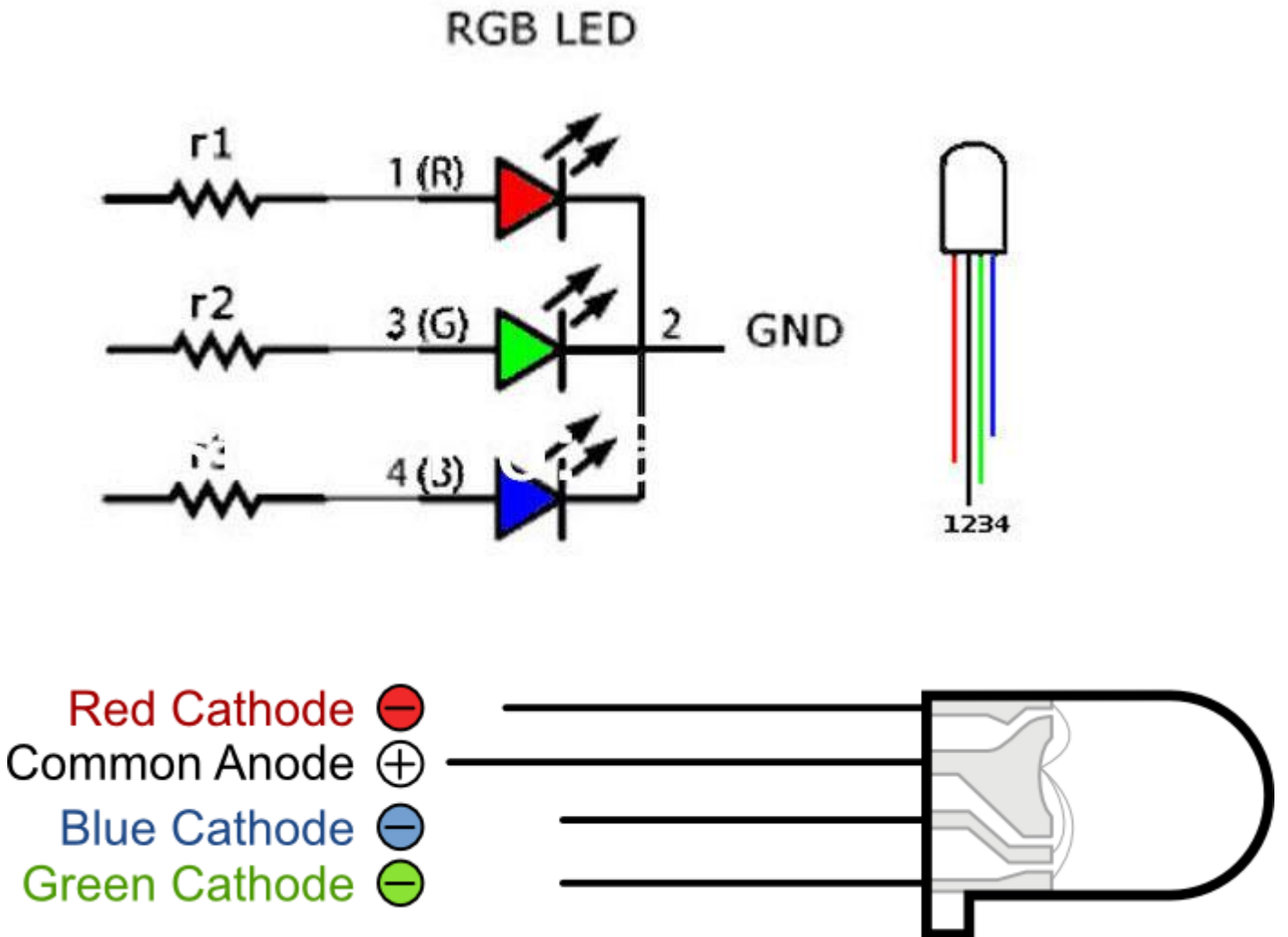
Şekil 6.8 Çıktılar

- Bu komutlarda da Python ile arduino'nun serial haberleşmesi ve pythondaki çıktı değerinin arduinodan okunması.Eğer çıktı değeri 1 ise lambaya bağlı pinlerin (8-9-10)değerlerinin belirlenmesi buzzerin ötmesi (11) ve oled ekranda 1 değeri altında istenilen plaka isminin yazarak göstermesi garaj kapısına bağlı servo motoru istenilen açıda döndürmesi ve mesafe sensörünün her hangi bi engel görmediği zaman tekrar motorun konumunu eski hale getirmesi kod edilmiştir.Diğer komutlarda bu düzeyde alınan girdiye göre değiştirilerek programa aktarılmıştır.(Şekil 6.8)

BÖLÜM 7

RGB LED

RGB – **R**ed (kırmızı) , **G**reen (yeşil) , **B**lue (mavi) renklerin baş harfleri birleştirilerek oluşmuş bir terimdir. Genel çalışma prensipi; bu üç rengi kullanarak, farklı kombinasyonlarda, çok fazla renk verebilir. RGB LED 'ler, bir kontrol devresi yardımıyla 16 renk verebilmektedirler.



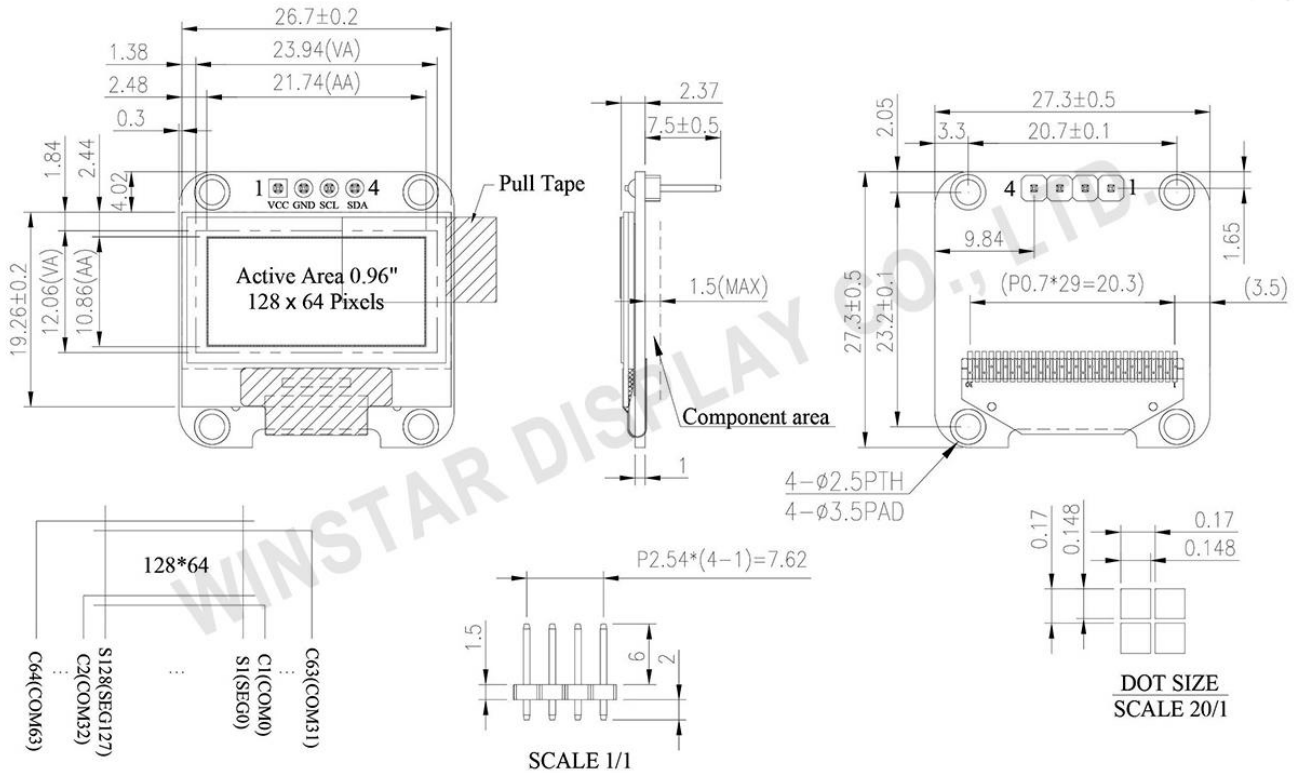
Şekil 7.1 RGB Led Yapısı

BÖLÜM 8

OLED EKRAN

LED ve LCD gibi bir ekran teknolojisi olan OLED ya da adının Türkçe açılımıyla **Organik Işık Yayan Diyotlar**, LED teknolojisinin farklı bir versiyonudur. OLED'i genel olarak diğer ekran teknolojilerinden ayıran özelliği yarıiletkenlerden geçen elektrik sayesinde ışığın oluşması ve ışığın ekranın alt katmanında yer alan yayıcı tabadaki deliklerine ışığın yönltilmesiyle görüntü vermesidir. Bu teknoloji ilk olarak **Kodak** tarafından üretilmiştir.

İlk etapta bahsettiğimiz bu tanım kulağa karmaşık bir şeymiş gibi gelse de aslında OLED teknolojisi bu sayede **ışığın kaynağını** ve **renk dizilişini** aynı anda kontrol ederek her pikselde **uygun parlaklık** ve **tonlarda** ışık aktarılmasını sağlar. Böylelikle OLED ekrana sahip cihazlarda çok daha **parlak** ve **gerçek tonlara yakın** renklerle güzel bir görüntü oluşur. Aynı zamanda her pikselin parlaklığı farklı bir şekilde ayarlanabildiği için OLED ekranlar diğer ekran teknolojilerine göre **daha az enerji** tüketirler.



Şekil 8.1 128*64 Oled Datasheet

BÖLÜM 9

BUZZER

Buzzer; mekanik, elektromekanik ya da piezoelektrik prensiplerine bağlı olarak çalışan işitsel ikaz cihazı çeşididir. Kullanım alanları oldukça fazla olan buzzerlar, genel itibarıyla piezoelektrik prensibiyle çalışmaktadırlar. Buzzerlar, kullanım alanlarına da bağlı olarak alarm, zamanlayıcı, onaylama cevap ikazı gibi işlevlerde kullanılabilirler. Nitekim tanımda da belirttiğimiz üzere, buzzerlar işitsel ikaz cihazı çeşitleridir. Işıklı buzzer, ışısız buzzer, pasif buzzer ve aktif buzzer gibi türlere sahiptirler.

9.1 Çalışma Şekli

İlk önce buzzer direnç ve transistör kullanarak input pinine gelen dc voltu salınım sinyaline çevirir. İndüktör bobini kullanarak sinyal büyütülür. Piezo seramik diske yüksek gerilim uygulandığında radyal yönde mekanik olarak genişleme ve daralmaya sebep olur. Bu da içerideki metal plakanın ters yönde bükülmesine sebep olur. Metal plakanın sürekli olarak zıt yönde bükülmesi ve büzülmesi sonucu buzzer havada ses dalgaları üretir.

Buzzer enerjiyi bir yolla alır ve onu akustik enerjisine çevirir. Bazı buzzer lar kendi devrelerine sahiptir ve onlar gücü direkt olarak cihazın güç kaynağından alır. Diğer yandan bazı buzzerler ise pilli olabilir olası şebeke kesintisinde çalışmaya devam etmesi için. Bazı buzzerlar ise tehlikeli seviyede gerilime sahip güç kaynaklarının üzerinde bulunur ve şebeke yerine teklikeli gerilim hattından beslenerek çalışır.



Şekil 9.1 Buzzer

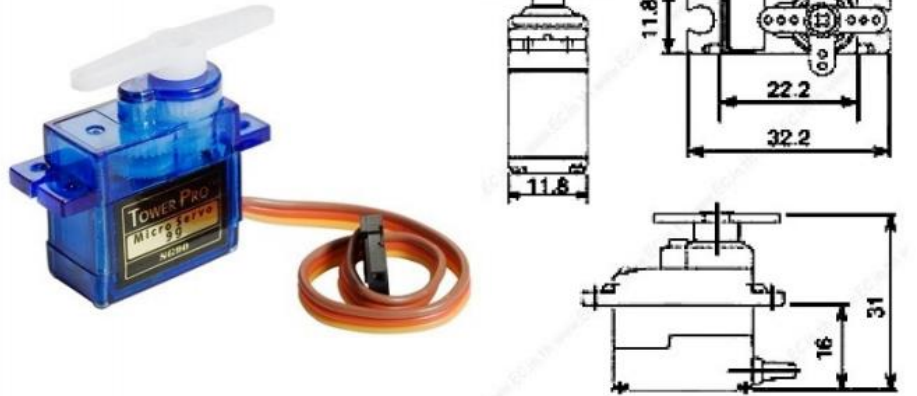
BÖLÜM 10

SERVO MOTOR

Servo motorlar içerisinde motorun hareketini sağlayan bir DC motor bulundurmaktadır. Bu motorun dışında bir dişli mekanizması, potansiyometre ve bir motor sürücü devreside bulundurulur. Potansiyometre, motor milinin dönüş miktarını ölçmektedir. Servo içerisindeki DC motor hareket ettikçe potansiyometre döner ve kontrol devresi motorun bulunduğu pozisyon ile istenilen pozisyonu karşılaştırarak motor sürme işlemi yapar. Yani, servolar diğer motorlar gibi harici bir motor sürücüyü ihtiyaç duymadan çalışmaktadırlar. Genellikle çalışma açıları 180 derece ile sınırlıdır fakat 360 derece çalışma açısına sahip özel amaçlı servo motorlar da vardır. Servolar genellikle 4.8-6V gerilim ile çalışmaktadırlar.

Teknik Özellikler

- Ağırlık: 9 g
- Boyut: 22.2 x 11.8 x 31 mm ~
- Durma Torku: 1.8 kgf·cm
- Çalışma Hızı: 0.1 s/60 derece
- Çalışma Gerilimi: 4.8 V (~5V)
- Ölü Bant Genişliği: 10 µs
- Sıcaklık Aralığı: 0 °C – 55 °C



Şekil 10.1 Servo Motor

SONUÇ

Gerçekleştirilen bu çalışmada görüntü işleme teknikleri kullanılarak bir plaka tanıma sistemi geliştirilmiştir. Daha önceki çalışmalardan farklı olarak yazılımda, hazır sistemler yerine Görüntü işleme kullanılarak bütün algoritmalar yeniden yazılmıştır. Bu yazılımda kişisel plakanın okunması sağlanmıştır. Sistemde, giriş/bekleme/çıkış sürelerini azaltmak için geliştirilen sistemin denemeleri yapılmış ve başarılı sonuçlar elde edilmiştir. Veri tabanında tutulan plakalar ile okunan plakalar karşılaştırılmış ve sistemin hatasız çalıştığı tespit edilmiştir.

KAYNAKLAR

<https://www.udemy.com/course/bilgisayarli-goru-yolov4-ile-nesne-tanma/>

<https://www.ceyrekmuhendis.com/goruntu-isleme-nedir-ve-nerelerde-kullanilir/>

<https://mesutpiskin.com/blog/opencv-nedir.html>

<https://www.mediatick.com.tr/tr/blog/python-nedir>

<http://gorselanaliz.com/yolo-nedir/#page-content>

<https://www.winstar.com.tw/tr/products/oled-module/graphic-oled-display/4-pin-oled.html>

<http://ibrahimozcelik.net/index.php/2017/07/30/opencv-arduino-haberlesmesi/>

<https://medium.com/@ilxmuhendislik1/buzzer-nedir-nas%C4%B1l-%C3%A7al%C4%B1%C5%9F%C4%B1r-7f902167f595>

<https://maker.robotistan.com/rc-servo-motor-nedir/>

<https://content.instructables.com/ORIG/FA2/O1SS/J7ARLNBW/FA2O1SSJ7ARLNBW.pdf>