T.C.
KÜTAHYA DUMLUPINAR ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ BÖLÜMÜ

VİRAJ LEVHASI İKAZ SİSTEMİ
LİSANS BİTİRME TEZİ

HAZIRLAYAN

AHMETCAN TOPÇUOĞLU

201513151015

TEZ DANIŞMANI

PROF. DR. HAMDİ MELİH SARAOĞLU

KÜTAHYA, 2021

ÖZET

Trafik işaretlerinin sürüş esnasında tespiti, sürücünün güvenliğini sağlaması açısından ve araçlarda oluşabilecek maddi hasarları önlemesi açısından önem arz etmektedir. Bu çalışma kapsamında ise Viraj işaretini tanıyabilecek ve algılayabilecek bir sistem önce benzetim ortamında sunulmuş daha sonra uygulama olarak hem benzetim ortamında hem gerçek zamanlı olarak viraj işareti levhasının tespiti gerçekleştirilmiştir. Uygulamada viraj işaretinin belirli bir mesafede tespit edilmesi neticesinde kullanıcıya sesli uyarı verdirilerek belirli bir süre önce sürücünün hızını azaltması beklenmektedir. Büyük otomobil üretici firmalarının bu sistemlerile donattıkları araçlar yüksek fiyatlarda alıcı bulabilmektedir. Bu tez çalışması ile hedeflenen,bu sistemlerin çok ucuz maliyetler ile ülkemizsınırlarının içerisinde üretilebilmesidir.

Viraj tabelası ikaz sistemi alıcıya sesli,ışıklı,ve yazılı şekilde uyarı veren sistemdir.Genellikle kazaların çoğu araştırmalar sonucu uzun düz bir yoldan sonra aniden çıkan bir virajda rastlanmaktadır,Uzun düz yollarda şoförler rahatlıkla yollarında seyahat ederken yolun hep düz olmasından kaynaklanan bir rahatlık sergilerler ve gözleri dalmaya başlar,uykunun azizliğine uğrayan şoförler uzun yolun ardından çıkan aniden virajda kazaya sebebiyet verebilmektedirler.Bende projemde yoldaki viraj yönlerini algılayan bir algoritma geliştirip viraj tabelasına göre virajın yönünü ve keskinliğini belirleyen sistem yaptım,Böylelikle kaza oranlarını en aza indirmeye çalışacağım,dalgınlağa uğrayan şoför sistemin tabelayı farketmesiyle birlikte şoförün isteğine göre hem sesli hem yazılı hemde ışıklı uyarıda bulunacaktır.

ABSTRACT

Detection of traffic signs during driving is important in terms of ensuring the safety of the driver and preventing material damage that may occur in vehicles. In the scope of this study, a system that can recognize and detect all bend signals was presented in the simulation environment first and then the application of bend marking plate was performed both in real time and in simulation environment. In practice, it is expected to reduce the speed of the driver for a certain period of time by giving an audible warning to the user in the event that the beep sign is detected at a certain distance. The cars that big carmakers make with these systems find buyers at high prices. The aim of this thesis is to produce these systems within the boundaries of our country with very cheap costs

Cornering sign warning system is a system that warns the driver with sound, light and written form. As a result of most studies, accidents are usually caused by a sudden bend after a long straight road. While traveling on long straight roads, the drivers are comfortable because the road is always straight and their eyes start to close. Drowsy drivers can cause an accident at a sudden corner after a long and straight road. In my project, I developed an algorithm that can detect bends on the road and built a system that determines the direction and sharpness of the bend according to the cornering sign.In this way, I will try to minimize the accident rates. When the system notices the sign, it will be able to make an audible, written and illuminated warning according to the driver's request.

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

Sayfa

İÇİNDEKİLER(Devam)

ŞEKİLLER DİZİNİ

Sayfa

ŞEKİLLER DİZİNİ(Devam)

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.Bazı durumlarda teknoloji, yaşamsal olayları kolaylaştırmanın yanı sıra insanların hayatlarına dokunarak yaşamsal faaliyetlerini koruma amaçlı da yapılmaktadır. Otomotiv sektöründeki teknolojik gelişmelere paralel olarak araç sayısının artması, kullanılan yolların sayısının ve karışıklığının artmasına dolayısı ile sürücülere kolaylık sağlamak amaçlı birçok trafik işaretinin kullanımına neden olmuştur Ulaşımda can ve mal güvenliliğinin sağlanması için sürücülerin trafik kurallarına ve işaretlerine uymalıdır. Fakat taşıt sayısındaki artış, yollardaki sıkışıklık ve değişkenlik ve çevresel etkilerden dolayı sürücülerin kurallara uymamasından kaynaklanan kazalar meydana gelmektedir. Bu kazalardan bir kısmı da trafik işaretlerinin görülmemesi ile ilgilidir. Trafik işareti tanıma sistemlerinin amacı yol üzerindeki işaretleri sürücüye bildirmek ve bu şekilde sürücünün kontrollü ve güvenli bir sürüş yapmasını sağlamaktır. Bu sistemler araçlarda standart donanım olabileceği gibi harici olarak da araca eklenebilmektedir. Trafik işareti tanıma sistemleri sürücünün dalgınlık, dikkatsizlik, uykusuzluk gibi nedenlerden dolayı trafik işaretini fark edememesi durumunda devreye girerek sürücüyü uyarmaktadır. Bende bu projemde trafik kazalarını konu alarak,kaza sebeblerini inceleyerek kaza oranını bir tık daha azaltıp can kayıplarını önleme amacıyla bu projeye adım attım.Viraj tabelalarını algılayan sistemimde, dalgın şoförlerin virajları algılamasında onlara yardımcı olacak sesli ve ışıklı sistem tasarladım.

Dalgınlığa karşı kazaları önleme amacıyla geliştirdiğim bu projemle bir çok uzun yol şoförüne yardımcı olabilmeyi,Şoföre gözden kaçırdığı tabelaları ikaz ederek daha günenilir hale getirmeyi ve daha konforlu yaşam sunmayı hedeflemekteyim.Sistemi araçlara ilk başta harici takılan bir donanım olarak üretip böyle bi teknolojiye sahip olmak isteyen Şoförlerin kolayca alıp araçlarına takabilcekleri uygun düzenekte montajı rahat bir şekilde üretilecek,ve sistem anahtar sayesinde istenildiği zaman devreye sokulup devre dışı edilebilecek.

BÖLÜM 2

Viraj TABELASI ALGILAYAN SİSTEM

2.1. Trafik Levhaları Nedir ?

Günlük hayatınızda aracınızda seyahat ederken yolda bir çok uyarıcıyla karşılaşırsınız bunların her birinin ayrı ayrı anlamları mevcuttur ama ortak tek bir yönleri vardır oda alıcıya,Şoföre ikazda bulunmak,Yolun durumunu koşulunu yolda çıkabilecek unsurları,hızınızın olması gerektiği durumu, yapmanız gereken bir çok faliyeti yolda şoföre aktarmaktadır.Fakat bir çok şoför bunlara bazen dalgınlıkla bazen göz ardı ederek dikkat etmemektedir.Bazı acemi şoförlerde temel tabelalar hariç çoğunun anlamını bilmemektedir.Dolayısı ile böyle bir düzen içinde kazalarda kaçınılmaz olmaktadır.Trafik levhalarının anlamları iyi öğrenilmeli ve trafikte dikkatle takip edilmelidir. Trafik levhaları, sürücü ve yayalara, uymaları gereken kuralları simgeler ile anlatır. Yeterli sayıda kullanılması gerekmektedir. Kolay görünür şekilde yerleştirilmesi, tercihen omega direk kullanılması tavsiye edilmektedir.

Şekil 2.1(a) Trafik Levhaları

Şekil 2.1(b) Trafik Levhaları

2.2 Trafik İşaretlerinin Standartları

Her gün trafikte, sokaklarda ve caddelerde gördüğümüz trafik işaret levhalarının diğer dış etmenlerden ayırt edilebilmelerini sağlayan belirli geometrik şekil ve renk özellikleri vardır. Trafik işaret levhaları temsil eden 4 (dört) farklı temel özellik vardır. Bunlar; işaret çerçevesinin şekli ve rengi, çerçeve içi dolgu rengi, her bir trafik işaretine özgü semboller olan piktogramın şekli ve renk dolgusu ile işaret levhasının genel şeklidir (Şekil 2.2)Bu 4 temel özelliğin tamamı veya bir kısmı farklı kombinasyonlarla biraraya getirilerek farklı semantik kategoriden trafik işareti sınıfları oluşturulur.

Şekil 2.2 Trafik İşaret Levhaları 4 (dört) temel özelliği

2.3. Trafik levhaları Nerelerde Kullanılmaktadır ?

>>>Şehir içi ve şehirler arası otoyollarda
>>> Açık veya kapalı tüm otopark alanlarında
>>> Yol çalışmalarında, araç ve yayaları uyarmak amacıyla
>>> Daralan yol ayrımlarında
>>> Geniş kullanım alanı: Yollarda uyarı ve çalışma alanlarında, sosyal alanlarda, can ve mal güvenliği gerektiren her türlü alanda kullanılmaktadır.

2.4. Viraj Levhaları Anlam ve Yapıları ?

Sağa tehlikeli viraj işareti: İleride sağa dönemeçli bir yol kesimine yaklaşıldığını bildirir.  a)

Sola tehlikeli viraj işareti: İleride sola dönemeçli bir yol kesimine yaklaşıldığını bildirir.  b)

Tehlikeli viraj yön işareti: Görüşü artırmak ve sürücülere daha iyi kılavuzluk sağlamak amacı ile güvenli seyir olasılığı bulunmayan keskin virajlarda, tehlikeli viraj işaret levhalarından sonra kullanılır.

 c)  d)  e)  f)

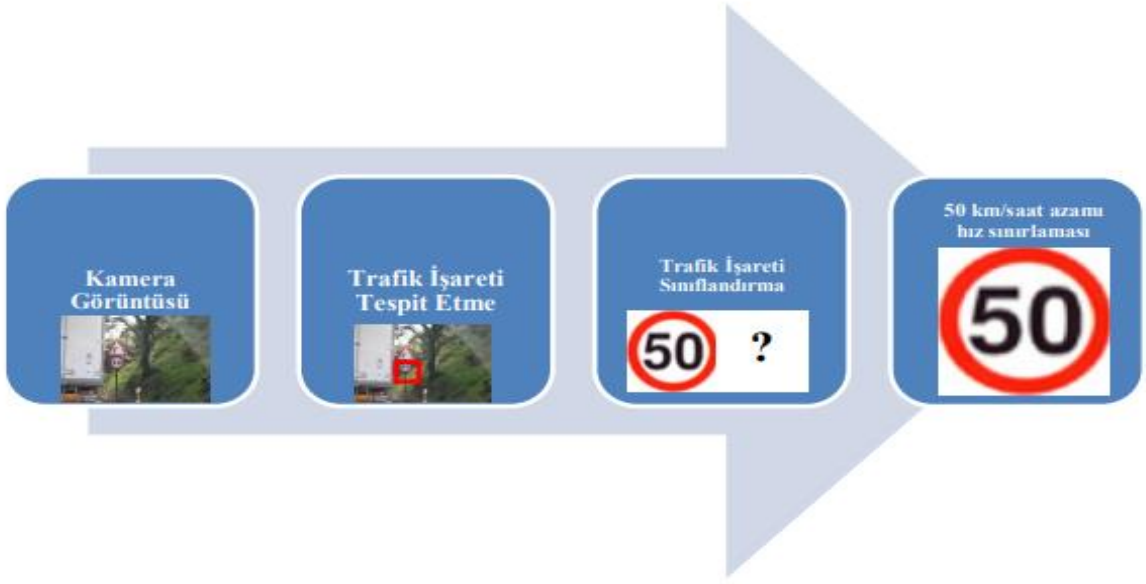
Sağa tehlikeli devamlı viraj işareti: İleride ilki sağa birbirini izleyen tehlikeli virajlar olduğunu bildirir.  g)

Sola tehlikeli devamlı viraj işareti: İleride ilki sola birbirini izleyen tehlikeli virajlar olduğunu bildirir.  h)

(Şekil 2.3 Viraj işaretlerine örnek a) Sağa tehlikeli viraj, b)Sola tehlikeli viraj,c)Sağa viraj, d)Sola viraj, e)Sola keskin viraj, f)Sağa keskin viraj, g)Sağa devamlı viraj, h)Sola devamlı viraj)

2.5. Trafik İşaret Tanıma Literatür Özeti

Trafik işaretlerini tanıma üzerine ilk çalışmalar Japonya'da yapılmıştır. Yayımlanmış birçok çalışmada trafik işaretini tanıma sistemi tespit ve sınıflandırma olarak iki aşamada incelenmiştir (Gürbüz ,2010; Ruta et al.,2010; Fleyeh and Davami, 2011; Mariut et al., 2011; Becer, 2011). Trafik İşaret Tanıma Sisteminin genel yapısı Şekil 2.3'de gösterilmektedir.



Şekil 2.3 Trafik İşareti Tanıma Sistemi Genel Yapısı

2.5.1.Viraj Levhası İkaz Sisteminin Çalışma şekli

Viraj levhası ikaz sistemi yolov3 ile geliştirilmiş viraj tabelalarını algılayan sistemdir.Görüntü işlemeye yönelik kullandığım yoloda viraj tabelalarını sisteme tanıtmak için elde ettiğim fotoğraflardaki viraj tabelalarını etiketleyip elde ettiğim verileri Google'ın ücretsiz gpu altyapısını kullanarak Colaboratory üzerinden analiz ettirerek birbirleriyle etkileştirip eğittim.Çıktı olarak topladığım verileri python dilini kullanarak spyder4 programı üzerinden yazdığım kodlarla bilgisayar ortamında test aşamalarını sürdürdüm,Çalışan programı arduino ile haberleştirip algıladığı tabelaya göre led yakan, sesli ikaz eden ve yazı yazan programı arduino üzerinden tasarladım.Kısaca özetleyecek olursak Projem,kameranın izlediği yolda algılanan trafik levhasını spyder üzerinden pythom dilinde yolov3 ile inceleyip algılanan tabelanın şekline göre arduino ile haberleşerek arduino üzerinden şoföre sesli,yazılı,ledli bir şekilde ikazda bulunmaktadır.

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;

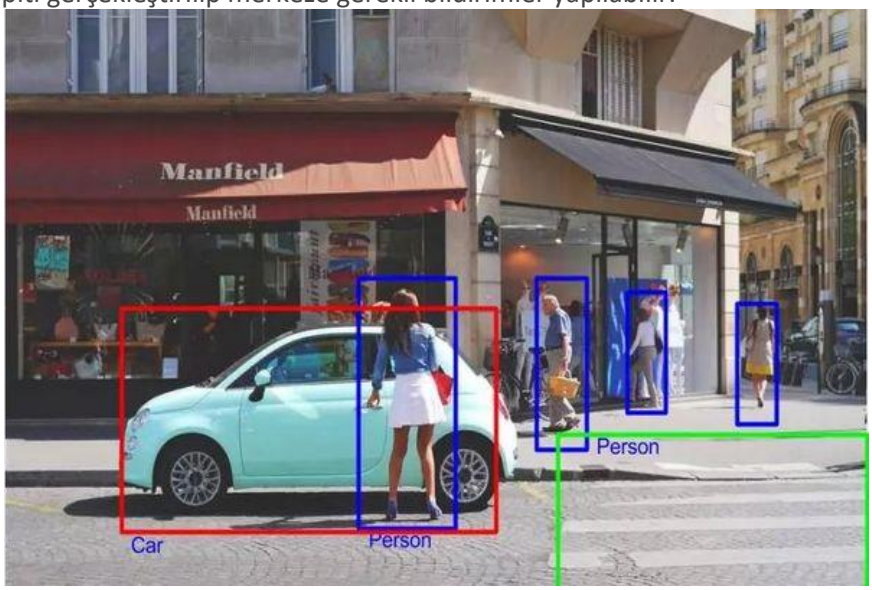
- Görüntü üzerinde bulunan gürültülerin arındırılması ve temiz bir görüntü elde edilmesi
- Görüntü üzerinde bulunan ve insan algısının görmekte zorlandığı nesnelerin tespiti
- Görüntünün daha kaliteli bir hale getirilmesi
- Nesne takibi yapılması
- Görüntü üzerindeki farklı nesnelerin birbirinden ayırt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

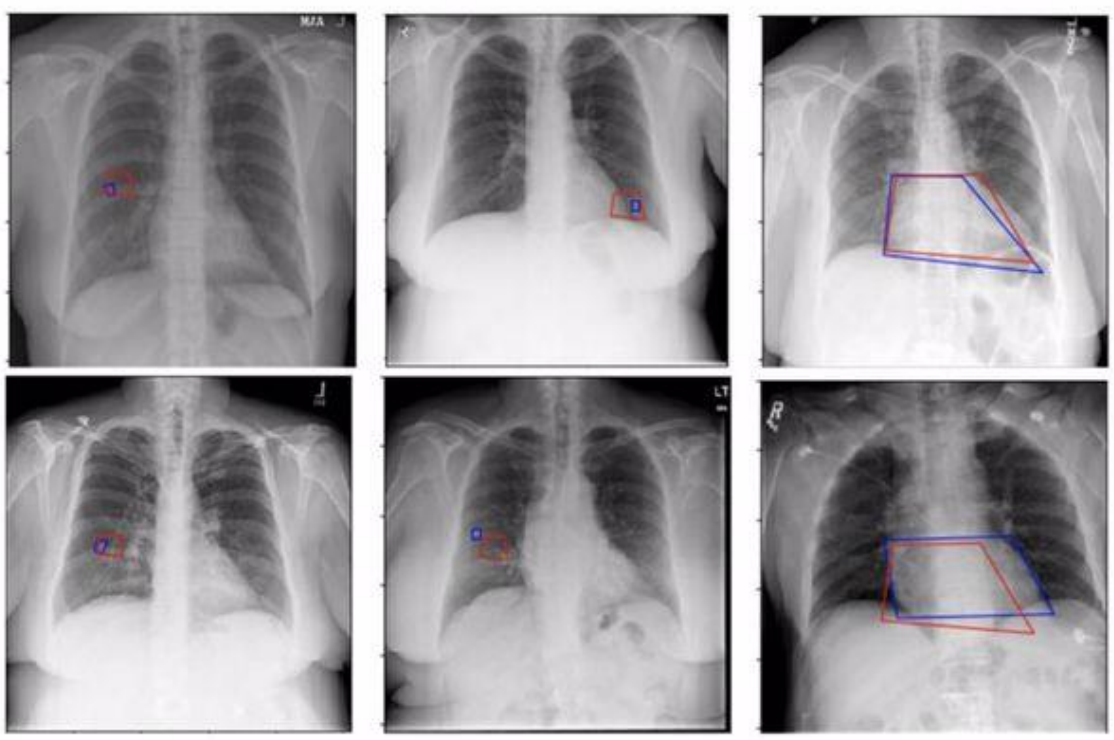
Şekil 3.1 Görüntü İyileştirme

- **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 tanıması, 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.

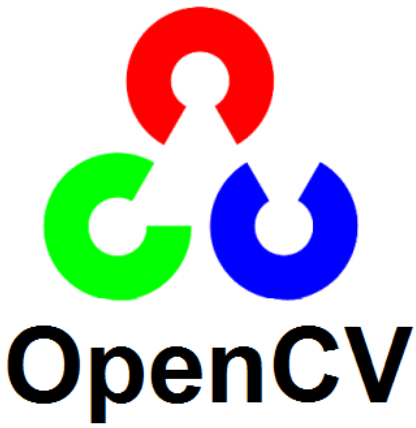
OpenCV: Günümüzde görüntü işleme çalışmalarında en aktif kullanılan kütüphanelerin başında gelmektedir. İçerisinde bir çok hazır fonksiyon bulundurması sebebiyle hızlı bir şekilde proje gerçekleştirilebilir. Aynı zamanda C++, Java, Python, C#, MATLAB gibi farklı programlama dilleri içerisinde kullanılabilmesi 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 İntel 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 ?

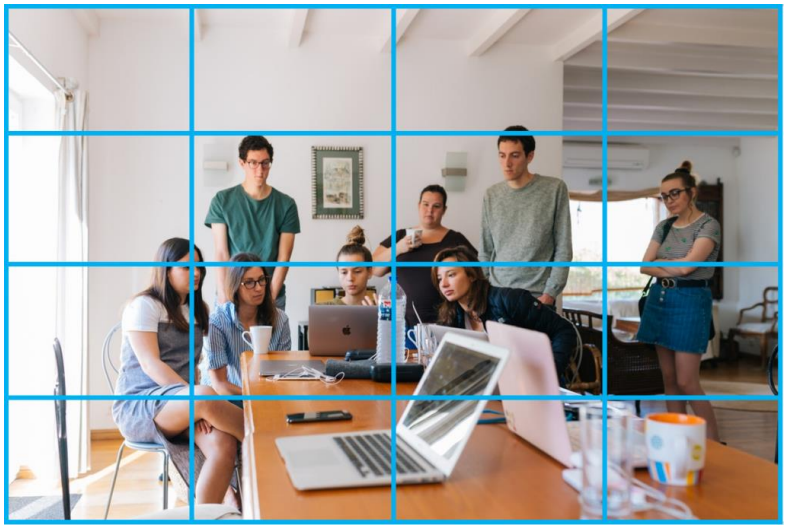
YOLO, konvolüsyonel sinir ağlarını (CNN) kullanarak nesne tespiti yapan bir algoritmadır. Açılımı "**You Only Look Once**", yani "**Sadece Bir Kez Bak**". Bu adın seçilmesinin nedeni algoritmanın nesne tespitini tek seferde yapabilecek kadar hızlı olmasıdır. YOLO algoritması çalışmaya başladığında görüntülerdeki 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 hesabı 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 olasığının oldukça düşük olduğudur. Diğer bir deyişle, YOLO yaptığı tahminin güvenilmez 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):

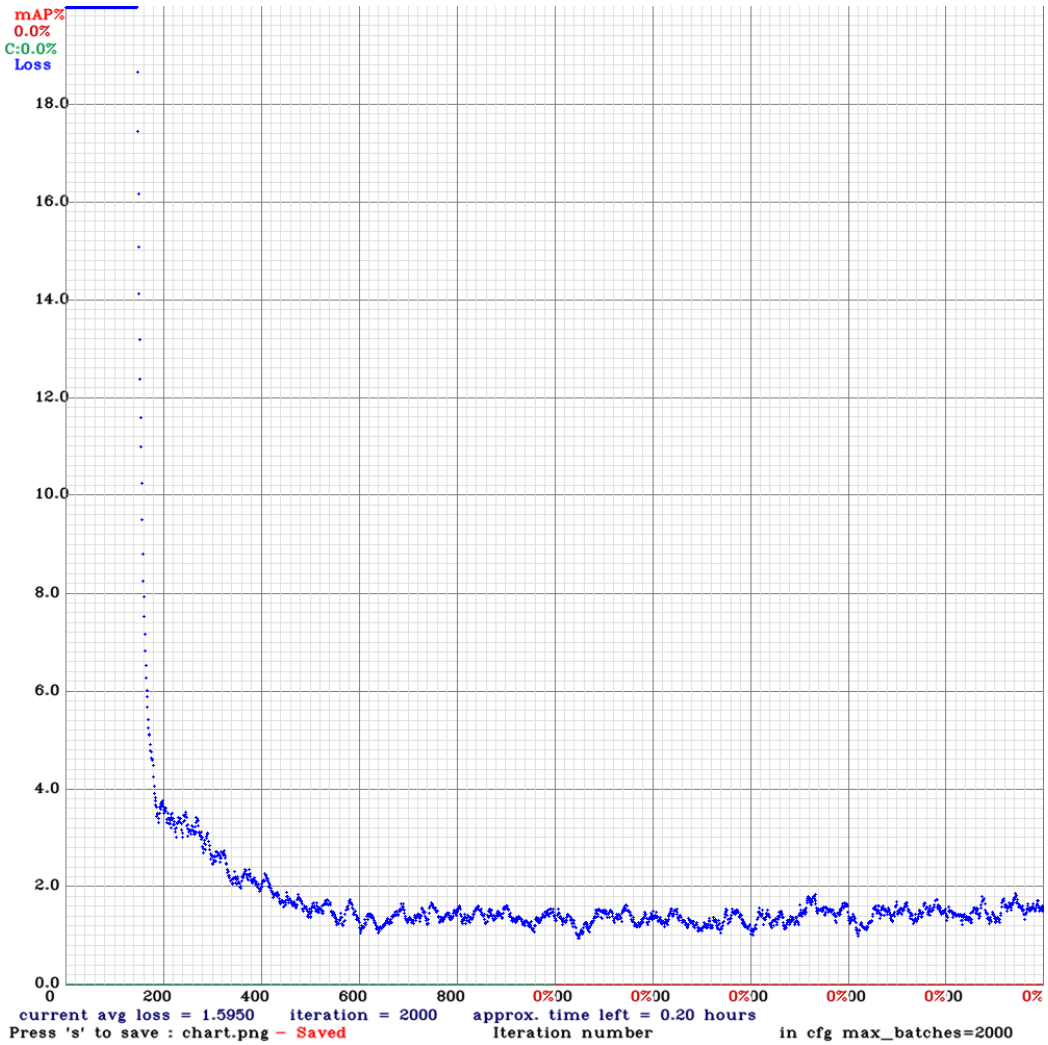
Şekil 4.2 Nesnelerin Tanımlanması

BÖLÜM 5

Viraj Tabelası Eğitim Sistemi Girdileri ve Çıktıları

5.1. Deneme 1

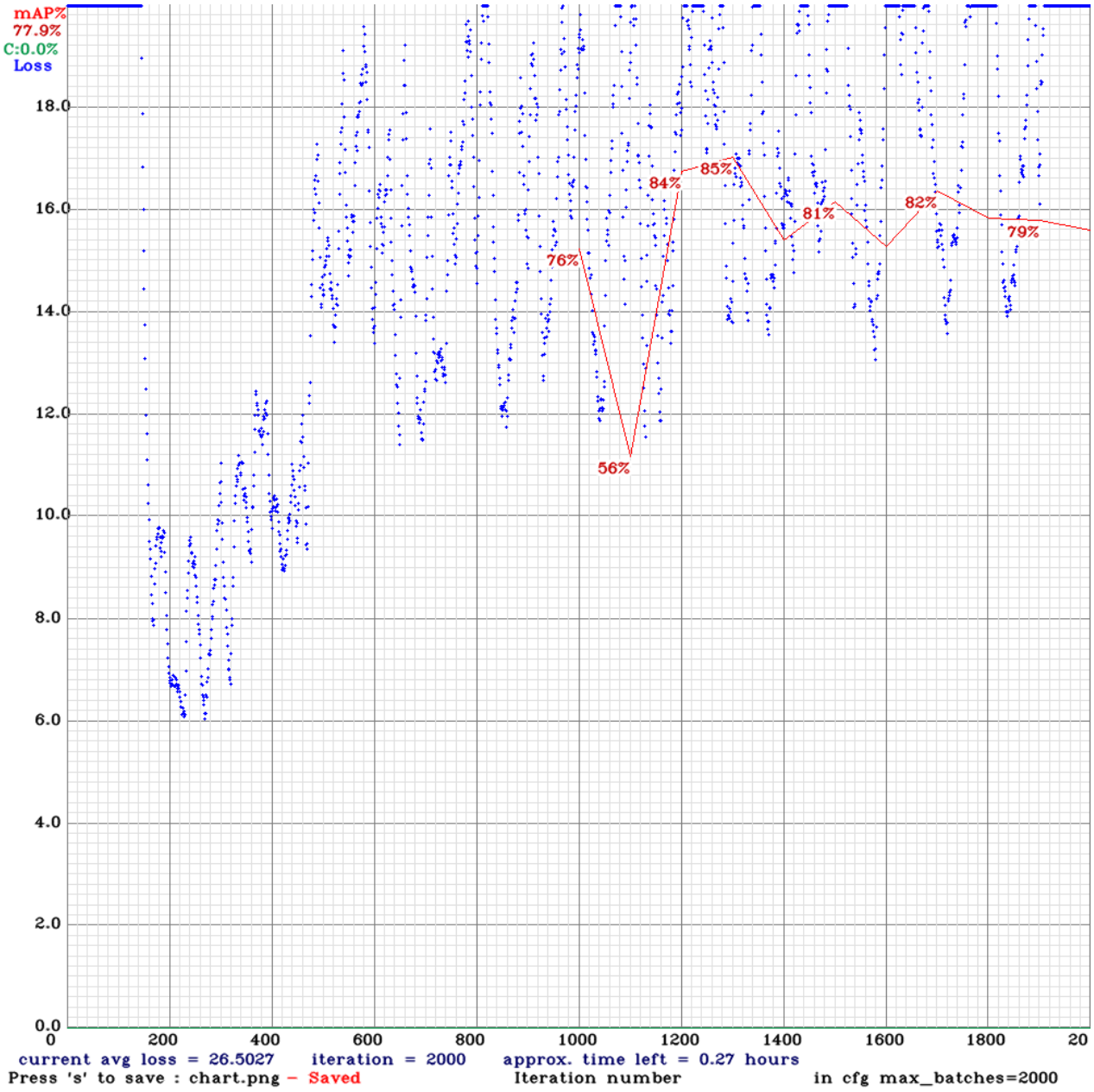
40 fotoğraf kullanılarak tanıtma 32 ‘si eğitim 8 tanesi test için kullanılmıştır.%0 güvenilirlik mevcuttur.



Şekil 5.1 Deneme-1

5.2. Deneme 2

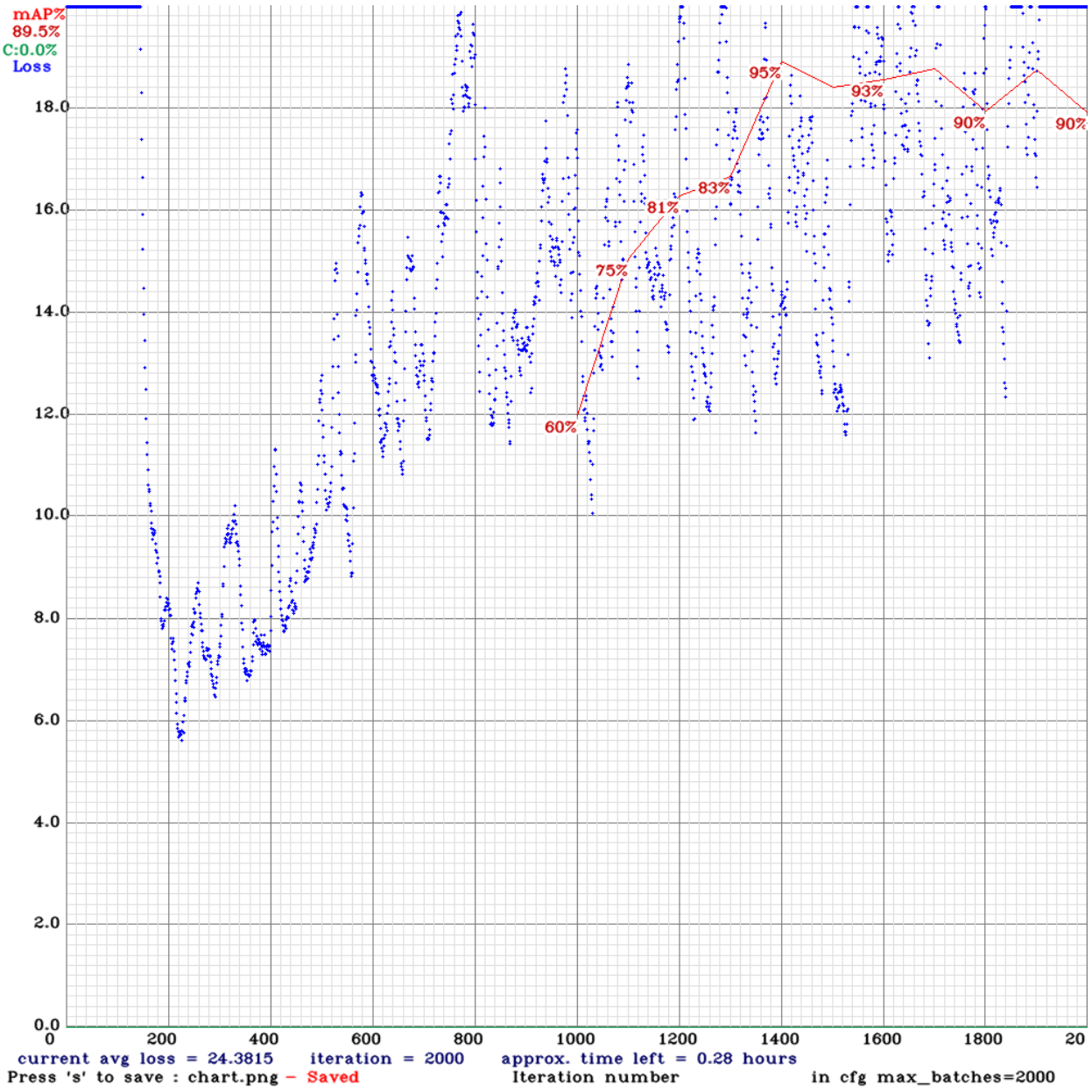
131 fotoğraf kullanılarak 105'i eğitimde 26 tanesi testte kullanılmıştır .%77.9 güvenilirlik mevcuttur.



Şekil 5.2 Deneme-2

5.2. Deneme 3

400 fotoğraf kullanılarak 320'si eğitimde 80 tanesi testte kullanılmıştır .%89.5 güvenilirlik mevcuttur.



Şekil 5.3 Deneme-3

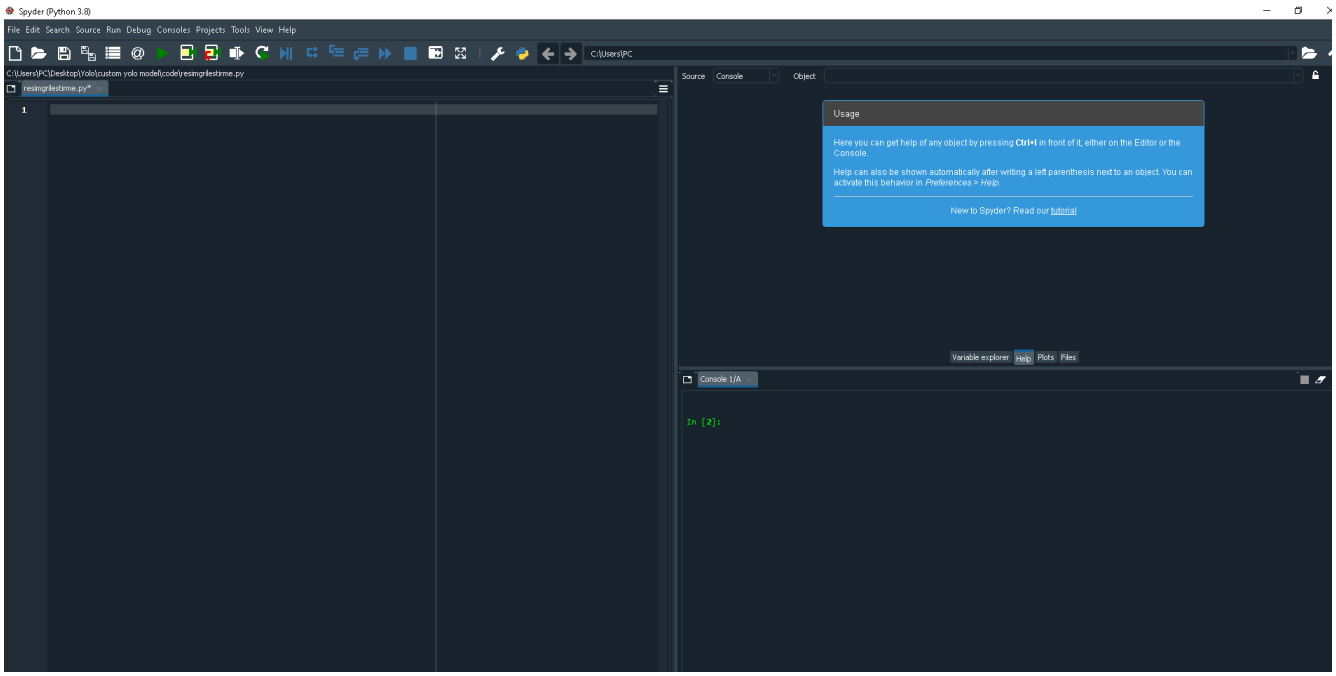
BÖLÜM 6

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.

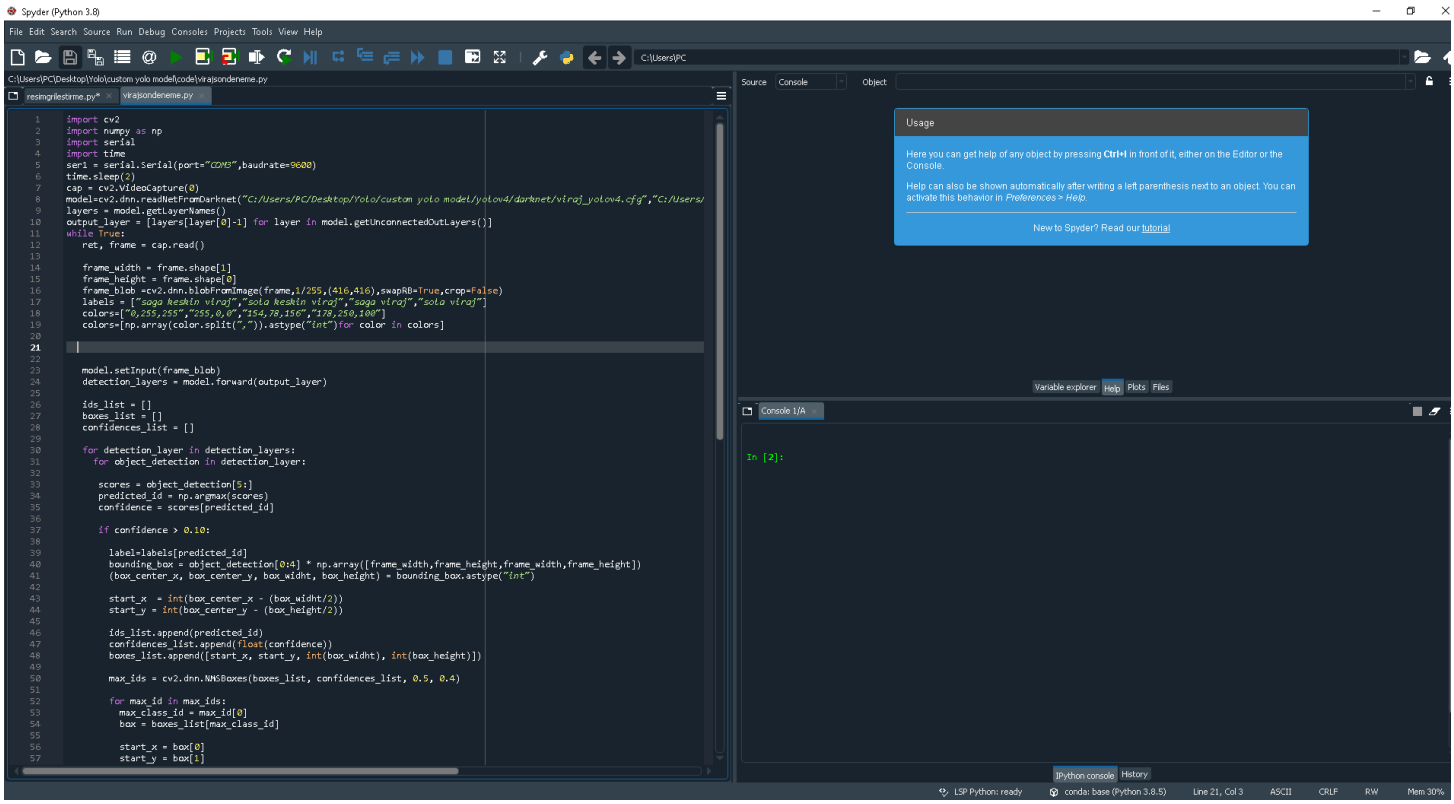
6.1. Spyder Arayüzü

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 6.1 Spyeder Arayüzü

6.2 Trafik Levhalarını Algılayan Sistemin Spyder4 Arayüzü ile Pythom Dilinde Kodlanması



Şekil 6.2 Python komutlarının Girdisi

6.2.1 Kodların İçeriklerinin Aktarılması

Kısım- 1

```
import cv2
import numpy as np
import serial
import time
```

Şekil 6.3 Kütüphane Girdisi

Kütüphanelerin import edilmesi sisteme yüklenmesi.(şekil 6.3)

Kısım-2

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

Şekil 6.4 Port Haberleşme

Arduino ile Seri port üzerinden sistemin haberleştirimesi(şekil 6.4)

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 6.5 Layers

Video üzerinden Görüntü alınması Yolo kütüphanesi oluşturulmuş dosya üzerinden alınan görüntünün layerlerının incelenmesi(Şekil 6.5)

Kısım-4

```
while True:
    ret, frame = cap.read()
    frame_width = frame.shape[1]
    frame_height = frame.shape[0]
    frame_blob =cv2.dnn.blobFromImage(frame,1/255,(416,416),swapRB=True,crop=False)
    labels = ["saga keskin viraj","sola keskin viraj","saga viraj","sola viraj"]
    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 6.6 Python Kodlar

İncelenen layerlar doğru iken çalıştırılan programda çerçevelerin analiz edilmesi çerçeve isimlerinin renk ve isimlerinin sisteme tanıtılması(şekil 6.6)

Kısım-5

```
ids_list = []
boxes_list = []
confidences_list = []

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([frame_width,frame_height,frame_width,frame_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)

     for max_id in max_ids:
       max_class_id = max_id[0]
       box = boxes_list[max_class_id]

       start_x = box[0]
       start_y = box[1]
       box_widht = box[2]
       box_height = box [3]

       predicted_id = ids_list[max_class_id]
       label = labels[predicted_id]
       confidence = confidences_list[max_class_id]

       end_x = start_x + box_widht
       end_y = start_y + box_height

       box_color = colors[predicted_id]
       box_color = [int(each) for each in box_color]
```

Şekil 6.7 Bounding Box

Sisteme çıtkı alınacak id lerin kutuların oranlarının listelerinin tanıtılması, For dönügüsü altınta Yolo üzerinden işlediğimiz görüntünün çerçevelerini bouinding 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 6.7)

Kısım-6

```
71
72
73              sag = "saga keskin viraj"
74              sol = "sola keskin viraj"
75              sa = "saga viraj"
76              so = "sola viraj"
77
78              if(confidence > 0.97 and label == sag):
79                  ser1.write("1".encode())
80
81
82              if(confidence > 0.97 and label == sol):
83                  ser1.write("2".encode())
84
85              if(confidence > 0.97 and label == sa):
86                  ser1.write("3".encode())
87
88
89              if(confidence > 0.97 and label == so):
90                  ser1.write("4".encode())
91
92
93
94              label = "{}: {:.2f}%".format(label, confidence*100)
95              print("predicted object {}".format(label))
96              cv2.rectangle(frame,(start_x,start_y),(end_x,end_y),box_color,2)
97              cv2.putText(frame,label,(start_x,start_y-10), cv2.FONT_HERSHEY_SIMPLEX, 0.5, box_color, 2)
98
99          cv2.imshow("Detection Window",frame)
100
101         if cv2.waitKey(1) & 0xFF == ord("q"):
102           break
103     ser1.close()
104     cap.release()
105     cv2.destroyAllWindows()
106
```

Şekil 6.8 Değerlerin Çıktısı

Çizdirilen bouind boxların güvenilirlik değerlerinin 0.97 den büyük olması durumunda virajın yön durumun belirtilecek şekilde sisteme tanıtılmış , sa,so veya sag,sol ifadesi altında arduinoya komut göndererek arduino ile haberleştrilmesi sağlanmıştır, eğer labelımız sag ifadesine eşitse arduinoya 1 çıktısı göndererek arduino 1 parametresi altındaki verileri işleyerek uygulamalı olarak çıktı almamızı sağlamaktadır.Komut dosyalarının tamamlanması çıktıların bize gösterilmesi ve pencerelerin kapatılması komutlarıylada python dilindeki kodlamalarımız son bulmaktadır.(Şekil 6.8)

BÖLÜM 7

ARDUİNO

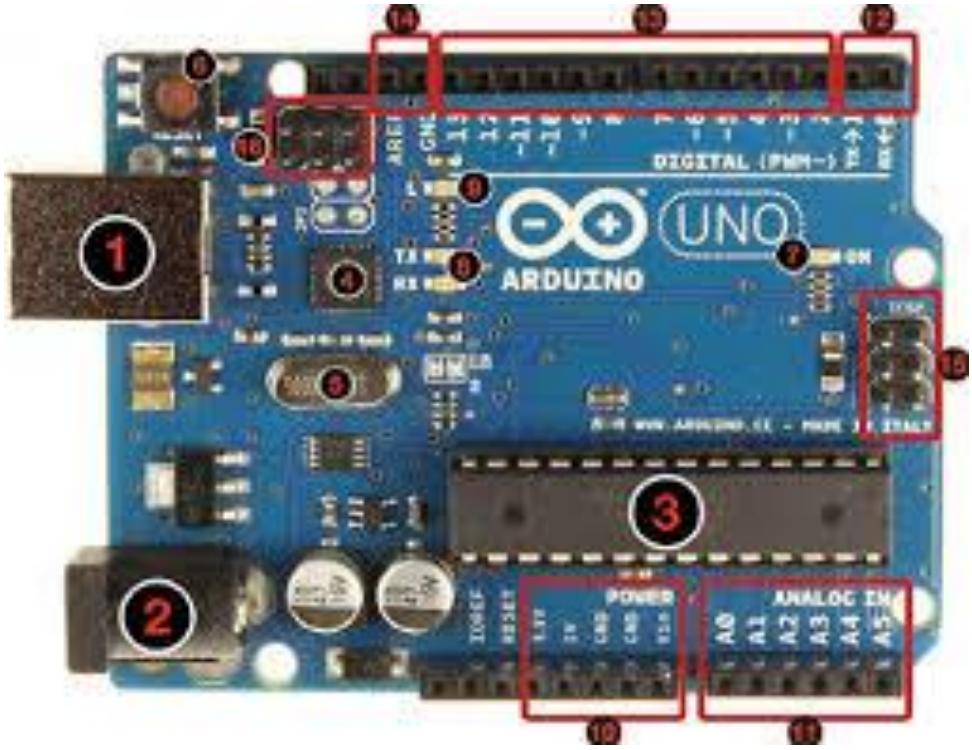
Arduino açık kaynaklı bir fiziksel programlama platformudur. Tek başına bir bilgisayar gibi düşünülebilir. Arduino sadece mikrodenetleyici kartlardan meydana gelmeyip geniş kütüphanesi, örnek projeleri ve öğrenimi kolay diliyle en çok tercih edilen proje geliştirme platformudur. Arduino' nun programlama 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.

7.1.Arduino Çeşitleri

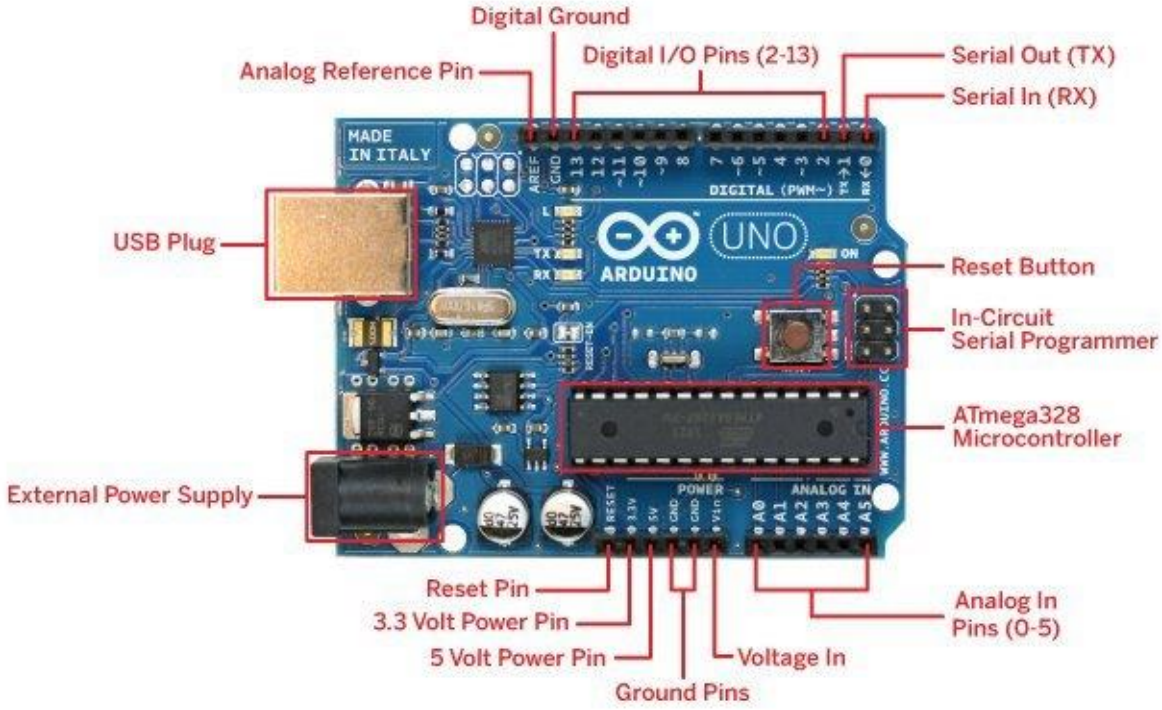
Arduino kartları kullanım alanlarına göre değişiklik göstermektedir. Arduino UNO; Arduino MEGA; Arduino LILYPAD; Arduino ADK, Arduino Leonardo, Arduino Mini, ArduinoNano, Arduino Yun bunlardan bazılarıdır. Yapılan çalışmada Atmega 328 mikroişlemciye sahip olan ArduinoUno kullanılmıştır.

7.2.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 7.1 Arduino Uno



Şekil 7.2 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

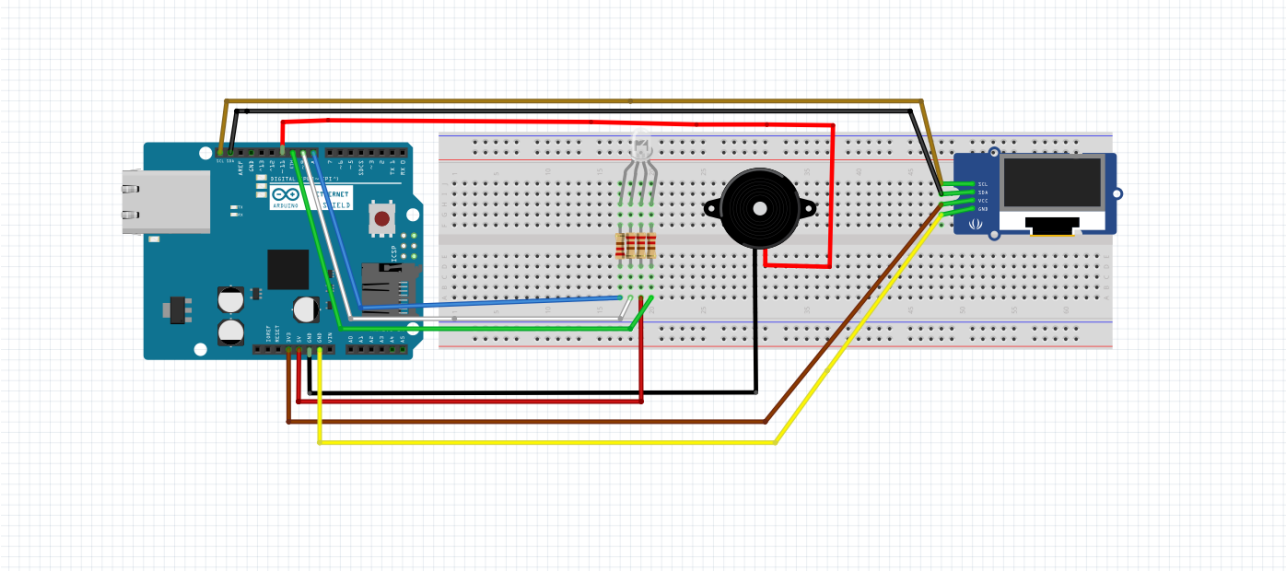
Şekil 7.3 Arduino Uno Sistem Bilgisi

7.2.1. 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.

7.3. Trafik Levha İkaz Sistemi Arduino Program Kodları ve Bağlantı şeması



Şekil 7.4 Frizting Bağlantı Şeması

```
#include <SPI.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#include <OLED_I2C.h>
```

Şekil 7.5 Kütüphane

- Kütüphanelerin programa tanıtıldığı kısım(Şekil 7.5)

```
OLED  myOLED(SDA, SCL, 8);
char viraj;
Adafruit_SSD1306 display(-1);

const unsigned char saga [] PROGMEM = {
  0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
  0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
  0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
  0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
  0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xc0, 0xfc, 0x0f, 0xe0, 0x7f, 0xe0, 0x7e, 0x07, 0xf0, 0x3f,
  0xf0, 0x3f, 0x03, 0xf8, 0x1f, 0xf8, 0x1f, 0x81, 0xfc, 0x1f, 0xfc, 0x0f, 0xc0, 0xfe, 0x0f, 0xfe,
  0x07, 0xe0, 0x7f, 0x07, 0xff, 0x03, 0xf0, 0x3f, 0x03, 0xfe, 0x07, 0xe0, 0x7e, 0x07, 0xfc, 0x0f,
  0xc0, 0xfc, 0x0f, 0xf8, 0x1f, 0x81, 0xf8, 0x1f, 0xf0, 0x3f, 0x03, 0xf0, 0x3f, 0xe0, 0x7e, 0x07,
  0xe0, 0x7f, 0xc0, 0xfc, 0x0f, 0xc0, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
  0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
  0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
  0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
  0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff
};
// '138', 40x40px
const unsigned char sola [] PROGMEM = {
  0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
  0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
  0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
  0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
  0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xfe, 0x07, 0xf0, 0x3f, 0x03, 0xfc, 0x0f, 0xe0, 0x7e, 0x07,
  0xf8, 0x1f, 0xc0, 0xfc, 0x0f, 0xf0, 0x3f, 0x81, 0xf8, 0x1f, 0xf0, 0x7f, 0x03, 0xf0, 0x3f, 0xe0,
  0xfe, 0x07, 0xe0, 0x7f, 0xc0, 0xfc, 0x0f, 0xc0, 0xff, 0xe0, 0x7e, 0x07, 0xe0, 0x7f, 0xf0, 0x3f,
```

Şekil 7.6 Şablonlar

- 128*64 oled ekranına sağa sola viraj resimlerinin gösterilmesi için şekillerin girdisi(Şekil 7.6)

```
void setup()
{
  display.begin(SSD1306_SWITCHCAPVCC, 0x3C);
  Wire.begin();

  // Tampon Bellek sıfırlama
  display.clearDisplay();




  pinMode(8, OUTPUT);
  pinMode(9, OUTPUT);
  pinMode(10, OUTPUT);
  pinMode(12, OUTPUT);
  Serial.begin(9600);
}
```

Şekil 7.7 Pin Ataması

- Oled ekranının başlatılması ve temizlenmesi, arduino üzerindeki 8-9-10 ve 12. Pinlerin çıkış olarak atanması. (Şekil 7.7)

```
void loop()
{

  digitalWrite(8, HIGH);
  digitalWrite(9, HIGH);
  digitalWrite(10, HIGH);
```

Şekil 7.8 Pin Değerleri

- 8,9,10. Pinlere bağlı rgb ledin başlangıçta yanmaması için pinlere 5 v değer girilmesi .(Şekil 7.8)

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


  if (viraj == '1') {
    digitalWrite(8, LOW);
    digitalWrite(9, HIGH);
    digitalWrite(10, HIGH);
    {
      digitalWrite(12, HIGH);
      delay(300);
      digitalWrite(12, LOW);
    }


    display.setTextSize(1, 4);
    display.setTextColor(WHITE);
    display.setCursor(10, 0);
    display.println("saga keskin viraj");
    display.drawBitmap(40, 30, saga, 40, 40, BLACK, WHITE);
    display.display();
    delay(1000);
    display.clearDisplay();
  }
```

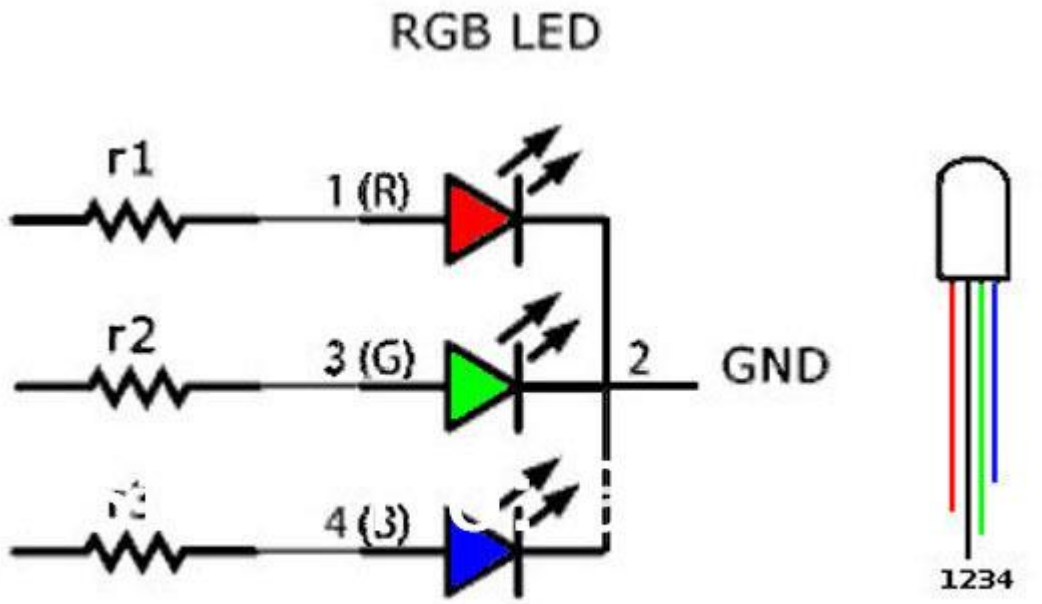
Şekil 7.9 Çıktılar

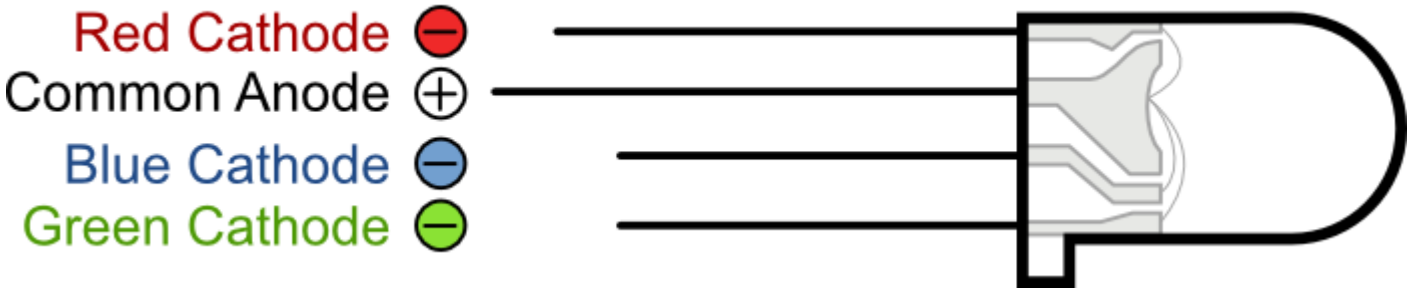
- Bu komutlarda da Python ile arduino nun serial haberleşmesi ve pythondaki viraj değerinin arduinodan okunması.Eğer viraj değeri bir ise lambaya bağlı pinlerin (8-9-10)değerlerinin belirlenmesi buzzerin ötmesi (12) ve oled ekranda 1 değeri altında istenilen viraj isminin yazarak şekiliyle birlikte göstermesi ve ekranın temizlenmesi kod edilmiştir.Diğer komutlarda bu düzeyde alınan girdiye göre değiştirilerek programa aktarılmıştır.(Şekil 7.9)

BÖLÜM 8

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.





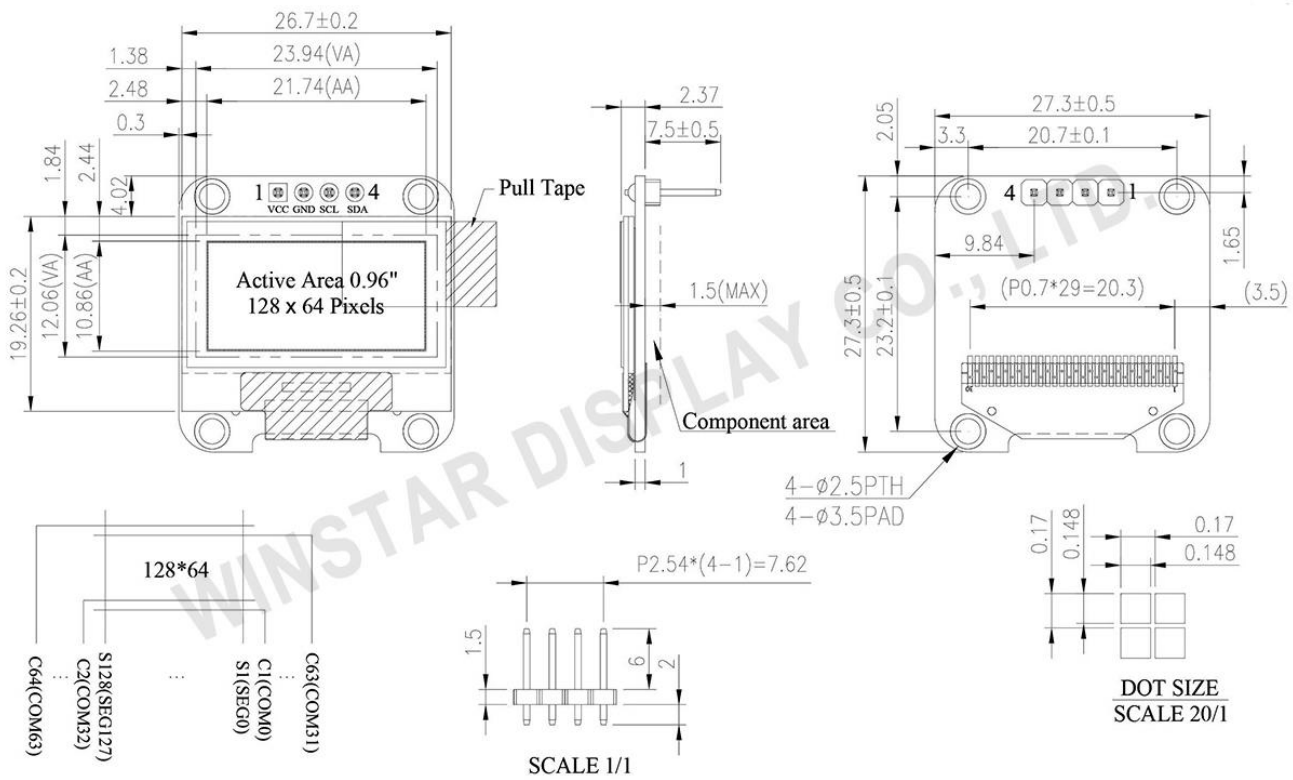
Şekil 8.1 RGB Led Yapısı

BÖLÜM 9

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 yöneltilmesiyle 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 9.1 128*64 Oled Datasheet

BÖLÜM 10

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 kullanılabilmektedirler. Nitekim tanımda da belirttiğimiz üzere, buzzerlar işitsel ikaz cihazı çeşitleridir. Işıklı buzzer, ışıksız buzzer, pasif buzzer ve aktif buzzer gibi türlere sahiptirler.

10.1 Çalışma Şekli

İlk önce buzzer direnç ve transistör kullanarak input pinine gelen dc voltu salınım sinyaline çevirir. İnduktör 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 10.1 Buzzer

SONUÇ

Sonuç olarak,Görüntü işlemede Trafik alanında geliştirilen projelerde,Genelden çok özele inerek trafik levhaları arasından viraj levhasını sisteme algılatarak,Gelişen teknolojiyle birlikte birbirinden farklı donanımlara sahip araçlara hem dahili hem harici montajı yapılabilecek bir sistem geliştirdim.Kaza sürücüye yol esnasında konforlu bir devamlılık sağlayarak kaza oranlarını azaltabilmeyi yolculuğu daha aktif hale getirmeyi amaçladım.Başarılı bir şekilde 2000 iterasyonda yaklaşık %90 verimle çalışan projemi sizlerin karşısına çalışır bir vaziyette sunmuş bulunmaktayım.

a)

b)

Şekil 11. a) b) Viraj Levhası Tanımlama

KAYNAKLAR

https://www.udemy.com/course/bilgisayarl-goru-yolov4-ile-nesne-tanma/

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

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

https://www.mediaclick.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