



**ARAÇ FARLARI İÇİN DİNAMİK BİR MODEL ÜZERİNDEN
OTOMATİK KONTROL YÖNTEMİYLE
YENİ BİR FAR SİSTEMİNİN GELİŞTİRİLMESİ**

YÜKSEK LİSANS TEZİ

Şafak YAŞAR

Danışman

Dr. Öğr. Üyesi Mustafa ŞAHİN

İkinci Danışman

Dr. Öğr. Üyesi Onur AKAR

ELEKTRİK ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI

Şubat 2023

AFYON KOCATEPE ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

YÜKSEK LİSANS TEZİ

ARAÇ FARLARI İÇİN DİNAMİK BİR MODEL ÜZERİNDEN
OTOMATİK KONTROL YÖNTEMİYLE
YENİ BİR FAR SİSTEMİNİN GELİŞTİRİLMESİ

Şafak YAŞAR

Danışman

Dr. Öğr. Üyesi Mustafa ŞAHİN

İkinci Danışman

Dr. Öğr. Üyesi Onur AKAR

ELEKTRİK ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI

Şubat 2023

TEZ ONAY SAYFASI

Şafak YAŞAR tarafından hazırlanan “Araç Farları İçin Dinamik Bir Model Üzerinden Otomatik Kontrol Yöntemiyle Yeni Bir Far Sisteminin Geliştirilmesi” adlı tez çalışması lisansüstü eğitim ve öğretim yönetmeliğinin ilgili maddeleri uyarınca 16/02/2023 tarihinde aşağıdaki jüri tarafından **oy birliği** ile Afyon Kocatepe Üniversitesi Fen Bilimleri Enstitüsü **Elektrik Elektronik Mühendisliği Anabilim Dalı’nda YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Danışman : Dr. Öğr. Üyesi Mustafa ŞAHİN

İkinci Danışman : Dr. Öğr. Üyesi Onur AKAR

Başkan : Prof. Dr. Yüksel Oğuz
Afyon Kocatepe Üniversitesi, Teknoloji Fakültesi

Üye : Dr. Öğr. Üyesi Mustafa ŞAHİN
Sağlık Bilimleri Üniversitesi, Hamidiye Sağlık Hizmetleri
Meslek Yüksekokulu

Üye : Dr. Öğr. Üyesi Süreyya KOCABEY
Sağlık Bilimleri Üniversitesi, Hamidiye Sağlık Hizmetleri
Meslek Yüksekokulu

Afyon Kocatepe Üniversitesi
Fen Bilimleri Enstitüsü Yönetim Kurulu’nun
..... /..... /..... tarih ve
..... sayılı kararıyla onaylanmıştır.

.....

Prof. Dr. İbrahim EROL
Enstitü Müdürü

BİLİMSEL ETİK BİLDİRİM SAYFASI
Afyon Kocatepe Üniversitesi

Fen Bilimleri Enstitüsü, tez yazım kurallarına uygun olarak hazırladığım bu tez çalışmada;

- Tez içindeki bütün bilgi ve belgeleri akademik kurallar çerçevesinde elde ettiğimi,
- Görsel, işitsel ve yazılı tüm bilgi ve sonuçları bilimsel ahlak kurallarına uygun olarak sunduğumu,
- Başkalarının eserlerinden yararlanılması durumunda ilgili eserlere bilimsel normlara uygun olarak atıfta bulunduğumu,
- Atıfta bulunduğum eserlerin tümünü kaynak olarak gösterdiğimi,
- Kullanılan verilerde herhangi bir tahrifat yapmadığımı,
- Ve bu tezin herhangi bir bölümünü bu üniversite veya başka bir üniversitede başka bir tez çalışması olarak sunmadığımı

beyan ederim.

17 / 02 / 2023

Şafak YAŞAR

ÖZET

Yüksek Lisans Tezi

ARAÇ FARLARI İÇİN DİNAMİK BİR MODEL ÜZERİNDEN OTOMATİK KONTROL YÖNTEMİYLE YENİ BİR FAR SİSTEMİNİN GELİŞTİRİLMESİ

Şafak YAŞAR

Afyon Kocatepe Üniversitesi

Fen Bilimleri Enstitüsü

Elektrik Elektronik Mühendisliği Anabilim Dalı

Danışman: Dr. Öğr. Üyesi Mustafa ŞAHİN

İkinci Danışman: Dr. Öğr. Üyesi Onur AKAR

Bu tez çalışmasında mevcut otomobillerde kullanılan standart far teknolojisine alternatif olarak daha sofistike bir far tasarımı yapılmıştır. Bu tasarım düşük ışık koşullarında kısa ve uzun farların dinamik olarak kontrol edilmesini sağlayarak daha güvenli bir sürüş ortamı yaratmaktadır. Bu amaçla LED'lerin hücresel kontrol esnekliğinden yararlanılmıştır. Kullanılan sensörlerin ve şerit tespit sisteminin, araç farlarının pozisyon kontrolünü sağlayabilmesi sayesinde de topografya fark etmeksizin eğimli ve virajlı yollarda görüş açısı doğrultusunda aydınlatma sağlanmıştır. Aynı zamanda sürücünün gece sürüşlerinde trafik işaretlerini görmeme ihtimalini ortadan kaldırmak adına makine öğrenimine dayalı olarak trafik işareti algılama teknolojisi sisteme entegre edilmiştir. Oluşturulan sistemin far pozisyonunu yukarı aşağı ve sağa sola ayarlayabilmesi, araç tespiti ile kısa hüzmenin yanında uzun hüzmenin de karşıdan gelen sürücüyü rahatsız etmeden kullanabilmesi, trafik işaretini tanıyabilmesi ve bunu sürücü ekranında gösterebilmesi kaza için oluşabilecek birçok etkeni ortadan kaldırmıştır. Ayrıca gece araç algılama ve trafik işareti tanıyabilme fonksiyonları için makine öğrenimiyle oluşturulan modeller sistemin doğruluk oranını doğrudan arttırmıştır.

2023, ix + 66 sayfa

Anahtar Kelimeler: Araç aydınlatma sistemi, Makine öğrenmesi, Trafik işareti tanıma, Otoyol şerit tanıma, Araç tanıma, Görüntü işleme.

ABSTRACT

M.Sc. Thesis

DEVELOPMENT OF A NEW HEADLIGHT SYSTEM VIA AUTOMATIC CONTROL METHOD WITH DYNAMIC MODEL FOR VEHICLE HEADLIGHTS

Şafak YAŞAR

Afyon Kocatepe University

Graduate School of Natural and Applied Sciences

Department of Electrical and Electronics Engineering

Supervisor: Asst. Prof. Mustafa ŞAHİN

Co-Supervisor: Asst. Prof. Onur AKAR

The novelty in this study is the design and presentation of the standard headlight technology used in current vehicles with a more sophisticated approach. The current design ensured lighting in the direction of the angle of vision at inclined and curved roads regardless of topography through headlight position control using the flexibility of LEDs in order to create a more secure driving environment under low light conditions, as well as presented compulsory low beam and high beam headlights. In addition, traffic-sign detection technology based on machine learning was integrated into the system in order to eliminate the possibility of non-recognition of the traffic-signs by the driver. The features such as adjustment of the headlight's position to up-down and right-left, vehicle detection, flexibility of the use of low beam as well as high beam without discomforting the driver approaching from the opposite direction, traffic-sign recognition and displaying of it on the driver information display eliminated many factors that may cause accidents. Moreover, the models formed through machine learning for the night-time vehicle detection and traffic-sign recognition functions directly increased the system's accuracy rate.

2023, ix + 66 pages

Keywords: Vehicle headlight system, Machine learning, Traffic-sign recognition, Lane detection, Vehicle recognition, Image processing.

TEŞEKKÜR

Bu araştırmanın konusu, deneysel çalışmaların yönlendirilmesi, sonuçların değerlendirilmesi ve yazımı aşamasında yapmış olduğu büyük katkılarından dolayı tez danışmanım Sayın Dr. Öğr. Üyesi Mustafa ŞAHİN'e ve ikinci danışmanım Dr. Öğr. Üyesi Onur AKAR'a, her konuda öneri ve eleştirileriyle yardımlarını gördüğüm hocalarıma ve sevgili eşim Öğr. Gör. Şeyma ŞAFAK YAŞAR'a teşekkür ederim.

Bu araştırma boyunca maddi ve manevi desteklerinden dolayı aileme teşekkür ederim.

Şafak YAŞAR
Afyonkarahisar 2023

İÇİNDEKİLER DİZİNİ

	Sayfa
ÖZET	i
ABSTRACT	ii
TEŞEKKÜR	iii
İÇİNDEKİLER DİZİNİ.....	iv
SİMGELER ve KISALTMALAR DİZİNİ	vi
ŞEKİLLER DİZİNİ	vii
ÇİZELGELER DİZİNİ.....	viii
RESİMLER DİZİNİ	ix
1. GİRİŞ.....	1
2. LİTERATÜR BİLGİLERİ	6
3. MATERYAL ve METOT	12
3.1 Yazılım ve Donanım Karakteristikleri.....	12
3.1.1 Arduino Uno	12
3.1.2 WS2812 8x4 LED Matris.....	13
3.1.3 MG995 Servo motor	14
3.1.4 Kamera	15
3.1.5 Eğim Sensörü	16
3.1.6 Açık Sensörü	17
3.2 Tensorflow ile Araç Tanıma	18
3.2.1 COCO Veri Seti Modeli.....	19
3.2.2 Arduino ile Uzun Far Kontrolü	20
3.2.3 Arduino ile Far Pozisyonunun Kontrolü	21
3.3 Trafik İşareti Tanıma	21
3.3.1 Derin Öğrenme.....	23
3.3.2 Denetimli Öğrenme.....	24
3.3.3 Evrişimli Sinir Ağı (CNN) Modeli Oluşturma.....	24
3.3.3.1 Conv2D Sınıfının Önemli Parametreleri	25
3.3.3.2 MaxPooling2D Sınıfının Önemli Parametreleri.....	26
3.3.3.3 Flatten Sınıfı	26
3.3.3.4 Dense Sınıfı	27
3.3.3.5 Aktivasyon Fonksiyonları	27
3.3.4 Optimizasyon Algoritması	29

3.3.5 Görüntü İşleme Prosesi	29
3.4 Şerit Çizgisi Tespiti	30
3.4.1 Görüntü Filtreleme	30
3.4.2 Canny Kenar Tespiti	30
3.4.3 Görüntüdeki İlgili Alanların Belirlenmesi (ROI).....	31
3.4.4 Hough Dönüşümü	31
3.4.5 Şerit Çizgilerinin Tespiti ve Eğrilik Hesabı	32
4. BULGULAR	34
4.1 Adaptif Far Tasarımı.....	35
4.2 Araç Tespit Uygulaması	38
4.3 Trafik İşareti Tanıma Uygulaması	41
4.4 Şerit Algılama Uygulaması.....	46
4.4.1 Görüntü İşleme Proses Süreci	46
5. TARTIŞMA ve SONUÇ	51
6. KAYNAKLAR.....	55
ÖZGEÇMİŞ.....	58
EKLER	59

SİMGELER ve KISALTMALAR DİZİNİ

Simgeler

$R(z)$	ReLU fonksiyonu
$\sigma(z)$	Softmax fonksiyonu
z	Yapay sinir ağında gelen görüntü değeri
G_x	Yatay gradyan değeri
G_y	Dikey gradyan değeri
$G(x,y)$	2D Gaussian fonksiyonu

Kısaltmalar

API	Application Programming Interface
AREF	Analog Reference
AVR	Augmented Voltage Right
CNN	Convolutional Neural Network
COCO	Common Objects in Context
CPU	Central Process Unit
GPU	Global Photographic Union
GSM	Global System for Mobile Communications
HID	High Intensity Discharge
IDE	Integrated Development Environment
LDR	Light Dependent Resistor
MLP	Multilayer Perceptron
MP	MegaPixel
PIR	Passive Infrared
PWM	Pulse Width Modulation
ReLU	Rectified Linear Unit
RGB	Red Green Blue
ROI	Region of Interest
USB	Universal Serial Bus

ŞEKİLLER DİZİNİ

	Sayfa
Şekil 1.1 Geçici körlüğe sebep olan ışık hüzmelerinin yansıması.....	4
Şekil 1.2 Geçici körlüğe sebep olan ışık hüzmelerinin yansıması	5
Şekil 3.1 8x4 5050 RGB LED modül kartının arka ve ön tarafı	15
Şekil 3.2 MG995 servo motorun kablo bağlantısı.....	15
Şekil 3.3 Servo motorun PWM sinyali ile kontrol edilmesi.....	16
Şekil 3.4 LabelImg uygulamasından araç farı etiketleme	19
Şekil 3.5 LED pozisyonunu belirleyen örnek algoritma	22
Şekil 3.6 Trafik işareti tanıma akış diyagramı	23
Şekil 3.7 Yapay sinir ağı modelinin yapısı.....	24
Şekil 3.8 CNN ağı mimarisi	26
Şekil 3.9 ReLU fonksiyon grafiği	29
Şekil 3.10 Sigmoid fonksiyon grafiği.....	29
Şekil 3.11 Sistemin akış diyagramı	34
Şekil 4.1 Geleneksel far aydınlatma alanı (a), adaptif far aydınlatma alanı (b).....	37
Şekil 4.2 Adaptif far donanım sistemi.....	38
Şekil 4.3 Araç tanıma ve uzun far sistemi.....	41
Şekil 4.4 Araç tanıma algoritmasında farların davranışının görüntülenmesi.....	42
Şekil 4.5 Evrişimli sinir ağı yapısı(CNN)	43
Şekil 4.6 Veri setinin eğitim ve validasyon kayıp grafiği	43
Şekil 4.7 Veri setinin eğitim ve validasyon doğruluk grafiği	44
Şekil 4.8 Görüntü filtreleme proses adımları	47
Şekil 4.9 Canny ile kenar tespiti.....	48
Şekil 4.10 Soldaki ROI, sağdaki kuşbakışı görüntü.....	48
Şekil 4.11 Görüntünün Hough dönüşüm uygulanmış çıktısı	49
Şekil 4.12 Görüntüdeki tespit edilen beyaz piksellerin histogram grafiği	49
Şekil 4.13 Kayan pencereler uygulaması gösterimi	50
Şekil 4.14 Reel şerit genişliği ve uzunluğu	51
Şekil 4.15 Şerit çizgilerin tespit edilmesi.....	51
Şekil 4.16 Sürücü multimedya ekranı	52

ÇİZELGELER DİZİNİ

	Sayfa
Çizelge 3.1 Ardunio Uno kartının özellikleri.....	133
Çizelge 4.1 CNN modeli trafik işareti tanıma algoritması test sonuçları.....	444



RESİMLER DİZİNİ

	Sayfa
Resim 3.1 Proje prototipinde kullanılan webcam kamerası.....	16
Resim 3.2 Proje prototipinde kullanılan tilt sensörü.....	17



1. GİRİŞ

İlk araba 1880'lerde sahneye çıktığından beridir far, otomobil tarihi boyunca tüm araçların önemli bir parçası olmuştur. Gece veya düşük ışık koşullarında güvenli sürüş söz konusu olduğunda farların varlığı mutlak bir gereklilik olmuştur ve bugün onlarsız bir otomobilin nasıl olacağını hayal etmek zor olmaktadır. Far, tipik olarak otomobillerdeki en önemli güvenlik cihazlarından biri olarak kabul edilmektedir. Ölümlü kazaların yaklaşık %40'ının gece meydana geldiği ve azalan görsel performansın en büyük etken olduğu bildirilmektedir (Bullough vd. 2013). Tabi ki geçmişte kullanılan farlar ile günümüzde kullanılan farların arasında teknolojik farklar zamanla oluşmuştur.

İlk otomobillerde petrol ve asetilen gazıyla çalışan farlar bulunmaktaydı. O dönemde yaşayan insanların evlerde kullanmış olduğu gaz lambalarını anımsatan bu farlar proto farlar olarak adlandırılmaktadır. Bu farlar, dağınık bir ışığın öndeki yolu aydınlatmasına izin veren yansıtıcı bir aynaya sahip bir fenerden yapılmıştır. O zamanlarda rüzgara ve yağmura dayanıklı olması sebebiyle çok popüler olsalar da, asetilen gazıyla çalışan bu farların oldukça zehirli bir madde olan yakıcı kireç oluşumunu önlemek için sık sık temizlenmesi gerekmektedir. Bu farlar ideal bir far olmasa da elektrikli farlar keşfedilene kadar sürücülerin en büyük yardımcısı olmuşlardır. İlk olarak Elektrikli Araç Şirketi (Electric Vehicle Company) vasıtasıyla farlar isteğe bağlı bir aksesuar olarak kullanıcıya sunulmaktaydı. Tarihte belli bir süre aynı teknoloji kullanılmıştır. Amerika 2. Dünya Savaşı'na girmeden önce kapalı hüzmeli farlar dünyaya tanıtılmıştır. Bu far tasarımı sürücülere önceki farlara göre daha çok odaklanmış bir ışık hüzmesi sağlamıştır (İnt. Kyn. 1).

Araba gövde altı ışıkları olarak da bilinen kısa hüzmeli farlar, Guide Lamp şirketi tarafından 1915'te tanıtılmıştır. Ancak Cadillac'ın sürücülerinin uzun fardan kısa fara basit bir şekilde geçiş yapmasına yardımcı olan sistemi 1917'ye kadar popüler olamamıştır. 1927'de, ışığı yürüyerek ayarlamak için uygun ve akıllı sistem tanıtılmıştır. Ayakla çalıştırılan bu sistemi kullanan son araba, 1991 Ford F-Serisi idi. 1938'de Cadillac, tasarımlarında otomotiv sis farları kullanılmıştır. Aynı şirket, kısa far ve uzun

far arasında geçişe izin veren otomatik sistemi icat etmiştir. 17 yıl boyunca Amerikan hükümeti, lambanın 7 inç boyutunu zorunlu kılmıştır ve bu süre zarfında yeniliğe karşı çıkmıştır. 1957'de farlarla alakalı olan yasa, yolu düzgün bir şekilde aydınlattıkları sürece farklı boyut ve şekillerde ışıklara izin verecek şekilde değiştirilmiştir. Dönme hareketi yapabilen ilk otomobil farı 1948 yılında icat edilmesine rağmen ancak 2003 yılından itibaren otomobillerde kullanılmaya başlanmıştır(Aydın ve Ünaldı 2010).

1960'lı yıllara gelindiğinde ise, halojen kapalı farlar Avrupa'da standart haline getirilmiştir. Halojen gazının eklenmesiyle daha uzun süren parlaklıklar sağlanmıştır. Halojen ampullerde filaman, halojen gazıyla dolu kuvars bir kapsül içine alınmaktadır. Bu gaz inerttir ve iyot ve bromdan oluşmaktadır. Buharlaştırılmış tungsten, tungsten-halojen molekülleri oluşturmak için cam zarf içindeki halojen gazıyla birleşmektedir. Bu moleküller daha sonra filamana geri dönerek cam zarfın kararmasını ortadan kaldırmaktadır. Tungsten yeniden birikmekte ve filaman üzerinde geri dönüştürülmektedir. Tungsten filamanı güçlendirmekte, lambanın ömrünü uzatmakta ve halojen gazı daha sonra döngüyü tekrar başlatmak için serbest kalmaktadır. Ancak bu far teknolojisi Amerikalılar için 70'lerin sonuna kadar standart olmamıştır (İnt. Kyn. 2).

Bu teknoloji 1983'te yerini kompozit far teknolojisine bırakmıştır. Kompozit farlardan önce ampullerinizden biri işlevsiz kaldığında yani yandığında tüm seti değiştirmeniz gerekmekteydi. Tüm seti değiştirmek kullanıcıya büyük maddi yükler getirmektedir. Federal Motorlu Taşıt Güvenlik Standardı, tüm kompozit far tertibatlarının standart olmayan bir şekle ve yedek ampullere sahip olmasına izin verecek şekilde değiştirildi. Bu değiştirilen standartlar sayesinde değiştirilebilir halojen farlar kısa süre içinde çok tercih edilmiştir. Halojen farlar zamanla yerini Xenon farlara bırakmıştır (İnt. Kyn. 1).

Resmi olarak Yüksek Yoğunluklu Deşarj (HID) farlar olarak bilinen Xenon farlar, renk sıcaklıkları ve ışık çıkışları nedeniyle halojen lambalardan daha uygun bir çözüm olarak kabul edilmiştir. İlk olarak 1991 yılında BMW 7 Serisi'nde ortaya çıkmışlardır ve giderek birçok otomobil üreticisinin bir numaralı tercihi haline gelmişlerdir. HID lambaların çalışma prensibi neon ampullerle aynıdır. Her iki ucunda gaz ve elektrotlarla dolu kapalı bir ampul ve içinden akan bir elektrik akımı yardımıyla aydınlatma

sağlamaktadırlar. Arabalardaki HID farlar, şeffaf bir kuvars kabuk, bir tungsten elektrot ve iki elektrot arasında akan yüksek voltajlı bir akım tarafından tahrik edilen bir gaz karışımından oluşmaktadır. Halojen farlarla karşılaştırıldığında, HID farlar daha parlaktır (yaklaşık üç kat daha parlak) ve çok daha uzun ömürlü olmaktadır. Farlarınıza beklenmedik bir şey olmazsa, muhtemelen HID, araca sahip olmanızdan bile daha uzun süre dayanmaktadır. Bu, bakım için para harcamanıza gerek olmadığı anlamına gelmekte, ancak bir şekilde hasar görürlerse, onları değiştirmek için epeyce para harcamanız gerekmektedir. HID farların ışığı da daha eşit dağılmaktadır, yolun her iki tarafı da iyi aydınlatılabilmektedir. Geceleri tehlikeye neden olabilecek yayaların, bisikletçilerin veya hayvanların olduğu bir bölgede yaşıyorsanız bu gerçekten fayda sağlamaktadır. Farların kazanmış olduğu diğer bir yetenek LED farlardır.

LED, yakın zamanda geliştirilmiş yeni bir teknoloji farıdır. Xenon gibi gazla veya halojen gibi akkorla parlamak yerine, bir uyarma akımı uygulandığında LED farlar küçük diyotlarla parlamaktadır. Bu tür bir projektör yalnızca çok az miktarda enerji gerektirmektedir. Ancak yarı iletken diyotlardan oluştuğu için önemli miktarda ısı açığa çıkmaktadır. LED farlar aydınlatmak için daha az enerji tükettiğinden, üreticiler bu teknolojiyi kullanmayı tercih etmektedirler. Çünkü alternatörün onlara güç sağlamak için çok çalışması gerekmemektedir. 11-22 yıl öncesine kadar değişim gerektirmeden çalışabilirler ve montajı çok kolay yapılabilir. Bu durum, araç sahipleri için daha az maliyet demektir. LED farların ışığı, HID veya halojene göre daha az kör edici olduğundan, yoldaki karşı araçlar için daha güvenli olmaktadır. Bunun dışında, araç sahiplerinin tercihlerine göre kurulumu ve değiştirilmesi çok kolaydır (İnt. Kyn. 2)(Long vd.2015).

Lazer farlar, en son araç içi aydınlatma teknolojisidir. Şu anda BMW i8 veya Audi R8 gibi yalnızca çok az sayıda süper otomobil modeli bu teknolojiyle donatılmıştır. Lazer farlar, LED'lerden 1000 kat daha güçlü bir ışık demeti üretebilmekte, ancak LED'lerden yalnızca 2/3 oranında, hatta yarısı kadar güç tüketmektedir. BMW i8'deki lazer ışığı, LED ışıklar kullanıldığında 300 metreye kıyasla otomobilin önündeki 600 metrelik bir mesafeyi aydınlatabilmektedir. Bu farların dezavantajı ise teknolojinin henüz ana akım haline gelmemesi, bu nedenle fiyatın yüksek olması ve orta pazara uymaması olmuştur.

Günümüzde, trafik kazaları insan hayatında önemli bir sorun olmaya devam etmektedir. Özellikle karanlık ve dağlık alanların dar, dik ve virajlı alanlarında sıkça kazalara rastlanmaktadır.(Bullough vd. 2011) Bu kazaların en başlıca sebeplerinden biri araç farına bağlı sürücü görüşünün zayıf olmasıdır. Karşı yöndeki araçlardan gelen ışık, sürücülerin birkaç saniye kör olmasına neden olmaktadır. Yine sürücü, normal ışığı aldıktan birkaç saniye sonra her şeyi net bir şekilde görebilir. Bu durumda kazaya sebebiyet verebilecek önemli konulardan biridir.

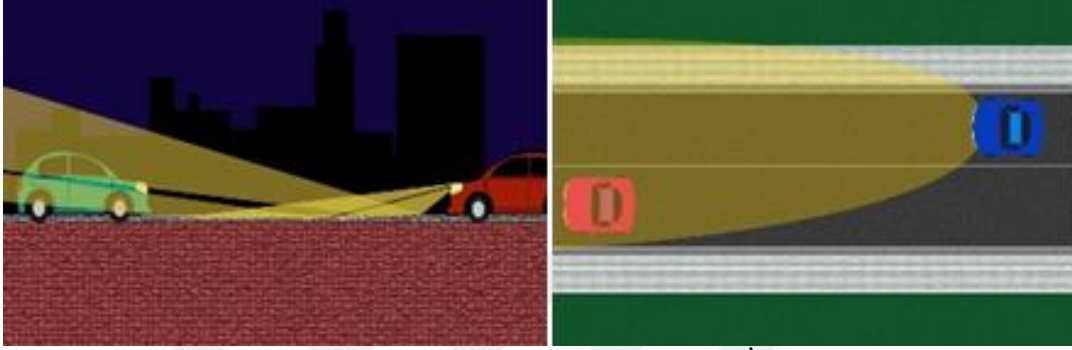
Sürücüler, özellikle şehirlerarası ulaşımında, karşı yönden gelen sürücüler için; uzun hüzmeyi veya kısa hüzmeyi veya tersini değiştirmeye çok dikkat etmektedir. Bu durumsa, ters yönde giden sürücüler için çok fazla sorun yaratmaktadır. Gerekğinde otomatik olarak uzun farı kısa fara değiştiren yarı akıllı farlara sahip birkaç lüks otomobil günümüzde mevcuttur. Ancak bu teknoloji, ekonomik otomobil modelleri tarafından henüz uygulamada ne yazık ki mevcut değildir.

Tarih boyunca tüm bu gelişmeler olurken teknolojinin de gelişmesiyle birlikte araçların daha akıllı olması ve bazı durumlarda otonom olarak hareket edebilmesi bir ihtiyaç haline gelmiştir. Söz konusu otomobil olduğunda kaza ve ölüm riski göz ardı edilememektedir.

Trafik kazaları insan hayatını etkileyen en önemli sorunların başında gelmektedir. Türkiye’deki ölümlü kazaların yaklaşık %10’u arkadan çarpışmalardır (İnt Kyn. 3). Çok sayıda kaza ve sebepleri incelendiğinde bu kazaların genellikle insan görüşünün ve dikkatinin mükemmel olmamasından kaynaklandığı görülmüştür. Ayrıca topografyanın da bu kazaların oluşmasındaki rolü kaçınılmaz bir gerçektir. Topografyanın zorlu olduğu noktalarda aydınlatmaların yetersiz oluşu ve kör noktaların varlığı da kazaların en büyük nedenleri arasında gelmektedir. Kör noktalar nedeniyle bu alanlar ileriye bakılarak veya yan aynalardan bakılarak tam olarak görülememektedir. Aynı zamanda yollarda var olan ani virajlar, kötü altyapı ve yetersiz aydınlatma da kör noktaların oluşmasına neden olmaktadır. Otomobil kullanmak öncelikle görsel bir görevdir çünkü görme fonksiyonu sürüş için gereken koşulların büyük bir çoğunluğuna katkıda bulunur. Görme fonksiyonunun araç kullanırken önemi göz önüne alındığında araç

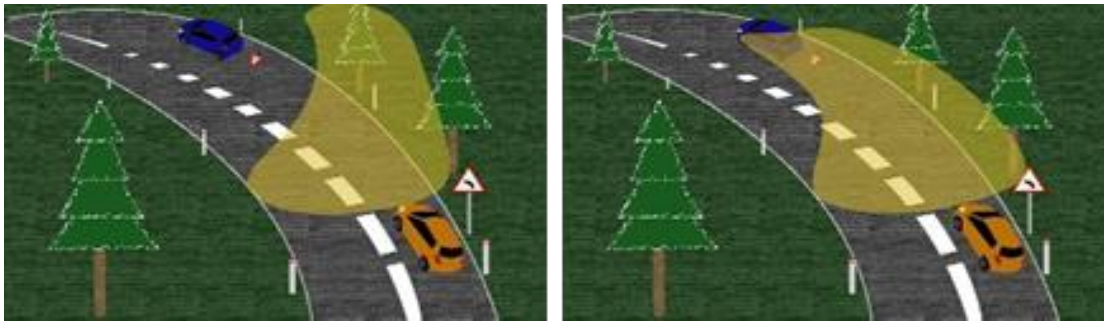
aydınlatmasının kalitesi ve esnekliği kaza riskini doğrudan artmasına sebep olacağı kaçınılmaz bir gerçektir.

Şekil 1.1’de görüldüğü gibi özellikle gece ve karanlık ortamlarda çift şeritli yollarda karşı yöndeki araçlardan gelen ışık, sürücülerin birkaç saniye kör olmasına neden olabilmektedir.



Şekil 1.1 Geçici körlüğe sebep olan ışık hüzmelerinin yansıması (İnt. Kyn. 4).

Sürücüler, şehirlerarası ulaşımında, karşı yönden gelen araçlar için; uzun farı kısa fara veya tersini değiştirmeye dikkat etmek zorundadır. Bu durumsa, ters yönde olan sürücüler için sorunlara neden olmaktadır. Değişim esnasında dikkat dağınıklığı olabileceği gibi anlık körlük durumları da meydana gelebilmektedir. Ani virajlarda sürücü karşılaşacağı riske hazırlıklı olamamaktadır. Bunun sebebi Şekil 1.2’de de görüldüğü üzere araç farlarının, aracın doğrultusunda bir aydınlık alan oluşturmasıdır. Araç doğrultusunda olan bir aydınlatma, virajı alırken karanlık bir bölgenin oluşmasına sebep olmaktadır. Bu karanlık bölge de sürüş güvenliğini tehdit ederek kazalara sebebiyet vermektedir (Shreyas vd. 2014).



Şekil 1.2 Geçici körlüğe sebep olan ışık hüzmelerinin yansıması (İnt. Kyn. 4).

2. LİTERATÜR BİLGİLERİ

Karayolu araçlarındaki sistem gelişmeleri ve sürüş esnasındaki iyileştirmeler daha güvenli bir karayolu yolculuğuna imkan sağlamaktadır. Bu sebeple araçlara, araç manevrası için daha gelişmiş yetenekler sağlanması gerektiği düşünülmüştür. Ancak yine de yapılan çalışmada sürücülerin yardımı ile manevraların yapılması gerektiğinin altı çizilmiştir. Kher vd. (2000) nin çalışmasında aracın far şiddeti üzerine bir tasarlama yapılmıştır. Yapılan tasarımda farklı yol ve hava koşullarında optimum şekilde aydınlatma sağlanması üzerinde durulmuş ve test edilmiştir. Test verileri istenen yoğunluk oranın elde edildiğini göstermiştir (Kher vd. 2000).

Connell vd. (2011) tarafından yapılan bir diğer çalışmada araç yönünde ileri doğru bakan bir video kamera aracılığıyla, kameraya gelen girdi verisine bağlı olarak aracın far sistemini kontrol eden bir tasarım yapılmıştır. Bu girdi verisine göre aracın farları uzun ya da kısa olarak ayarlanıp kontrol edilebilmektedir. Yapılan sistem bilgisayarla görme tekniklerine dayalı olarak gerçekleştirilmiştir. Ayrıca daha yüksek performans sağlamak adına otomatik denetimli öğrenme metotları da kullanılmıştır. Ayrıca sistemin farklı hava koşullarındaki performansı da test edilmiştir.

Bir diğer çalışmada gelişmiş bir far sistemi için yeni bir görüş sunulmuştur. Hava durumuna ve yol geometrisine bağlı olarak sürüş güvenliğinin bu çalışmada önemi vurgulanarak göz kamaşmasına ve aydınlatma aralığına bağlı olarak bir optimal far sistemi üzerine çalışılmıştır. Bu far sistemi kullanıcıların gözlerini kamaştırmadan yolun yeterli bir şekilde aydınlatılmasını sağlamak amacıyla tasarlanmıştır. Temelinde araçlardan ve çevreden gelen çok sayıda parametreye bağlı olarak bir aydınlatma yapılması planlanmıştır. Ayrıca çalışmada; tüm kullanıcılar ve sürücüler tarafından oluşturulacak sisteme ek bilgiler verilmeden mükemmel derecede bir yol aydınlatmasının otomatik ve akıllı far sistemleri hala zor bir iş olduğunun altı çizilmiştir (Büthorn vd. 2013).

De (2014) tarafından yapılan bir diğer çalışmada ise evrensel bir adaptif far sistemi tasarlanmıştır. Bu far sistemi, gece saatlerinde ve keskin virajlarda verimli bir sürüş

sağlamak ve sürüş güvenliğini artırmak amacıyla meydana gelmiştir. Tasarlanan sistem çalışmada ekonomik, basit ve güvenilir olarak açıklanmıştır. Bu sistem direksiyona takılı bir sensör aracılığıyla çalışacak şekilde tasarlanmıştır. Direksiyona takılı sensörün açısal olarak dönüşüne bağlı olarak yolun devamlı olarak aydınlatılmasına imkan sağlamaktadır. Özellikle bu sistemin keskin virajlı yollarda direksiyonun açısal hareketine bağlı olarak aydınlatmada önemli avantajlara sahip olduğu ifade edilmiştir. Ayrıca bu far sistemi araçlarda tampona kolayca eklenebilecek şekilde yapılmıştır. Bu şekilde yapılmasının amacı birçok araca kolayca monte olabilmelerini sağlamak olmuştur. Kullanımı ekonomik hale getirmek amacıyla çalışmada basit mantık kullanan bir devreden yararlanılmıştır. Daha gelişmiş şekilde yapılabilmesi için mikrodenetleyiciler ve özel dönüş açısı sensörlerinden yararlanabilmektedir. Bu sistem sürüş durumuna göre yolun uygun şekilde aydınlatılmasını sağlamak amacıyla tüm araçlarda aksesuar olarak da kullanılabilir. Ayrıca böyle bir sistemden yararlanılması sürücüyü de sürüş esnasında önemli derecede yardım sağlamaktadır. Fakat önemli avantajlarının yanı sıra bu sistemin çeşitli dezavantajları bulunmaktadır. LED farlar özellikle tampona uygun şekilde monte edilmesi önem arz etmektedir. Çünkü titreşim veya çeşitli etkiler nedeniyle yerinden çıkması mümkündür. Ayrıca devre için sürekli bir güç beslemesine ihtiyaç duyulmaktadır. Bunun yanı sıra potansiyometrenin açı sensörü olarak kullanılmasından dolayı zamanla eğilim gerçekleşecektir ve bu durum güvenilir değildir. Mevcut prototipi tüm araçlarda kullanabilmek istenirse tüm hidrolik direksiyon tabanlı araçlarla uyumlu hale getirilmesi için değiştirilmesi gerekmektedir.

Magar (2015) ise karayolu güvenliğine atıf yaparak giderek öneminin arttığını belirtmiştir. Güvenliği sağlayabilmek amacıyla diğer sürücülerini rahatsız etmeden otomatik şekilde ayarlanabilir ve yolu mümkün olduğunca fazla şekilde aydınlatmak amacıyla bir otomatik çalışan far sistemi üzerine bir sistem çalışması yapmıştır. Bu sistem yol güvenliğini sağlamak için araca uyarlanabilir bir ön aydınlatma sistemidir. Virajlı yollarda gece yol aydınlatmasını sağlamak amacıyla ön aydınlatma sisteminin bir prototipinin oluşturulması üzerine çalışmıştır.

Farları yatay şekilde ayarlamak için direksiyon simidini kullanan geleneksel yöntemden farklı olarak farları hem yatay hem de dikey olarak ayarlayacak şekilde bir sistem

oluşturulmuştur. Bu sistem aracın önündeki virajı ve ayrıntıları algılamak ve yakalamak için görüntü sensörü olarak kamera kullanmaktadır. Yaklaşan aracı algılaması içinse ultrasonik bir sistemden yararlanılmıştır. Çalışmada virajlı yollar için yol aydınlatmasının daha iyi sağlanmasına yönelik olacak şekilde bir aydınlatma elde edilmiştir.

Chen ve Chiu (2018) ise yeni bir far kontrol sistemi ile birlikte sürüş esnasında çeşitli görevleri yerine getirmek üzere yeni bir yöntem sunmuşlardır. Önerilen far kontrol sistemi uygulamaları olarak bir araca entegre edilmiştir. Entegre edilen araçta aydınlatmanın iz düşüm açısı tespit edilmiştir. Ayrıca far kontrol sisteminin daha başarılı şekilde çalışmasını sağlamak amacıyla V_{SS} çıkış frekansı ile birlikte aracın eğim açısı verileri de incelenmiştir. V_{SS} çıkış frekans değerinin araç hızındaki artışla birlikte arttığı tespit edilmiştir. Farın kontrol edilmesi için tasarlanan elektronik tasarım, aracın eğim açısına bağlı olarak yol durumunu takip etmektedir. Aracın yokuş yukarı durumunda araç hızının artmasına bağlı olarak far yansıtma açısı azalış göstermiştir. Yokuş aşağı yol durumlarında ise, far açısının sabit bir değere sahip olması nedeniyle sürücü için çok daha iyi bir aydınlatma sağlanmıştır. V_{SS} ve elektronik olarak araç durumu izlendiğinde; bu açı, bir far kontrol sistemi tarafından değiştirilebilir olmuştur. Bu çalışmada farklı yol koşullarına bağlı olarak kullanılan araç için far kontrol sisteminin uygulanabileceği sonucuna ulaşılmıştır. Kontrol ünitesi, farklı çalışma gereksinimleri ve fonksiyonel çeşitli gereksinimler altında parametrelerin ayarlanmasına imkan sağlayan bir kontrol kutusu ve elektronik sistemden meydana gelmiştir. Bu ise sistemin birçok araçta kullanılabilir olmasına imkan sağlamıştır. Ayrıca bu tasarlanan sistem bir aracın yanından başka bir araç geçerken; sürücünün, diğer aracın yan tarafına çarpmasına da engel olarak trafik kazalarının önlenmesine yardım etmektedir.

Gowrishankar vd. (2019), ölümcül trafik kazalarının genellikle gece saatlerinde meydana geldiğini vurgulamışlardır. Bunun sebebini ise sürüş esnasında, sürücünün bulunduğu araca yaklaşan nesnelerin akşam saatlerinde genellikle daha geç şekilde ayırt edilmesi olarak açıklanmıştır. Bu noktada gece saatlerinde oluşacak kazaları önlemek amacıyla ön far sistemine yoğunlaşılması gerektiğini savunmuşlardır. Sürücüye yaklaşan nesnelerin algılanabilirliğini iyileştirmek amacıyla bir çalışma yapılmıştır.

Çalışmada ön sis farını yükseltmek ve yönlendirilebilir bir sis farı tasarlamak için model oluşturulmuştur. Çünkü araçlarda sis farının temel amacı; kötü ışık ve mevsim şartlarında güvenliği en yüksek düzeyde tutmaktır. Fakat ışık kaynaklarındaki tüm ilerlemelere ve trafik yoğunluğunun da az olmasına rağmen, kazaların büyük çoğunluğu ne yazık ki gece saatlerinde meydana gelmektedir. Yapılan çalışma farlara eğim verilmesi şeklinde gerçekleştirilmiştir. Farı döndürebilmek ve veriyi alabilmek için araç tekerleklerine konum algılayıcı elektronik sensörler yerleştirilmiştir. Farlara merkezden 16 derece çevrilmeye olanak sağlayacak şekilde bir hareket aralığı sağlanmıştır. Dolayısıyla bu yapılan çalışma sıradan ve basit olan ön far sistemini daha iyi bir sistemle değiştirmeyi önermiştir. Örneğin araç 60km/h hızla yoluna devam ettiğinde uzun farlar yanacaktır ve ters yönden araç geldiğinde açısı aşağı doğru dikey olarak hareket edecektir. Ardından orta konuma tekrar dönecektir. Ayrıca çalışmada kesinliği ve reaksiyon süresini de iyileştirmek için farklı testler de uygulanmıştır.

Poornima ve Harih (2019), araç farklarını kullanırken gece saatlerinde uzun farları kullanmanın daha fazla tehlikeli olduğunu ifade etmişlerdir. Uzun farlar kısa süreliğine de olsa sürücüde ters yönde bir parlama etkisi meydana getirmektedir. Bu durum ise kazalara sebep olmaktadır. Ayrıca gece boyunca sokak lambaları da devamlı olarak yandığından dolayı bir elektrik israfı da söz konusu olmaktadır. Bu ise gelecek nesilleri de etkileyen önemli bir durumdur. Çalışmada akıllı enerji yönetim sistemi ile araç far otomasyonu üzerine çalışılarak her iki sorunu da yok etmeye dayanan bir sistem tasarlanmıştır. Bu sistemde karşıdan gelen aracın uzun farlarını algılayabilmek için bir LDR sensörü kullanılmıştır. LDR sensörü uzun farın ayarını kontrol etmektedir ve Arduino UNO'ya girdi sağlamaktadır. Bu teknikte her iki sürücünün de en az düzeyde farlardan etkilenmesi amaçlanmıştır. Elektrik direklerine akıllı enerji yönetim sistemlerinin yerleştirilmesiyle birlikte elektrik enerjisi tüketimlerini kontrol ederek insan gücünü ise en aza indirmek hedeflenmiştir. PIR sensörü ile yapılmış olan akıllı enerji yönetim sistemi, aracın veya sokaktaki herhangi bir nesnenin hareketini algılayarak yalnızca araç veya nesnenin yakınındaki sokak lambasının yanmasını sağlamaktadır. Ayrıca araç ya da nesne hareket ettikçe lambayı kapatarak enerji tasarrufunu da maksimum düzeyde sağlamaktadır. Güneş doğar doğmaz ise ışık LDR tarafından algılanacak ve otomatik olarak sokak lambaları kapanacaktır.

Shashikant ve Mechkul (2019), uyarlanabilir bir ön aydınlatma sistemi geliştirmişlerdir. Bu sistem bir aracın ışık dağılımını otomatik olarak düzenleyen bir sistemdir. Farklı sürüş şartlarında, örneğin; virajlı yollar ve karşıdan gelen araç gibi durumlarda çalışan özel bir kontrol algoritması geliştirilmiştir. Sistem, direksiyon simidi açısı ve gelen araç ile söz konusu araç arasındaki mesafe gibi parametreler yardımıyla çalışmaktadır. Farın yatay hareketi bu çalışmada direksiyon milinin hareketiyle belirlenmektedir. Farın dikey yöndeki hareketi ise iki araç arasındaki mesafeden elde edilmektedir. Sistemin görüntü işleme algoritmaları aracılığıyla geliştirilebileceği ve her yönden gelen aracı tarayabilmek için radar tipi bir mekanizmanın da çalışmaya eklenebileceği belirtilmiştir. Bu eklemeler ile hem komşu hem de arka taraftaki bir aracın da takip edilebileceği eklenmiştir. Ayrıca yollardaki ışık dağılımını tespit edebilmek amacıyla küresel uydu konumlama sistemlerinden de faydalanılabileceği belirtilmiştir.

Subhi vd. (2019), akıllı bir araç far kontrol sistemi üzerine çalışmıştır. Bu çalışmada araç farlarını çevredeki aydınlatma durumlarına bağlı olarak otomatik şekilde kontrol etmek için bir Arduino kullanılmıştır. Ayrıca ışığa bağımlı bir sensör ile birlikte ultrasonik sensör de kullanılmıştır. Işığa bağımlı sensör çevrenin aydınlatma durumunu algılamak için kullanılmıştır. Ultrasonik sensör ise karşı yönden gelen aracı algılamak için kullanılmıştır. Bu teknoloji sadece araç farlarında değil aynı zamanda sokak aydınlatma sistemlerinde, ev aydınlatma sistemlerinde ve çok daha karmaşık sistemlere entegre edilebilir. Kablosuz şekilde kontrol edilebilir.

Kablosuz GSM/GUI tarafından kontrol edilen aydınlatma sistemleri, ağlar, analog devrelere ve zamanlayıcılara bağlı olarak ışığa bağlı direnç (LDR) ile ilgili birçok araştırma makalesi bulunmaktadır. Bu projede hedef, çevredeki aydınlatma yoğunluk seviyelerinden etkilenen ışığa bağlı dirençten ve karşıdan gelen bir aracın mevcudiyetini algılamak için ultrasonik sensörden veri alarak ve bu verileri Arduino ile işleyerek gelişmiş bir araç far kontrol sistemi uygulamaktır. Bu nedenle, çevredeki aydınlatma yoğunluğuna ve karşı yönden gelen bir aracın mevcudiyetine bağlı olarak, sistem otomatik olarak araç farlarının yoğunluğunu kontrol edecek ve gece boyunca uygun bir araç farları yoğunluk seviyesi ayarlayacaktır.

Son alıřmalar meydana gelen kazaların daha ok saat 18:00 ile 06:00 arasında olduėunu gstermektedir (Balaji 2020). Bu saat aralıėında oluřan kazaların en byk nedenlerinden biri srclerin karřı taraftan gelen araların farlarının geici krlklere neden olmasıdır. Ayrıca yaėmurlu ve yoėun sisli hava kořullarında kaza olasılıėı, normal hava kořullarına gre olduka yksektir. Bu hava kořullarında genellikle sis farlarından yararlanılmaktadır. Gnmzde lks otomobillerde kullanılan nemli teknolojiler bulunmaktadır. Ancak bu teknolojilerin daha ekonomik aralara aktarılması olduka pahalı bir durumdur. Balaji (2020) alıřmasında kazaların yoėun olarak gerekleřtiėi saatlerde ve kt hava kořullarında kazaları azaltabilmek amacıyla far sistemi iin yeni bir teknoloji nermiřtir. alıřmadaki temel ama ekonomik ve otomobil endstrisi tarafından kullanılabilecek bir far sistemi oluřturmaktır. Ayrıca birok arařtırmacı ve otomobil endstrisi tarafından da fazlaca dikkate alınmayan iki ve  tekerlekli aralar iin de bir far sistemi tasarlanmıřtır (Balaji 2020).

Yapmıř olduėumuz alıřmada ise kamera destekli makine ėrenimi algoritmaları ile srcnn gece srř sırasında yol grřn arttırmak iin dinamik bir far modeli tasarlanmıřtır. Bu tasarımda makine ėrenimi ile programlanmış grnt iřleme destekli, aydınlatma alanının saėlaması, karřıdan gelen araca gre uzun fardaki LED hcrelerinin dinamik olarak ayarlanması, trafik iřareti tanıma sistemi ve řerit takip sistemi ile far pozisyonlarının otomatik ayarlanması konularını ele alınmıřtır. Bu doėrultuda aydınlatmadan kaynaklı kaza riskine zm sunacak yeni bir dinamik far modeli sunulmuř ve analizleri yapılmıřtır.

3. MATERYAL ve METOT

3.1 Yazılım ve Donanım Karakteristikleri

Kullanılan programlama dili Python 3.7, gerçek zamanlı görüntü işleme için OpenCV, nesne tespiti yapmak ve data setleriyle eğitilmiş bir model oluşturmak için Keras kütüphanesi kullanılmıştır. Görüntü işleme sistemi, 2.20 GHz 8 çekirdekli Intel Core i7-2670QM CPU'dur. Python yazılımı Pycharm IDE üzerinden derlenmiştir. Araç, trafik işareti, şerit tespiti için gerçek zamanlı görüntü USB girişli bir kamerayla sağlanmıştır. Uzun hüzmeli ve kısa hüzmeli far için iki adet 8x4 LED modül, adaptif farların hareketi için servo motorlar kullanılmıştır. Servo motorlara hareket tetiğini vermek için yatay eksen de açı sensörü, dikey eksen de eğim sensörü kullanılmıştır. Prototip için Arduino kullanıldığından dolayı sensörler endüstriyel değil Arduino uyumlu sensörlerdir. LED modülünü Arduino ile kontrol edilebilmek için "FastLED.h" kütüphanesinden yararlanılmıştır.

3.1.1 Arduino Uno

Uno, temel olarak AVR mikrodenetleyici Atmega328'e dayanan açık kaynaklı bir elektronik platformu olan Arduino.cc tarafından geliştirilmiş bir mikrodenetleyici kartıdır. R3 ise, Arduino Uno'nun üçüncü ve en son revizyonudur. ATmega328P ile birlikte mikrodenetleyiciyi desteklemek için kristal osilatör, seri iletişim, voltaj regülatörü vb. diğer bileşenlerden oluşmaktadır. Arduino Uno, harici elektronik devrelere bağlanmak için kullanılan USB arabirimine, 6 analog giriş pimine, 14 I/O dijital bağlantı noktasına sahiptir. 14 I/O portundan 6 pin PWM çıkışı için kullanılabilir. Atmega328 mikrodenetleyici, zamanlayıcılar, sayaçlar, kesmeler, PWM, CPU, G/C pimleri gibi bir dizi özellikle birlikte gelen ve daha fazla frekans üretmeye yardımcı olan 16 MHz'lik bir frekansa dayanan kart üzerine yerleştirilmiştir. 6 analog pin, A0 ile A5 olarak ayarlanmıştır ve 10 bit çözünürlük sağlayabilmektedir. Bu pinler 0 ile 5V arasında ölçüm yapabilmekte, ancak analogReference() işlevi ve AREF pini kullanılarak yüksek aralığa ayarlanabilirler. Panoda tüm kartı sıfırlayan ve çalışan programı başlangıç aşamasına alan reset pini mevcuttur. Bu pin, kart çalışan programın

ortasında kilitlendiğinde kullanışlı olmaktadır; bu pime basmak programdaki her şeyi temizler ve programı en baştan başlatmaktadır. Komut dosyaları sahip olunan hafızanın 13 KB'lık kısmında depolanmaktadır. Arduino Uno'nun teknik özellikleri Çizelge 2.1'de verilmiştir.

Çizelge 3.1 Ardunio Uno kartının özellikleri.

Özellik Adı	Arduino Uno
Çalışma gerilimi	5 V
Önerilen adaptör giriş gerilimi	7-12 V
Adaptör giriş geriliminin sınırı	6-20 V
Dijital pin sayısı	14 (6 pin PWM özelliklidir.)
PWM pin sayısı	6
Analog pin sayısı	6
Giriş çıkış pinleri için akım değeri	20mA
3.3V pin için akım değeri	50mA
Flash bellek hafızası	32 KB
SRAM	2 KB
EEPROM	1 KB
Saat hızı	16 MHz
Uzunluk	68.6 mm
Genişlik	53.4 mm
Ağırlık	25 g

3.1.2 WS2812 8x4 LED Matris

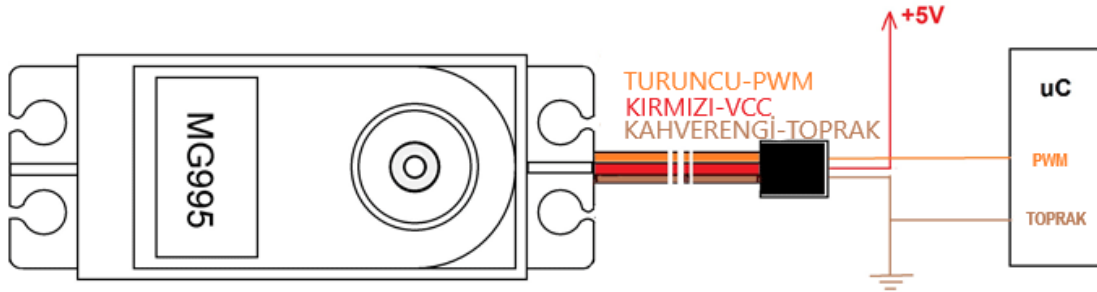
Projemizde kullanılan 32 adet adreslenebilir WS2812 RGB LED'den oluşan kart modülün ön ve arka yüzü Şekil 3.1'de gösterilmiştir. Kartın arkasında 3 giriş ve 3 çıkış lehim alanı bulunmaktadır. Pozitif besleme ve topraklama hattının yanında ve 1 adet data hattı bulunmaktadır. Giriş data sinyali kartın diğer ucundan dışarı çıkmaktadır ve bunun sayesinde birden fazla modül birbirine bağlanabilmektedir. Ayrıca hassas olarak haberleşme yapabilecek yapıya sahiptir ve üstelik toplam 1024 adete kadar LED seri olarak bağlanabilmektedir.



Şekil 3.1 8x4 5050 RGB LED modül kartının arka ve ön tarafı.

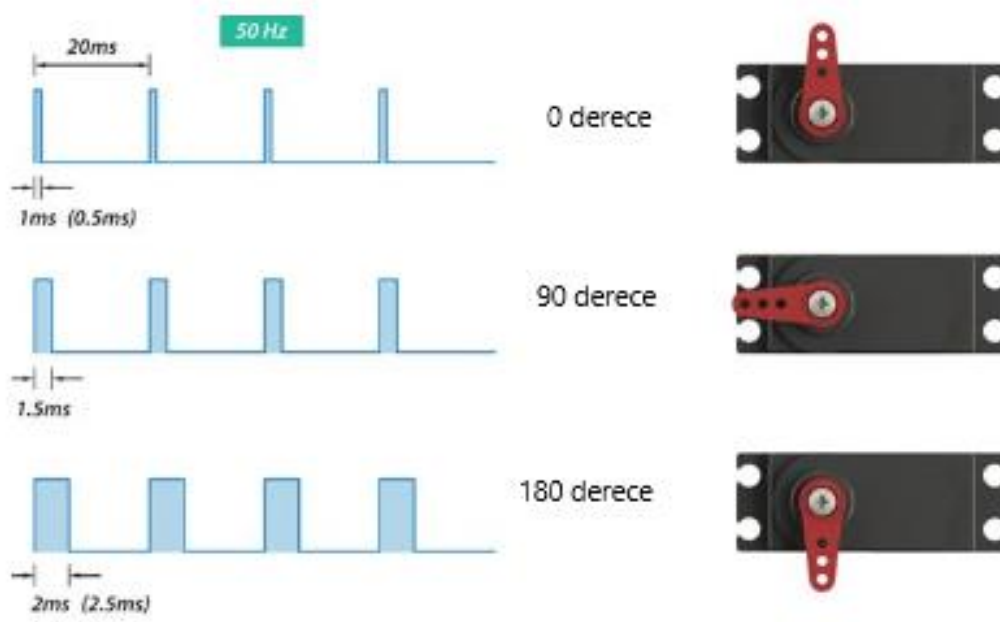
3.1.3 MG995 Servo motor

Bu servo motorlar yaklaşık olarak 120° döndürülebilmektedirler. Çalışma gerilimi 5 voltur. Bu çalışma geriliminde yaklaşık olarak 8,5 kg/cm zorlanma torkuna sahipken 6 V çalışma geriliminde zorlanma tork değeri 10 kg/cm olmaktadır. 60° dönme açısına yaklaşık olarak 0,2 saniyede ulaşmaktadır. Projede kullanılan servo motorun iki adet besleme ucu ve PWM sinyal ucu bulunmaktadır ve bağlantı şekli Şekil 3.2’de gösterilmiştir.



Şekil 3.2 MG995 servo motorun kablo bağlantısı.

Bir servo motor, sinyal hattından bir dizi pals gönderilerek kontrol edilmektedir. Kontrol sinyalinin frekansı 50 Hz olmalı ya da her 20 ms’de bir pals oluşmalıdır. Pals genişliği, servo motorun açısal konumunu belirlemektedir. Genelde bu motorlara 1 ms pals uygulandığında 0° , 1,5 ms pals uygulandığında 90° , 2 milisaniye uygulandığında ise 180° dönmektedir. Şekil 3.3’te servo motorun verilen pals aralığına göre dönme açısı gösterilmektedir. Bu değerler markadan markaya farklılık gösterebilmektedir.



Şekil 3.3 Servo motorun PWM sinyali ile kontrol edilmesi.

3.1.4 Kamera

Görüntü işleme algoritmalarının yürütülebilmesi için görsellerin girdi olarak kullanılması gerekmektedir. Çalışmada prototip olarak Resim 3.1’de gösterilen basit bir webcam kamerası kullanılmıştır. 640x480 piksel maksimum video görüntü çözünürlüğüne ve 1.3 MP sabit görüntü çözünürlüğüne sahip bu olan bu kamera, arabirim olarak USB 2.0 kullanmaktadır ve 5 Volt ile çalışabilmektedir. Bu kameradan elde edilen görüntüler üzerinde görüntü işleme algoritmaları yürütülmüştür.

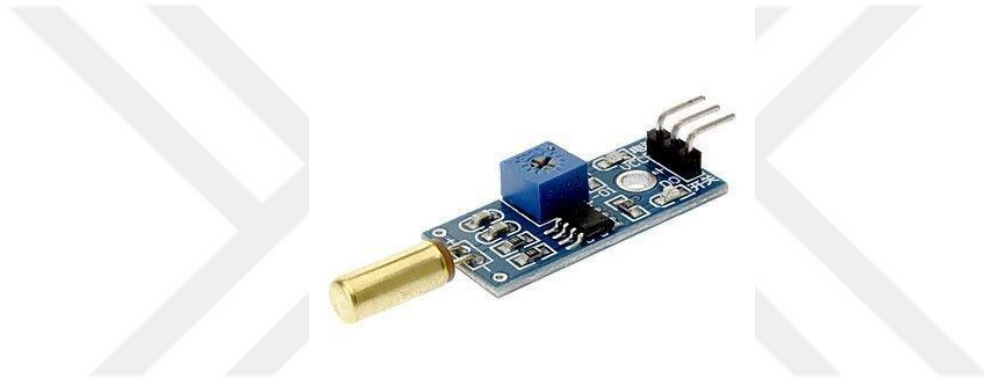


Resim 3.1 Proje prototipinde kullanılan webcam kamerası.

3.1.5 Eğim Sensörü

Eğim Sensörleri aynı zamanda eğimölçerler olarak da bilinmektedir. Bir nesnenin açılarını veya eğimini ölçmek için kullanılan bir tür konum sensörüdür. Eğimölçerler, mevcut en yaygın konum sensörü türlerinden biridir ve birçok endüstride kullanılmaktadır. Eğim sensörlerinin farklı türleri bulunmaktadır. Basit eğim sensörleri, sensör içinde hareket eden iki pimi birbirine bağlayan metalik bir top kullanarak çalışmaktadır. Sensörü "açık" veya "kapalı" konuma getiren devreyi bağlayan sensör eğilirken top konumunu hareket ettirmektedir. Daha karmaşık eğim sensörleri, cihazın yönünü belirlemek için yerçekiminden çekmenin yönünü ölçen dahili bir jiroskop kullanmaktadır. Kullanıldığı alanlara göre farklı teknik özelliklere sahip olmaları gerekmektedir. Eksen sayısı, uygulamadan uygulamaya değişiklik gösterdiğinden eksen sayısı önemli bir faktördür. Video oyunu denetleyicilerinde ve oyun çubuklarında üç eksenli bir eğim sensörü gerekmektedir. Bazı akıllı telefonlarda dört eksenli sensörler kullanılmaktadır. Çözünürlük, sensör tarafından algılanan minimum eğim anlamına gelmektedir ve diğer bir önemli parametredir. Hassasiyet, sensörün küçük değişikliklere tepki verme yeteneğidir. Ölçüm aralığı, Sensör tarafından yönetilebilen eğim aralığını ifade etmektedir. Bazı sensörler 10°'ye kadar ölçüm yapabilirken diğerleri 60°'ye kadar bir aralığı kapsayabilmektedir. Gürültü toleransı, sensörün işlevinde harmonik bozulmalara neden olma eğilimindedir, bu da çıkış değişimine ve sistem etkinliğinin

azalmasına neden olmaktadır. Gürültü düzeyleriyle ilgili üretici yönergelerine uyulması gerekmektedir. Yukarıda bahsedilen tüm bu parametreler seçilecek sensörün önemini ifade etmektedir. Uygulanacak projemiz prototip aşamasında olmasından kaynaklı olarak Arduino'ya kolayca entegre edebileceğimiz bir tilt sensörü kullanılmıştır. Amacımız eğimin yönünü tayin edebilmek ve eğimin tersi yönünde far sistemimizi yaklaşık olarak 3.6° döndürebilmektir. Resim 3.2'de gösterilen basit tipte bir tilt sensörü olan ürün, içindeki serbest haldeki bilye bir uca geldiği zaman iki uç arasındaki iletim sağlayarak eğim olduğu bilgisini vermektedir. Eğim tespit edildiğinde “D0” pininden 5 Volt dijital çıkış vermektedir ve üzerindeki gösterge ışığı yanmaktadır.



Resim 3.2 Proje prototipinde kullanılan eğim sensörü.

3.1.6 Açı Sensörü

Günümüz teknolojisinde hidrolik direksiyonlar yerini yavaş yavaş elektrikli direksiyonlara bırakmaktadır. Elektrikli direksiyonlarda, sürücü direksiyonu çevirmesiyle direksiyon mili de dönmektedir. Bununla birlikte bu mile bağlı olan elektrik motoru da çalışmakta ve direksiyon milinin dönme kuvvetini arttırmaktadır. Böylelikle direksiyon simidi daha yumuşak döndürülebilirken, araç manevrası daha kolay ve hızlı olmaktadır. Projede açı sensörü olarak direksiyon miline bağlı bir potansiyometre kullanılmıştır. Direksiyon döndürme açısı ve yönü aynı anda farların yatay düzlemdeki hareketini tayin etmektedir. Direksiyonun dönme açısının farın dönme açısına oranı 20:1 oranında ayarlanmıştır. Böylelikle farın dönme açısı 15° ile sınırlandırılmıştır. Yani direksiyon 360° derece döndürüldüğünde bile farlar maksimum

15° dönebilmektedirler.

3.2 Tensorflow ile Araç Tanıma

Geceleri yüksek doğruluk oranıyla araçları algılayabilen bu modeli eğitmek için far ve arka lambaların görsellerinin olduğu bir veri seti kullanılmıştır. Bu veri seti geceleri araçları tanımasını sağlamak için 1000’den fazla gece çekilen trafik görüntüsünden oluşmaktadır. İyi bir nesne tanıma sınıflayıcısı yüzlerce görüntüye ihtiyaç duymaktadır. Aynı zamanda görüntü kalitesi de sınıflayıcının iyi öğrenmesi için önemlidir. Tüm görseller, özel bir nesne algılama modeli oluşturabilmek için Şekil 3.1’de görüldüğü gibi LabelImg uygulaması kullanılarak etiketlenmiştir. LabelImg programı her görüntü için etiket verilerine sahip bir “.xml” dosyası kaydeder. Bu “.xml” dosyaları, TensorFlow eğiticisinin girdilerinden biri olan “TFRecords” oluşturmak için kullanılmaktadır. Farlar ve arka lambalar etiketlendikten sonra bu görseller %80 eğitim için %20 test için ayrılmıştır. Araç tanıma sistemini eğitmek için Tensorflow kütüphanesinin bize sağlamış olduğu “ssd_inception_v2_coco” kullanılmıştır.



Şekil 3.4 LabelImg uygulamasından araç farı etiketleme.

Araç algılama modeliyle tespit edilen araç sayısına göre oluşturulan üç karakterli bir

değişken Python yazılımıyla oluşturulmaktadır. Bu değişken ile 2 araçtan fazla araç idare edilememektedir. Eğer tek bir araç tespit edilmişse üç karakterli değişkenin ilk karakteri “x”, ikinci karakteri “y”, üçüncü karakteri ise tespit edilen aracın pozisyonunu tutan değişkendir. İki araç algılanmış ise bu değişkenin ilk karakteri yine “x” olurken ikinci ve üçüncü karakteri tespit edilen araçların konumunu hafızada tutmaktadır. Herhangi bir araç algılanmadığında Arduino’ya “xy0” değişkeni gönderilerek LED farın tüm hücrelerinin aktif olması sağlanmıştır. İkiden fazla araç algılandığında ise “xy9” değişkeni gönderilerek uzun farın tüm LED hücreleri söndürülmektedir. Bu oluşturulan değerler araç farındaki LED’leri kontrol edebilmek için seri port üzerinden Arduino mikrodenetleyicisine gönderilmiştir. LED modülü 4 sıra 16 kolon olmak üzere 64 ayrı çipten oluşmaktadır. Ancak 8 kolonu uzun hüzmeler için kullanılmaktadır ve kalan 8 kolon kısa hüzmeler için hep açık kalmaktadır. Tespit edilen araçların alanı dinamik olarak karanlık tutulurken kalan alanlar uzun hüzmelerle aydınlatılmaktadır. Böylece sürücünün görüş alanı arttırılırken kaza riski de azaltılmıştır. İkiden fazla araç tespit edilirse, mevcut 8 kolon fazla parçaya bölüneceğinden dolayı aydınlatma verimi düşmektedir, bundan dolayı da uzun hüzmeler tamamen kapatılmaktadır.

3.2.1 COCO Veri Seti Modeli

Tensorflow, Google tarafından sunulan ve önceden eğitilmiş modelleri nesne tanımak için bir araçtır. Bu hazır olan modeller COCO veri seti ile eğitilmiş ve modellerin birbirleri arasında hız ve başarı yüzdeleri ortaya çıkmıştır. Burada kendi veri setimizi eğitebilmek için seçeceğimiz model kullandığımız donanımına göre değişiklik gösterebilmektedir. Örneğin, güçlü bir ekran kartına sahipseniz başarı yüzdesi yüksek bir model seçmek mantıklı olacaktır. Aksi durumda ise başarı yüzdesi düşük ama hızı yüksek bir model seçmek yerinde olacaktır. Bu çalışmada gerçek zamanlı nesne tanıma işlemi yapıldığından dolayı donanımına uygun olabildiğince hızlı tespit yapabilen bir model olan `ssd_inception_v2_coco` modeli tercih edilmiştir. CONTEXT'teki Microsoft Ortak Nesneleri (MS COCO) veri kümesi, 82'si 5.000'den fazla etiketli örneğe sahip olmak üzere 91 ortak nesne kategorisi içerir, veri kümesinde toplamda 328.000 görüntüde 2.500.000 etiketli örnek vardır. Popüler ImageNet veri setinin aksine, COCO daha az kategoriye ancak kategori başına daha fazla örneğe sahiptir (Lin vd. 2014).

3.2.2 Arduino ile Uzun Far Kontrolü

Python ile yürütülen algorithmadan gelen üç karakterli değişken seri port üzerinden Arduino mikroişlemcisinde veri olarak işlenmektedir. Eğer tek bir araç tespit edilmişse üç karakterli değişkenin ilk karakteri “x”, ikinci karakteri “y”, üçüncü karakteri ise tespit edilen aracın pozisyonunu tutan değişkendir. İki araç algılanmış ise bu değişkenin ilk karakteri yine “x” olurken ikinci ve üçüncü karakteri tespit edilen araçların konumunu hafızada tutmaktadır. Gelen veriye göre LED kontrolünü sağlamak için FastLED.h kütüphanesine ihtiyaç duyulmaktadır. Görüntü işleme algoritması tarafından hiçbir araç algılanmamış ise Arduino’ya “xy0” değişkeni gönderilecektir. Bu değişken Arduino tarafında tüm LED’lerin aktif olduğu position0() fonksiyonunu çalıştırmaktadır. Sadece bu pozisyona ait Arduino yazılım kodu Şekil.3.5’te verilmiştir.

```
char ch[3] = "";  
  
ch[i] = Serial.read();  
  
if(ch[0] == 'x' && ch[1] == 'y' && ch[2] == '0')  
    position0();  
  
void position0()  
{  
    for(int i = 0; i <= 24; i+=8)  
    {  
        led_matrix[i] = CRGB::White;  
        led_matrix[i+1] = CRGB::White;  
        led_matrix[i+2] = CRGB::White;  
        led_matrix[i+3] = CRGB::White;  
        led_matrix[i+4] = CRGB::White;  
        led_matrix[i+5] = CRGB::White;  
        led_matrix[i+6] = CRGB::White;  
        led_matrix[i+7] = CRGB::White;  
        FastLED.show();  
    }  
}
```

Şekil 3.5 LED pozisyonunu belirleyen örnek algoritma.

İkiden fazla araç algılandığında ise algoritma tarafından “xy9” karakteri gönderilecektir

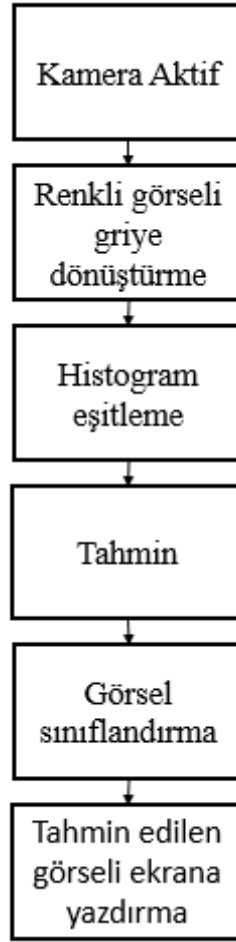
ve bu bilgiyi alan Arduino uzun fara ait tüm LED hücrelerini otomatik olarak söndürecektir. Tek araç ve iki araç algılandığında 8 kolonun 8 farklı pozisyonunu tanımlayan fonksiyonlar çağırılarak karşıdan gelen araca göre LED pozisyonları dinamik olarak değiştirilmektedir.

3.2.3 Arduino ile Far Pozisyonunun Kontrolü

Python ile yürütülen algorithmadan gelen üç karakterli değişken seri port üzerinden Arduino mikroişlemcisinde veri olarak işlenmektedir. Direksiyon miline bağlı olan potansiyometre vasıtasıyla ölçme işlemi yapılmıştır. Direksiyon döndürme açısı ve yönü aynı anda farların yatay düzlemdeki hareketini tayin etmektedir. Direksiyonun dönme açısının farın dönme açısına oranı 24:1 oranında ayarlanmıştır. Böylelikle farın dönme açısı 15° ile sınırlandırılmıştır. Yani direksiyon 360° derece döndürüldüğünde bile farlar maksimum 15° dönebilmektedirler. Şerit çizgilerinin tespiti sonrasında da far pozisyonu ayarlanabilmektedir. Bundan dolayı direksiyona bağlı far pozisyonlama algoritması ancak şerit tespiti tam olarak yapılamadığında devreye girmektedir.

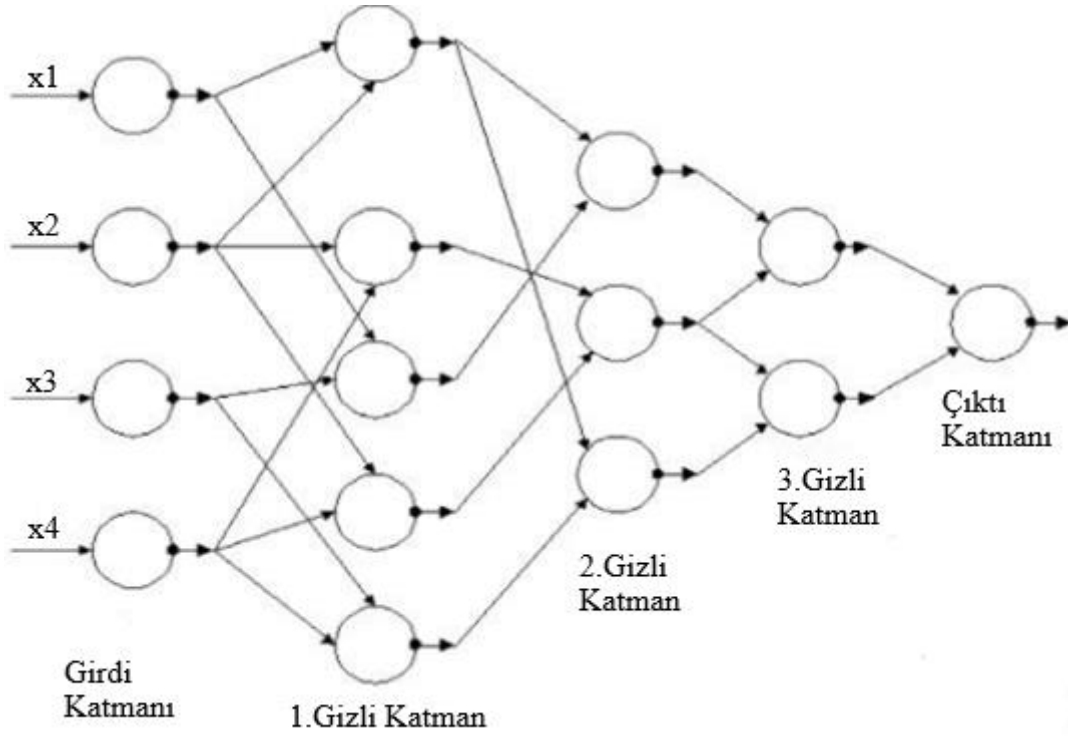
3.3 Trafik İşareti Tanıma

Trafik işaretleri, sürücülere ve diğer yol kullanıcılarına değerli bilgiler sağlamaktadır. Onları ihmal etmek ölümcül olabilmektedir. Uzun yolculuklarda sürücüler çoğu zaman yol güvenlik sinyallerini kaçırmaktadır ve bu işaretleri dikkate almadıkları için birçok kaza meydana gelmektedir. Bu nedenle, derin öğrenme yöntemine dayalı olarak yoldaki trafik işaretlerini gerçek zamanlı olarak tanıyan ve bunları bir ekranda sürücüye gösteren bir model oluşturulmuştur. Bu sistemin akış diyagramı Şekil 3.6'da gösterilmiştir.



Şekil 3.6 Trafik işareti tanıma akış diyagramı.

Bu derin öğrenme modeli yapay sinir ağları kullanılarak inşa edilmiştir. Evrişimsel sinir ağı, derin öğrenme sinir ağlarının bir sınıfıdır. Bu sınıf görüntü tanımda birçok yenilikle kullanıcıya kolaylık sağlamaktadır. Facebook’da fotoğraf etiketleme, otonom araçlar vb. alanlarda hızlı ve verimli olduklarından tercih sebebidir. Keras kütüphanesi Python’da yazılmış bir sinir ağı kütüphanesidir ve bilgisayarlı görme modelleri CNN’i desteklediği için tercih edilmiştir. Bir sinir ağı, gizli katmanlarda eğitim sırasında ayarlanan ağırlık değerleri ile eğitilerek girdi değerleri almaktadır. Eğitim tamamlandıktan sonra modele verilecek yeni girdi değerlerinden tahminler çıkarılmaktadır. Bir derin ağı modeli temel olarak Şekil 3.7’de gösterilmiştir.



Şekil 3.7 Derin ağ mimarisi (İnt. Kyn. 7).

3.3.1 Derin Öğrenme

Derin öğrenme, yapay zekanın alt alanıdır. Kendi kendine veri odaklı kararlar verebilen sinir ağı modelleri oluşturmakla ilgilenmektedir (Goodfellow vd. 2016). Derin öğrenme, sistemlerin deneyimlerden öğrenmesini ve dünyayı kavramlar hiyerarşisi açısından anlamasını sağlamaktadır (Kelleher 2019). Bilgisayar bilgileri toplama konusunda kendi kendine yeteneğe sahip olduğundan, bilgisayar için gerekli olan bilgiyi belirlemek için herhangi bir insan müdahalesi yoktur. Özellik çıkarma, derin öğrenmeyi özel kılan özelliklerden biridir. Özellik çıkarımı, özelliklerin modele sağlanan girdi verilerinden çıkarıldığı bir kavramdır. Bu şekilde, herhangi bir açık kodlama olmaksızın görüntü tanıma gerçekleştirilebilir.

Her görüntü piksellerden oluşur. Bu pikseller, komşu piksellerle ilişkilidir. Derin öğrenme teknikleri, daha büyük bir görüntünün küçük bölümlerinden özellikler çıkararak bu fenomenden yararlanır. Bu tür yöntemler, üst düzey özellikleri öğrenme yeteneğine sahiptir, böylece her yeni problem için bir özellik çıkarıcı geliştirme görevi ortadan kalkmaktadır. Derin öğrenme algoritması, makine öğrenme algoritmalarına göre

birçok parametreye sahip olduğundan, bir derin öğrenme modelini eğitmek genellikle daha fazla zaman almaktadır.

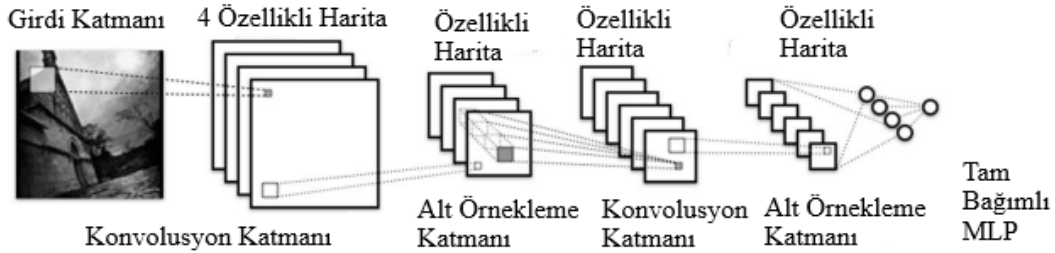
3.3.2 Denetimli Öğrenme

Denetimli öğrenme, makine öğreniminin yaygın dallarından biridir. Denetimli öğrenme, görünmeyen verilerden çıktıları tahmin etmek için kullanılan girdi ve çıktı verileri arasında bir eşleme öğrenmeyi gerektirmektedir (Cunningham vd. 2008). Denetimli öğrenmede model, etiketli verilerle, yani görüntü sınıflandırmasında sınıf etiketli görüntülerle eğitilmektedir. Eğitim aşamasında, özellikler eğitim verilerinden çıkarılmakta ve veri setinin modelini tanımlamaktadır. Test verileri, eğitim aşamasında edinilen bilgilere göre sınıflandırıldığı test aşamasında modele girdi olarak gönderilmektedir. Denetimli öğrenmenin temel amacı, tren verilerini analiz ederek elde edilen ön bilgilere dayalı olarak test verilerinin sınıflandırmasını doğru bir şekilde tahmin etmektir. Yapay sinir ağları, Bayes istatistikleri, analitik öğrenme, denetimli öğrenmede kullanılan birkaç yaklaşımdır (Ang vd. 2015).

3.3.3 Evrişimli Sinir Ağı (CNN) Modeli Oluşturma

CNN, ileri yönlü bir sinir ağıdır ve hayvanların görme merkezinden esinlenerek ortaya atılmış bir yapıdır. Burada uygulanan matematiksel konvolüsyon işlemi, bir nöronun kendi alanından uyarılar verdiği yanıt olarak ifade edilmektedir. CNN bir veya daha fazla konvolüsyonel katmandan oluşabildiği gibi bir veya daha fazla altörnekleme katmanı, standart çok katmanlı sinir ağından oluşabilmektedir (LeCun vd. 2015). Aynı sayıda gizli birimle tamamen bağlı ağlardan daha az sayıda eğitime ve daha az parametreye sahip olmaları CNN'in en önemli kolaylıklarındandır. Örneğin, bir konvolüsyon katmanındaki $a \times a \times b$ boyutundaki bir resim girdi olarak düşünüldüğünde buradaki "a" değerleri sırasıyla görüntünün yüksekliği ve genişliğini ifade ederken b ise kanalların sayısını ifade etmektedir. RGB bir görüntüde b ifadesi 3'e eşit olmaktadır. Konvolüsyonel katmanı ise $c \times c \times d$ boyutunda k adet filtreden oluşmaktadır. Filtre için görüntünün boyutundan daha küçük olan c ve b (kanal) değeriyle genellikle aynı seçilen d değeri seçilmektedir. Bu filtreler ile her biri $a-c+1 \times a-c+1$ boyutunda k adet birbirine yerel olarak bağlı özellik haritaları üretmektedir (Ciresan vd. 2011).

Daha sonra her harita için maksimum veya ortalama havuzlama (pooling) ile $p \times p$ 'lik alt örnekleme işlemi (p değeri, görüntü büyüklüğüne göre genellikle 2-5 arasında) uygulanmaktadır (Scherer vd. 2010). Evrişimli sinir ağı mimarisi Şekil 3.8'de gösterilmiştir.



Şekil 3.8 CNN ağının mimarisi.

Keras kütüphanesinde sıralı ve işlevsel olmak üzere iki tür API bulunmaktadır. Çalışmada sinir ağını oluşturabilmek için sıralı API kullanılmıştır. Bu modelde katmanlar birer birer ve sıralı şekilde oluşturulmuştur bundan dolayı bu modelle çalışmak işlevsele göre daha kolay olmaktadır. Ancak sıralı API dallanan katmanlar için çok esnek değildir çünkü ağa birden fazla giriş çıkışa izin vermemektedir. Bir evrişimli katman, iki boyutlu bir uzayda basitçe evrişim işlemi gerçekleştiren `Conv2D()` sınıfı kullanılarak oluşturulmaktadır.

3.3.3.1 Conv2D Sınıfının Önemli Parametreleri

Filtre sayısı, özellik haritasının derinliği olarak tanımlanmakta ve bir tam sayı değeri almaktadır. Diğer önemli parametre `kernel_size` olarak ifade edilen parametredir. Bu parametre evrişim penceresinin yüksekliğini ve genişliğini ifade etmektedir. Parametre tamsayı değer almaktadır ve (2,2) gibi iki tam sayıdan oluşan demetten oluşmaktadır. Çoğunlukla pencere aynı yükseklik ve genişliğe sahip bir karedir. Üçüncü önemli parametre literatürde `strides` olarak adlandırılan adım kaydırma parametresidir. Adım parametresi matris hesaplaması sırasında filtrenin giriş görüntüsü üzerinde ne kadar kaydırılarak işlem yapılacağını ifade etmektedir. Yükseklik ve genişlik boyunca adım için bir demet ifadesi almaktadır. Genişlik ve aynıysa demet yerine tek değer kullanılmaktadır. Dördüncü parametre “padding” parametresidir. “Padding” parametresi “valid” ve “same” olarak iki opsiyonu vardır. “Padding” ise “valid” olarak ayarlanırsa

“padding” işlemi gerçekleşmemektedir ancak “same” ayarlanırsa “strides=1” olduğundan özellik haritasının boyutunun girişin boyutuyla aynı olacak şekilde sıfırlarla tamamlanmaktadır. Bu parametre giriş matrisi boyutu ile çıkış matrisi boyutunu eşitlemek için kullanılmaktadır.

Beşinci parametre evrişimli katmanda kullanılacak olan etkileştirme işleminin türünü belirleyen aktivasyon parametresidir. Genelde her katmanda ReLU aktivasyon fonksiyonu kullanılmaktadır. Son parametre “input_shape” parametresidir. Bu parametre girişin yüksekliğini, genişliğini ve derinliğini bir tamsayı demeti olarak belirtmektedir. Başka bir deyişle, giriş görüntüsünün boyutudur. Modelde giriş katmanından hemen sonraki ilk katmanda bu argümanın ilk evrişim katmanında belirtilmesi zorunludur.

3.3.3.2 MaxPooling2D Sınıfının Önemli Parametreleri

“Pool_size”, pooling penceresinin boyut parametresidir. Varsayılan değer (2,2)’dir ve tamsayı ya da demet ifadesi alabilmektedir. Tamsayı kullanıldığında her iki boyut için aynı pencere uzunluğu kullanılmaktadır. “Strides”, özellik haritasındaki piksel sayısını ifade etmektedir. Bu değer yükseklik ve genişlik için bir demet ifadesi almaktadır. Yükseklik ve genişlik aynı olduğunda tamsayı ile ifade edilebilmektedir. “Padding”, “valid” ve “same” olarak iki opsiyon almaktadır. “Padding” “valid” olarak ayarlanırsa padding işlemi gerçekleşmemektedir ancak “same” olduğunda özellik haritasının boyutunun girişin boyutuyla aynı olacak şekilde sıfırlarla tamamlanmaktadır.

3.3.3.3 Flatten Sınıfı

CNN’de pooling katmanı ile dense katmanı arasında bulunan bir katmandır. MLP için giriş verilerinin tek bir sütunda tutmak için kullanılmaktadır.

3.3.3.4 Dense Sınıfı

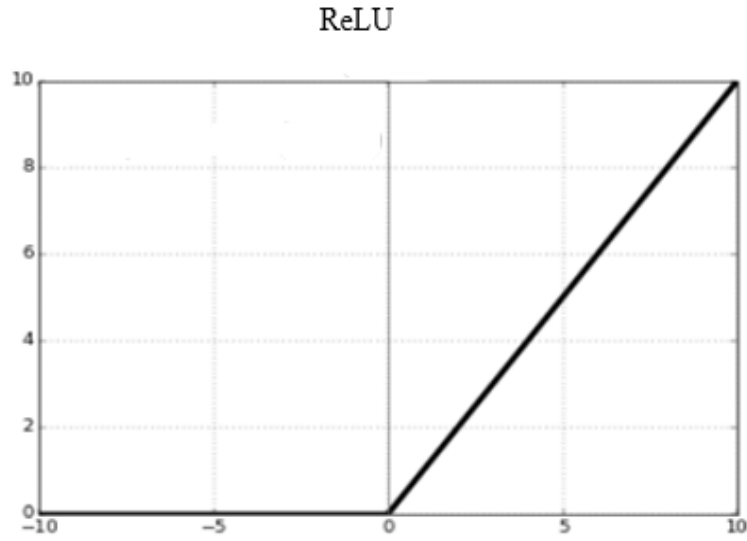
CNN’de son katmanlar tamamen bağlantılı katmanlardır. Keras kütüphanesinde bu katmanlar Dense sınıfı ile oluşturulmaktadır ve Dense katmanı olarak adlandırılmaktadır. Dense sınıfı “units”, “activation”, “input_shape” olmak üzere üç tane parametre almaktadır. “Units”, katmandaki düğümlerin sayısını ifade etmektedir. Gerekli bir bağımsız değişkendir ve tamsayı olmak zorundadır. “Activation”, katmanda kullanılacak aktivasyon fonksiyonunun türünü ifade etmekte ve “None” aktivasyon olmadığı anlamına gelmektedir. “Input_shape”, yoğun katmanlardan herhangi biri bir CNN’deki ilk katman olmadığı için bu parametreyi dahil etmeye gerek yoktur.

3.3.3.5 Aktivasyon Fonksiyonları

Modelimizde ReLU ve Softmax olmak üzere iki farklı aktivasyon fonksiyonu kullanılmıştır. Aktivasyon fonksiyonlarının kullanım sebebi, sinir ağlarında oluşan değerlerin doğrudan döndürülmesini engellemektedir. Oluşan değer doğrudan çıkış sinyali olarak gönderilirse sistem doğrusal bir yapı haline gelmektedir. Ancak birçok durumda yapının doğrusal olmayan pozisyonlarını da öğrenmesi beklenmektedir. İşte bu gibi durumlarda aktivasyon fonksiyonları kullanılması gerekmektedir. ReLU fonksiyonu günümüzde derin öğrenme modellerinde sıkça kullanılmaktadır. ReLU fonksiyonu, aldığı negatif değerlere herhangi bir işlem uygulamadan doğrudan 0 olarak döndürmektedir. Fonksiyon değeri Denklem (3.1)’de verilmiştir.

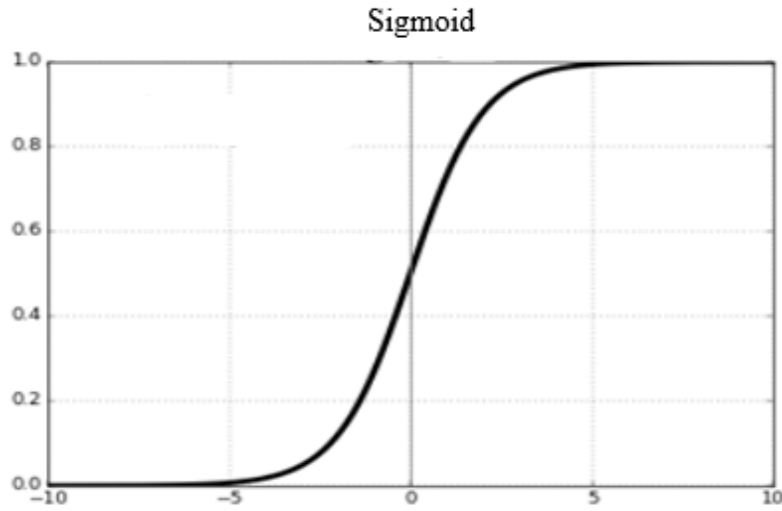
$$R(z) = \max(0, z) \quad (3.1)$$

Bu sebeple negatif değerlerle Şekil 3.9’da gösterilen ReLU aktivasyon fonksiyonunu kullanmak doğru bir tercih olmamaktadır (İnt. Kyn. 5).



Şekil 3.9 ReLU fonksiyon grafiği.

Son katmanda ReLU fonksiyonu yerine Şekil 3.10’da gösterilen Sigmoid fonksiyonu kullanılmıştır (İnt. Kyn. 5).



Şekil 3.10 Sigmoid fonksiyon grafiği.

Sigmoid 0 ile 1 arasında değerler almaktadır, fonksiyon değeri Denklem (3.2)’de son katmana gelen değerler bu fonksiyon ile işleme girmiştir.

$$\sigma(z) = \frac{1}{e^{-z} + 1} \quad (3.2)$$

Fonksiyondaki işlem sonucunda herhangi bir sınıfa ait olma durumu 0 ile 1 arasında

değerler ile elde edilmiştir. Bu durum özellikle sınıflandırma problemleri için çok kullanışlı olmaktadır.

3.3.4 Optimizasyon Algoritması

Yapay sinir ağlarında rastgeleliği azaltmak için optimizasyon algoritmaları kullanılmaktadır. Optimizasyon algoritmaları kullanılması sinir ağlarının niteliklerini değiştirmektedir ve böylelikle kayıplar azaltılmaktadır. Kayıpların azalmasıyla sonucun doğruluğu artmaktadır.

3.3.5 Görüntü İşleme Prosesi

Keras kütüphanesinde sıralı ve işlevsel olmak üzere iki tür API vardır. Oluşturulan CNN modeli Keras kütüphanesine ait load_model komutu ile algoritmaya tanıtılmıştır. Ardından kamera çözünürlük değerleri parametre olarak ayarlanmıştır. Eğitilen modelin tespit edilen görüntüyü tahmin edebilmesi için kameradan alınan gerçek zamanlı görüntüler görüntü işleme proseslerinden geçirilmiştir.

Kameradan gelen görüntü girdi olarak bir diziye dönüştürülür. Oluşturulan bu dizi, eğitim esnasında kullanılan veri setine uygun olacak şekilde OpenCV kütüphanesinin sağlamış olduğu cv2.resize() komutuyla yeniden boyutlandırılmıştır. Görüntüye ait yeniden boyutlandırılan bu dizi OpenCV kütüphanesinin sağlamış olduğu “cv2.cvtColor(boyut,cv2.COLOR_BGRGRAY)” komutu kullanılarak siyah beyaz bir görüntü dizisi elde edilmiştir. Elde edilen siyah beyaz görüntünün kalitesini arttırmak için “cv2.equalizeHist()” komutu uygulanmıştır. Siyah beyaz olarak elde edilen matris incelendiğinde 0 ile 255 arasında değerlerden oluştuğu görülmektedir. Ancak biz görüntünün 0 ile 1 arasındaki değerlerden oluşmasını istediğimizden dolayı matris 255 değerine bölünüp yeniden boyutlandırılarak tahmin algoritmasına gönderilmiştir. Tüm bu görüntü işleme proseslerinden geçen görüntü dizisi Keras kütüphanesinin sağlamış olduğu “load_model.predict()” komutuyla tahmin işlemi gerçekleştirilmiştir.

3.4 Şerit Çizgisi Tespiti

Şerit takip sistemi aracın direksiyon açısını tahmin ederek farların konumunu ayarlamaya yardımcı olmaktadır. Kamera ile donatılmış sistem, Python yazılım dilinin OpenCV kütüphanesi kullanılarak gerçek zamanlı şerit algılama yapabilmektedir. Kullanılan kameradan gerçek zamanlı olarak algoritmaya verilen görüntüler bazı görüntü işleme proseslerinde işlenmiştir ve şerit çizgilerinin doğru bir şekilde tespit edilmesi sağlanmıştır.

3.4.1 Görüntü Filtreleme

Gaussian Blur filtreleme, çekirdek matrisle resim üzerindeki piksellerin çarpılarak görüntünün bulanıklaştırma işlemidir ve 2D Gaussian fonksiyonu Denklem (3.3) ile ifade edilmektedir (Andrei vd. 2022).

$$G_{(x,y)} = \frac{1}{2\pi\sigma^2} \cdot e^{\frac{-(x^2+y^2)}{2\sigma^2}} \quad (3.3)$$

Denklem (3.3)'deki x ve y değerleri, yatay ve dikey yönde merkezden olan mesafelerdir. σ ifadesi Gaussian dağılımının standart sapmasını ifade etmektedir. Görüntü gürültüsünü ve ayrıntıları azaltmak için kullanılmaktadır. Bu bulanıklaştırma tekniğinin görsel etkisi, bir görüntüye yarı saydam bir ekrandan bakmaya benzemektedir. Gauss filtreleri alçak geçiren filtrelerdir. Alçak geçiren filtre olması sebebiyle yüksek frekansları zayıflatmaktadır. Bu özellik de kenar algılama algoritmalarında kullanılma sebebidir.

3.4.2 Canny Kenar Tespiti

Canny kenar algılama algoritması bulanık olan görüntüdeki çapraz, dikey ve yatay kenarları hesaplamak için dört filtre kullanmakta ve kenar yön açısı dikey, yatay ve iki köşegeni (0° , 45° , 90° , 135° derece) gösteren dört açıdan birine yuvarlamaktadır. Görüntü gradyanı Denklem 3.4'teki formülle hesaplanarak gri alan değerlerinde büyük değişiklikler olan kenar piksellerini tanımlanmaktadır. Birinci türevi dikey yönde G_y ve

yatay G_x yönde olarak tahmin etmekle sonuçlanmıştır (Sekehravani vd. 2020, Wolf vd. 2022). Böylelikle gri yoğunluğunun en fazla değiştiği alanlar algoritma ile tespit edilmiştir. G_x ve G_y sırasıyla yatay ve dikey gradyanları temsil etmektedir.

$$G = \sqrt{G_x^2 + G_y^2} \quad (3.4)$$

$$(x, y) = \arctan \frac{G_y}{G_x} \quad (3.5)$$

3.4.3 Görüntüdeki İlgili Alanların Belirlenmesi (ROI)

ROI, görüntüde sadece istenilen piksel alanları üzerinde çalışma yapmak, kuşbakışı bir görünüm demektir. Şerit çizgileri tespit edilmek istendiğinde görüntüde birçok nesnenin tespit edilmesi şerit algılamayı zorlaştırmaktadır. Bundan dolayı araç seyir halindeyken çerçevede belirli bir alanda şerit çizgilerini aramak tespit için daha doğru olmaktadır.

3.4.4 Hough Dönüşümü

Hough dönüşümü matematiksel olarak ifade edilebilen bir şekli tespit etmek için geliştirilmiş bir algoritmadır. Bir doğruyu tanımlamak için algoritmaya ait Denklem (3.6) ve (3.7) ile ifade edilmiştir (Fokkinga 2011). Bu çalışmada doğrusal bir şekil tespiti yapılmıştır.

$$y = mx + c \quad (3.6)$$

Doğrunun orijin ile yaptığı θ açısı ve buna bağlı olarak orijine olan mesafesi doğruyu ifade etmek için yeterlidir.

$$\rho = x \cos \theta + y \sin \theta \quad (3.7)$$

Bu teknik, görüntü üzerindeki tüm noktaların θ ve ρ değerlerini taramaktadır. Böylelikle pikseller üzerinde bu değerler tutuyorsa o piksel üzerinde çizginin var olduğu anlaşılmaktadır.

3.4.5 Şerit Çizgilerinin Tespiti ve Eğrilik Hesabı

Şerit çizgilerinin tam olarak nerden başladığını tespit edebilmek için Histogram çizmek önemlidir. Histogram grafiği incelendiğinde beyaz piksellerin iki farklı tepe oluşturduğu görülmektedir. Sağ ve sol şerit çizgilerinin nerde başlayıp nerde biteceği bu grafikte net olarak anlaşılmaktadır. Histogramın x koordinatı tespit yapılan çerçevenin x koordinatını oluşturmaktadır. Böylelikle görüntüde şerit çizgisi aramak için gerekli olan başlangıç noktaları elde edilmiş olacaktır. Görüntüdeki pikseller siyah-beyaz formata dönüştürüldüğünde oluşan sıcak pikseller ikinci dereceden bir polinom olarak ifade edilmektedir. Görüntüde oluşan sağ ve sol çizgiler neredeyse dikey olduğundan pikseller birden fazla y değeri için aynı x değerine sahip olmaktadır. Bundan dolayı eğrilik hesabı ikinci dereceden polinom katsayıları elde edildikten sonra y'nin 0 ile görüntü yüksekliği arasındaki değerler için Denklem (3.8a) kullanılarak y değerlerine karşılık gelen x değerleri hesaplanmaktadır.

$$x = f(y) = Ay^2 + By + C \quad (3.8a)$$

Fonksiyonumuz Denklem (3.8a) ile ifade edilmişti. Buradan f(y) fonksiyonun 1.türevi ve 2.türevi elde edilmiştir. Bir eğrinin belirli bir noktadaki eğrilik yarıçapı, o noktaya teğet olan dairenin yarıçapına eşittir. Eğrinin yarıçapı Denklem (3.8b) ile ifade edilmektedir (İnt. Kyn. 6).

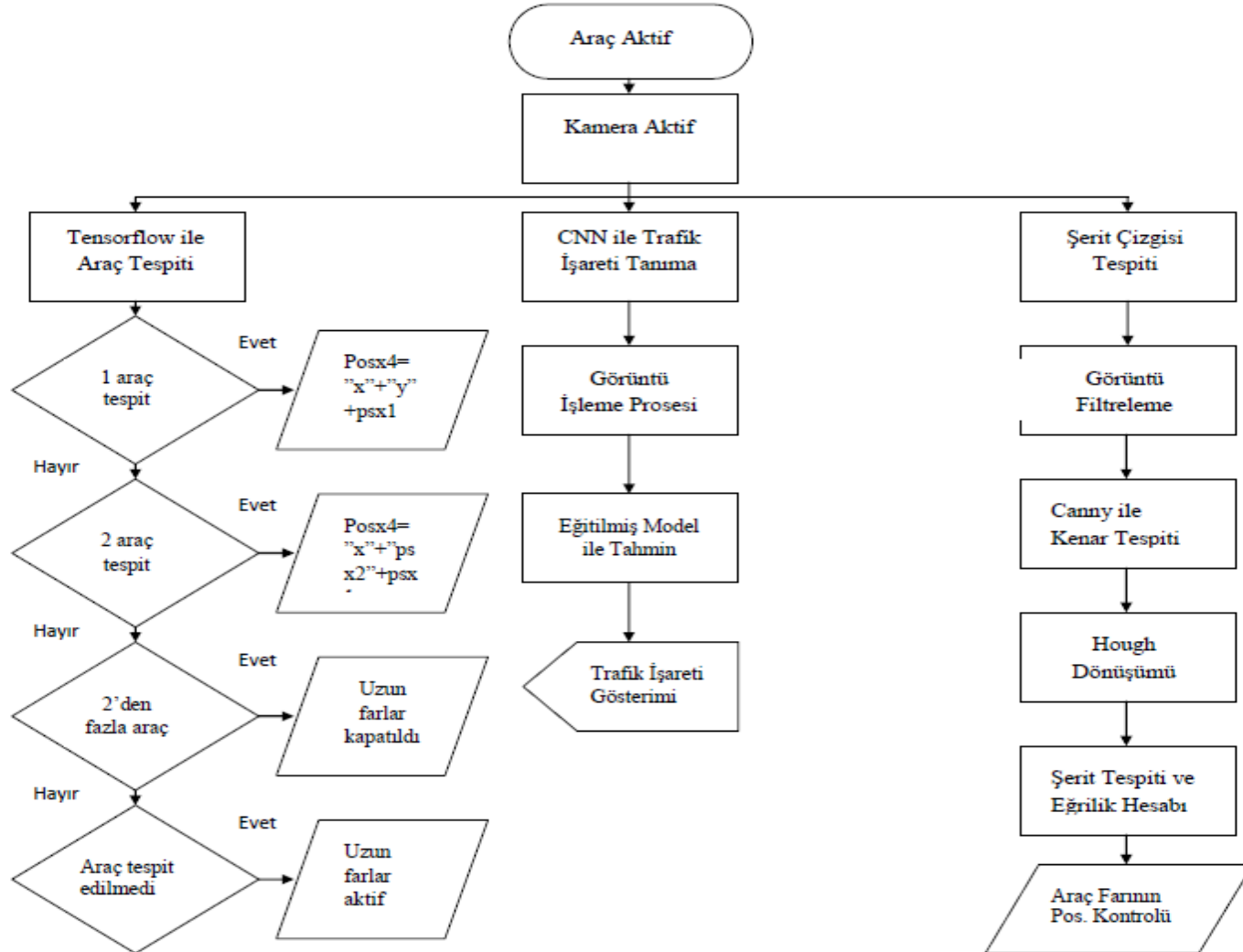
$$Eğrilik\ yarıçapı = \frac{[1 + (\frac{dx}{dy})^2]^{\frac{3}{2}}}{\frac{d^2y}{dx^2}} \quad (3.8b)$$

$$f(y)' = 2Ay + B \quad (3.8c)$$

$$f(y)'' = 2A \quad (3.8d)$$

Denklem (3.8c) ve (3.8d)'deki ifadeler Denklem (3.8b)'de yerine konulduğunda eğrilik yarıçapı formülü olan Denklem (3.8e) elde edilmiştir. Elde edilen bu denklem Şekil 3.11'de akış diyagramı gösterilen algoritmada eğri hesaplama formülü olarak ifade edilmiştir.

$$Eğrilik\ yarıçapı = \frac{[1 + (2Ay + B)^2]^{\frac{3}{2}}}{|2A|} \quad (3.8e)$$



Şekil3.11 Kamera destekli makine algoritmalarına dayalı dinamik far modelinin akış diyagramı.

4. BULGULAR

Bu çalışma, mevcut otomobillerde kullanılan standart far teknolojisine daha sofistike bir yaklaşımı göstermek için yapılmıştır. Zorunlu kısa ve uzun farlara sahip olmanın yanı sıra, düşük ışık koşullarında daha güvenli sürüş ortamı yaratmak için LED'lerin esnekliği kullanılmıştır. Bu teknolojinin temel amacı hem karşıdan gelen hem de öndeki trafikte diğer sürücülerin gözünü kamaştırmadan mümkün olduğunca yolun tamamını aydınlatmaktır ve farlarının pozisyon kontrolü sayesinde topografya fark etmeksizin eğimli ve virajlı yollarda görüş açısı doğrultusunda aydınlatma sağlamaktır. Aynı zamanda sürücünün trafik işaretlerini görmeme ihtimalini ortadan kaldırmak adına makine öğrenimine dayalı olarak trafik işareti algılama teknolojisi sisteme entegre edilmiştir. Sistemde bir web kamerası bulunmaktadır. Bu web kamerası araç algılayabilmek için yolu önceden analiz etmektedir ve temel prensibi farlar ve arka lambaları tespit etmek olan eğitilmiş bir algoritmaya görüntü akışı sağlamaktadır.

Bu modelin geceleri araçları tanımasını sağlamak için 1000'den fazla gece çekilen trafik görüntüsü kullanılmıştır. Nesne algılama modeliyle tespit edilen bu araçların konumları üç karakterli bir değişken olarak Python yazılım diliyle oluşturulmaktadır. LED'leri tek tek harekete geçirebilmek için bu değişken değer seri port üzerinden Arduino mikrodenetleyicisi tarafından alınmaktadır. Bu değişken ile 2 araçtan fazla araç idare edilememektedir. Eğer tek bir araç tespit edilmişse üç karakterli değişkenin ilk karakteri "x", ikinci karakteri "y", üçüncü karakteri ise tespit edilen aracın pozisyonunu tutan değişkendir. İki araç algılanmış ise bu değişkenin ilk karakteri yine "x" olurken ikinci ve üçüncü karakteri tespit edilen araçların konumunu hafızada tutmaktadır. Herhangi bir araç algılanmadığında Arduino'ya "xy0" değişkeni gönderilerek LED farın tüm hücrelerinin aktif olması sağlanmıştır. İki den fazla araç algılandığında ise "xy9" değişkeni gönderilerek uzun farın tüm LED hücreleri söndürülmektedir. İki den fazla araç tespit edilirse, mevcut 8 kolon fazla parçaya bölüneceğinden dolayı aydınlatma verimi düşmektedir, bundan dolayı uzun hüzmeler tamamen kapatılmaktadır. LED modülü 4 sıra 16 kolon olmak üzere 64 ayrı çipten oluşmaktadır. Bu modülün 8 kolonu uzun hüzmeler için kullanılmıştır, kalan 8 kolonu ise kısa hüzmeler için hep açık kalmaktadır. Tespit edilen araçların alanı dinamik olarak karanlık tutulurken kalan alanlar uzun hüzmelerle aydınlatılmaktadır. Böylece sürücünün görüş alanı arttırılırken

kaza riski de azaltılmıştır.

Sürücünün görüş alanını iyileştirmek için sisteme eklenen diğer bir özellik far modülünün yatay konumda pozisyonunun kontrol edilmesidir. Şerit çizgileri eğriliği kadar veya direksiyon simidinin döndürme açısının 24'de 1'i kadar pozisyon kontrolü sağlanabilmektedir. Şerit çizgilerinin tespitinde ve eğrilik hesabında görüntü işleme algoritmaları kullanılmıştır. Direksiyon simidinin dönme açısını hesap etmek için direksiyon miline bağlı bir potansiyometre entegre edilmiştir. Bu potansiyometreden alınan analog değer Arduino üzerinden far sistemindeki servo motorlara aktarılmıştır. Böylelikle gece sürüşlerde virajlardan kaynaklı kör noktalar artık oluşmamıştır ve güvenli bir sürüşe zemin sağlanmıştır.

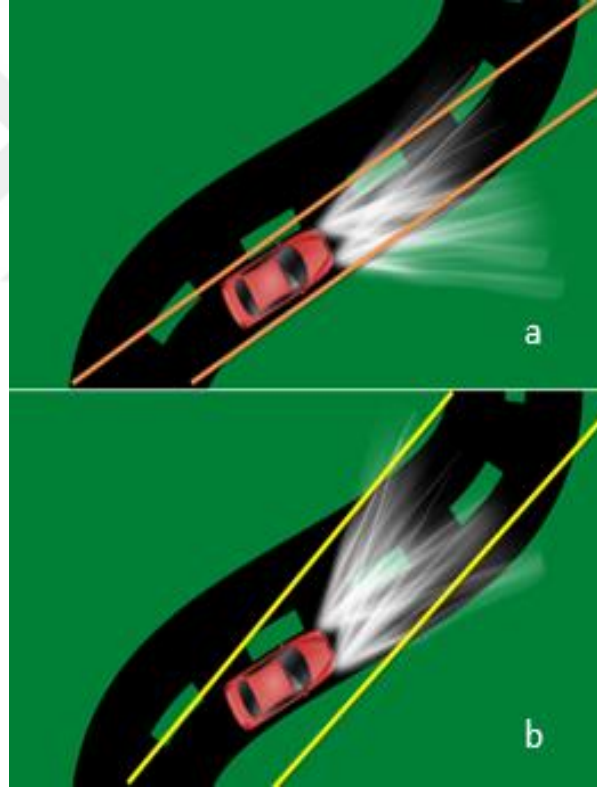
Sistemin diğer bir özelliği ise makine öğrenimine dayalı trafik işareti tanıma sistemine sahip olmasıdır. Gece yolculuklarında sürücüler çoğu zaman yol güvenlik işaretlerini kaçırmaktadırlar ve buna çok fazla dikkat etmemektedirler. Bu nedenle, makine öğrenimine dayalı olarak yoldaki trafik sinyallerini gerçek zamanlı olarak tanıyabilen ve bunları araç ekranında gösterebilen bir model oluşturulmuştur. Trafik işareti tanıma modeli yaklaşık 34800 adet görsel ile eğitilerek oluşturulmuştur. Bu model, Python yazılım dilinin sağlamış olduğu Keras kütüphanesi vasıtasıyla oluşturulmuş bir CNN modelidir. Keras kütüphanesi Python'da yazılmış bir sinir ağı kütüphanesidir ve bu kütüphane bilgisayarlı görme modelleri için CNN'i desteklediği için tercih edilmiştir. Oluşturulan model daha sonra görüntü işleme proseslerinden çıkan görüntünün tahminini yapabilmesi için kullanılmıştır. Bu modele girdi olarak verilen gerçek zamanlı görüntünün bazı görüntü işleme proseslerinden geçerek modelin tahmin yapabileceği standartlara gelmesi sağlanmıştır.

4.1 Adaptif Far Tasarımı

Adaptif far sisteminde araç viraj alırken sensörler araç hızını ve direksiyon açısını göz önünde bulundurup virajın yerleşimine uygun şekilde aydınlatma sağlamaktadır. Aynı zamanda yokuş yukarı çıkıldığında standart bir far daha yüksek bir seviyede aydınlatma sağlamaktadır. Bu da sürücünün karşıdan gelen aracı görmesini zorlaştırmaktadır.

Adaptif sistemde aracın konumuna göre far sistemine aşağı yukarı doğru hareket edebilme kabiliyeti kazandırılmıştır. Adaptif far sistemi yalnızca araç hareket halindeyken aktif edilmektedir. Araç durduğunda, geri vites konumunda olduğunda direksiyon hareket ettirilse bile sistem aktif olmayacak şekilde tasarlanmıştır. Bu şekilde ele alınmasının sebebi karşıdan gelen sürücüleri rahatsız etmemektir.

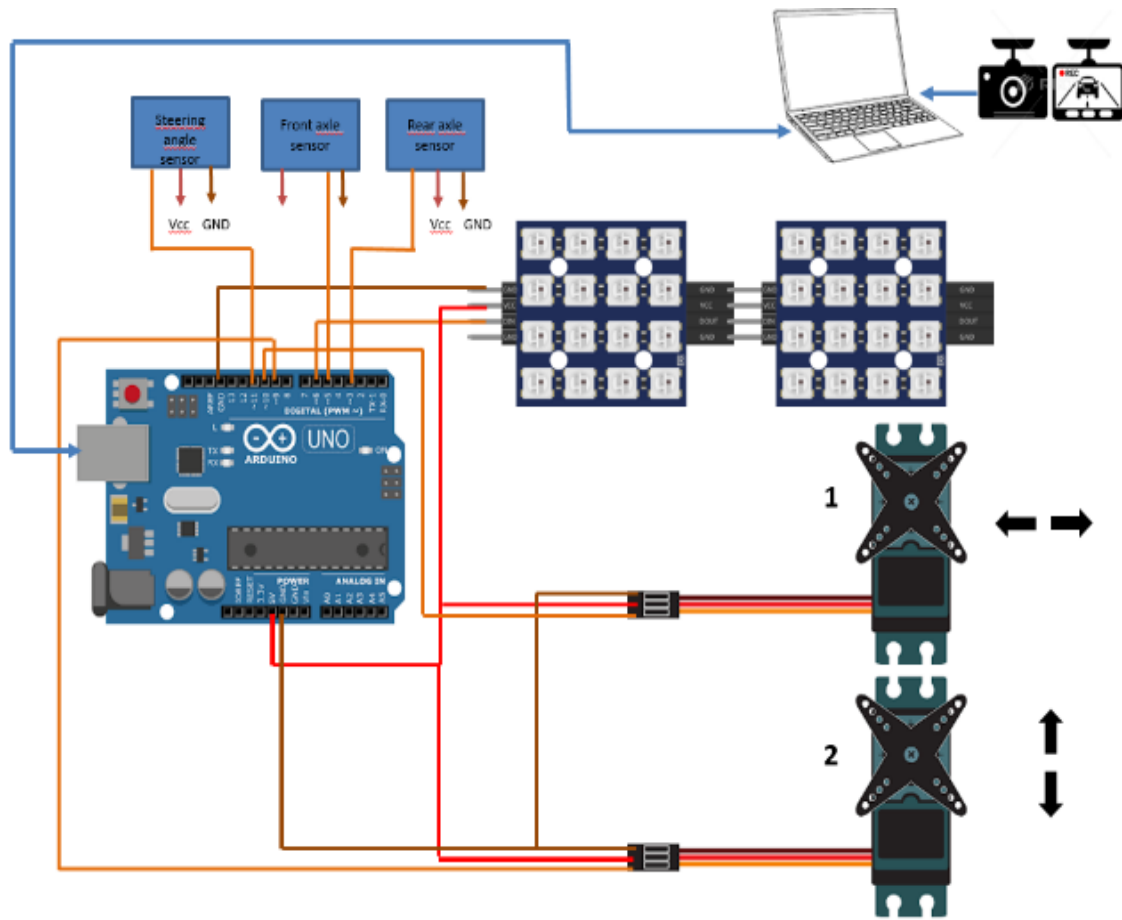
Şekil 4.1a geleneksel far sisteminin aydınlatma alanını göstermekte olup, Şekil 4.2b ise adaptif farın aydınlatma alanını göstermektedir. Görüldüğü üzere geleneksel far sistemi araç doğrultusunda bir aydınlatma sağlarken, adaptif far ise yol doğrultusu boyunca bir aydınlatma alanı sağlamaktadır. Geleneksel far sisteminde aydınlatma alanı yolun eksenine göre olmadığından dolayı ışık hüzmeleri yol ekseninin dışına taşmaktadır.



Şekil 4.1 Geleneksel far aydınlatma alanı (a), adaptif far aydınlatma alanı (b).

Projede tasarlanmış adaptif far sistemi Şekil 4.2’de gösterilmiştir. 1 numaralı servo motor direksiyon açısı sensöründen aldığı değer ile far sistemini sağa sola döndürmektedir. 2 numaralı servo motor, aracın yokuş yukarı ya da yokuş aşağı seyri

sırasında far konumunu aşağı ya da yukarı hareket ettirmektedir. Şekil 4.2’de görüldüğü üzere kameradan araç tanıma algoritmasıyla tespit edilen araç ya da araçların x koordinat bilgileri seri haberleşme protokolü vasıtasıyla Arduino işlemcisine aktarılmıştır. Arduino işlemcisinde tespit edilen araç sayısı eğer 1 veya 2 ise hesaplanan dinamik alan kadar LED söndürülmektedir. Araç yaklaştıkça karartılan bu dinamik alan değişmektedir. Örneğin, ikiden fazla araç tespit edilmişse uzun hüzmeye ait LED modülü tamamen karartılmaktadır. Ek olarak hiçbir araç tespit edilmemişse uzun hüzmeye ait LED modülü tamamen aktif hale getirilmektedir.



Şekil 4.2 Adaptif far donanım sistemi.

4.2 Araç Tespit Uygulaması

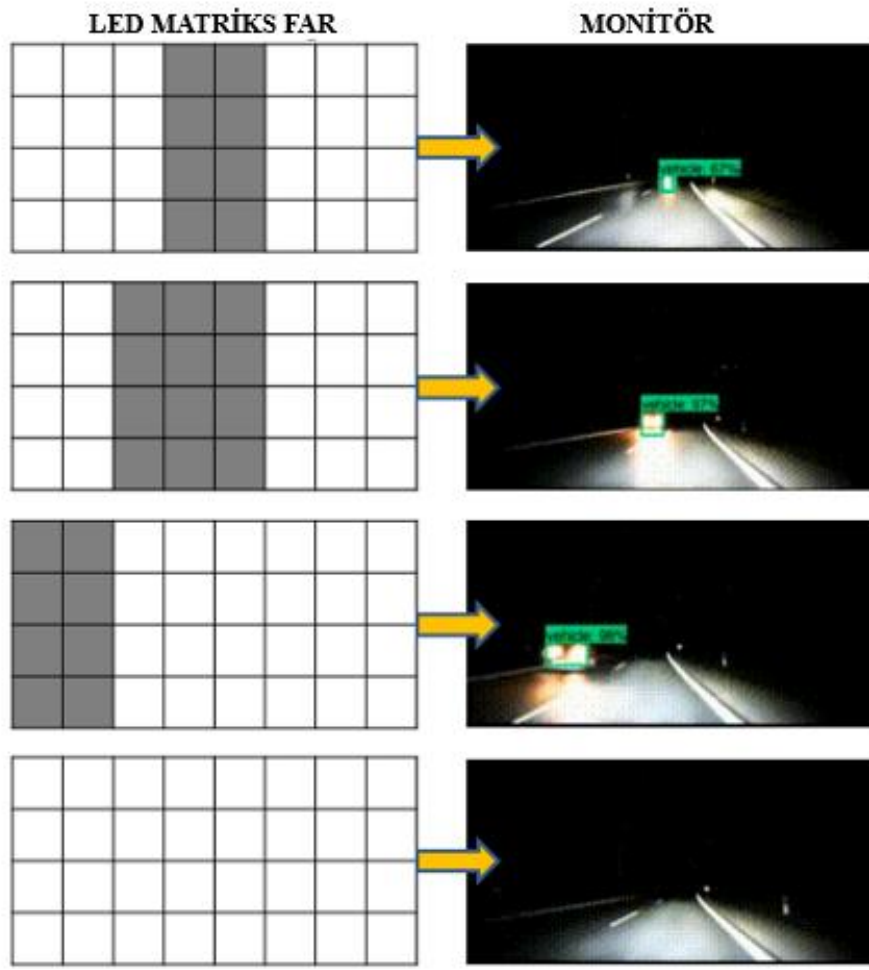
Geceleri yüksek doğruluk oranıyla araçları algılayabilen bu modeli eğitmek için far ve arka lambaların görsellerinin olduğu bir veri seti kullanılmıştır. Bu veri seti geceleri araçları tanımasını sağlamak için 1000'den fazla gece çekilen trafik görüntüsünden oluşmaktadır. Araç tanıma sistemini eğitmek için Tensorflow kütüphanesinin bize sağlamış olduğu “ssd_inception_v2_coco” kullanılmıştır”. Oluşturulan model daha sonra araç tespit algoritmasına entegre edilmiştir. Araçların konumlarını ifade eden 3 karakterli bir değişken Python yazılım diliyle oluşturulmaktadır. Eğer tek araç algılanmışsa ilk değeri “x”, ikinci değeri “y” ve üçüncü değeri aracın konumunu ifade eden 0-9 arasında bir değer olmaktadır. İki araç tespit edilmiş ise ilk değer aynı şekilde “x” olmaktadır, ikinci değer birinci arabanın konumunu ifade eden ve 0-8 arasında bir değer alan bir değişken, üçüncü değer ise ikinci tespit edilen aracın konumunu ifade eden ve 0-8 arasında bir değer alan değişkendir. Hiçbir araç algılanmamışsa bu değişken “xy0” değeri almaktadır. İki den fazla araç tespiti olduğunda değişken “xy9” olmaktadır.

Daha sonra Python yazılımı ile tespit edilen nesnelerin etrafını çevreleyen sınırlayıcı kutunun x koordinatındaki değer kamera çerçeve genişliğiyle çarpılması sonucunda elde edilen minimum ve maksimum değerlerinin ortalaması “inpX1” olarak tanımlanmıştır. Belirlenen çerçeve genişliğinden büyük ya da küçük olmasına göre oluşturulan üç karakterli değişkenin oluşturulma algoritması görülmektedir. Bu değer aslında LED modüldeki çiplerin yanıp sönmeye pozisyonları ayarlayan konum değerlerini hafızasında tutmaktadır.

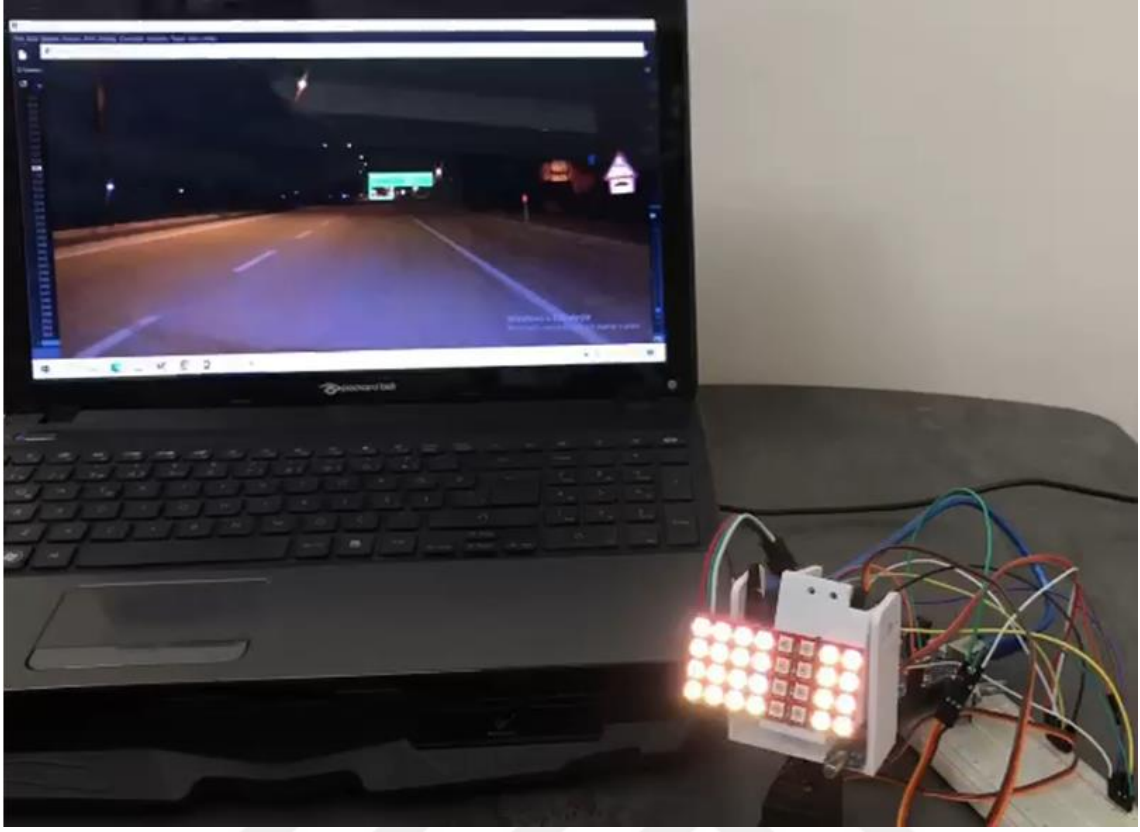
Bu konum değerleri araç farındaki LED’leri yakıp söndürebilmek için seri port üzerinden Arduino mikrodenetleyicisine gönderilmiştir. Arduino’da oluşturulan algoritmaya göre üç karakterden oluşan bu değere göre LED modül üzerindeki kolonlarda aktif olan çipler değişmektedir. Örneğin, Arduino’ya Python yazılımı ile seri port üzerinden gelen “xy0” değeri tüm LED çiplerini aktif etmektedir. Çünkü bu değer Python ile yazılan algorithma hiçbir arabanın karşıdan gelmediğini ifade etmektedir. İki den fazla araç tespit edilmişse Python yazılımı tarafından Arduino’ya “xy9” değişkeni gönderilmektedir. Gönderilen değişken üzerine position9() fonksiyonunu çalıştırarak aktif olan tüm LED’lerin söndürülmesine sağlamıştır.(Ek 1)

LED modülü 2 adet 4 sıra 8 kolon çipten oluşmaktadır. 32 adet çip uzun hüzmeler için diğer 32 adet çip de kısa hüzmeler için kullanılmıştır. Araç veya araçlar tespit edildiğinde bu araçların alanı dinamik olarak karanlık tutulmaktadır ve kalan alanlar uzun hüzmelerle aydınlatılmaktadır. Şekil 4.3'te araç tespit algoritması ikiden fazla araç tespit ettiğinde Arduino'ya uzun hüzmeleri tamamen kapatma bilgisi göndermektedir. Çünkü çok fazla araç tespit edildiğinde alanın birden fazla parçaya bölünmesi verimsiz bir aydınlatma alanı oluşturmaktadır. Şekil 4.4'te ise tek araç algılandığında anlık olarak farın nasıl davrandığı gösterilmiştir.





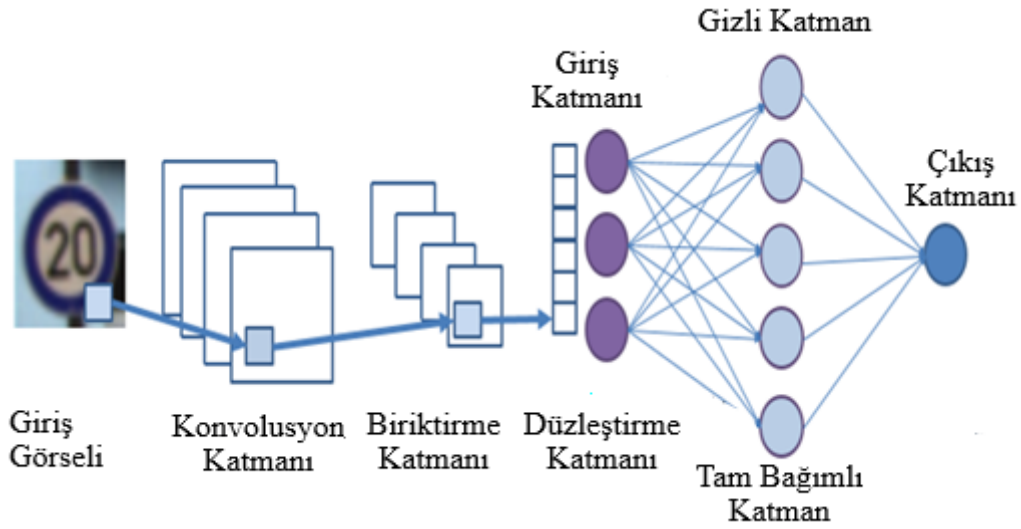
Şekil 4.3 Araç tanıma ve uzun far sistemi.



Şekil 4.4 Araç tanıma algoritmasında farların davranışının görüntülenmesi.

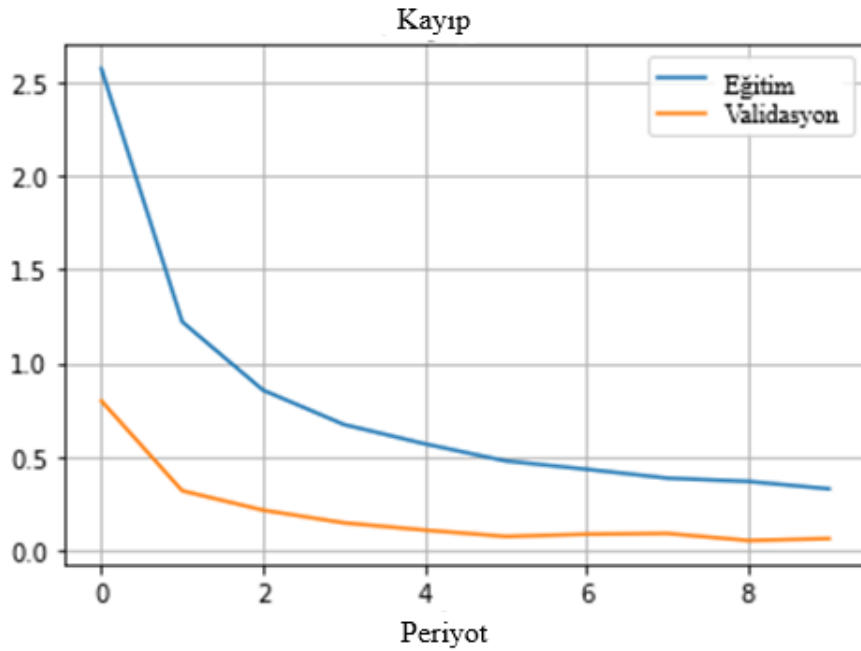
4.3 Trafik İşareti Tanıma Uygulaması

Bu derin öğrenme modelimizde Şekil 4.5'te görüldüğü gibi CNN kullanılmış olup, veri seti olarak 43 adet trafik işaretine ait yaklaşık 34.800 adet görsel girdi olarak kullanılmıştır ve bu veri seti eğitim ve test olarak iki parçaya bölünmüştür.

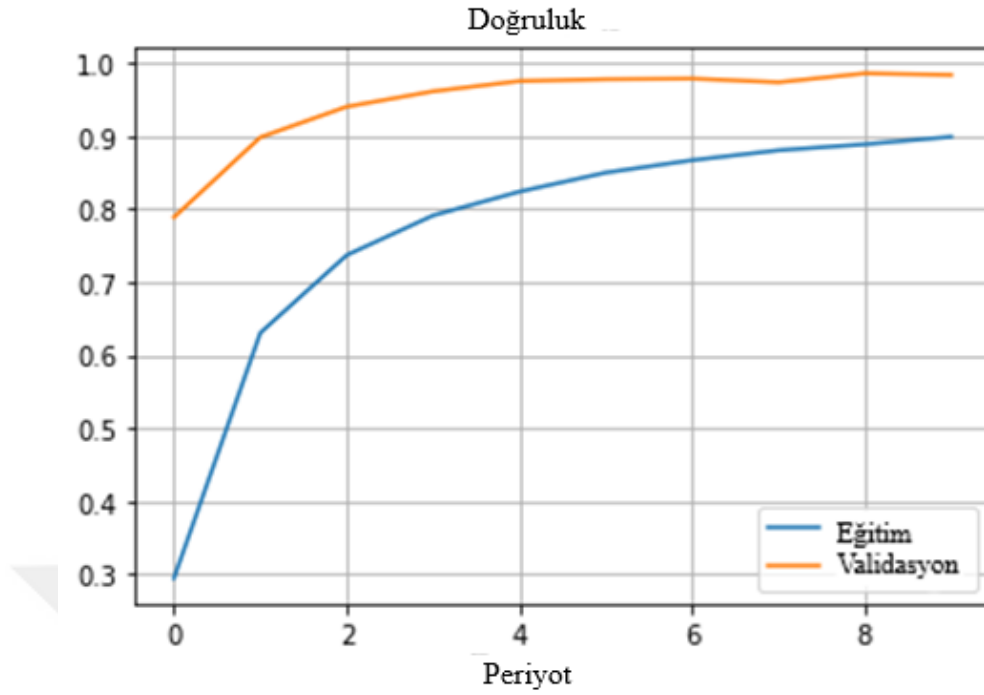


Şekil 4.5 Evrişimli sinir ağı mimarisi(CNN).

Bu verilerin %20'si modelin eğitimi esnasında test için, kalan verinin %80'ninin %20'si validasyon için ve kalan veri ise eğitim için ayrılmıştır. Model eğitilirken verinin tamamı aynı anda eğitime katılmamaktadır.”Batch_size” olarak adlandırılan bölümler halinde eğitime dahil olmaktadır. Zhang vd.(2020) çalışmasındaki model, CIFAR-10 genel veri setinde eğitip test edildiğinde %93,16 doğruluk oranı elde etmişken bu çalışmada. eğitimler sonucunda %97,3 doğruluk oranı elde edilmiş olup elde edilen doğruluk grafikleri Şekil 4.6 ve Şekil 4.7’de gösterilmiştir.



Şekil 4.6 Veri setinin eğitim ve validasyon kayıp grafiği.



Şekil 4.7 Veri setinin eğitim ve validasyon doğruluk grafiği.

CNN model görüntüyü çeşitli katmanlarla işlemektedir. “Convolutional Layer” özellikleri saptamak için, “Pooling Layer” ağırlık sayısını azaltmak ve uygunluğu kontrol etmek için, “Flattening Layer” ise klasik sinir ağına veri hazırlaması için kullanılmıştır.

Modelimizde iki adet evrişim katmanı kullanılmıştır. Evrişim katmanlarının her birinde 60 adet filtre mevcuttur. Bu katmanların aktivasyon fonksiyonu ReLU aktivasyon fonksiyonudur. ReLU fonksiyonunun kullanılma sebebi işlem yükünün diğer fonksiyonlara göre az olmasıdır. İşlem yükünün az olmasının sebebi ise negatif değerleri 0 olarak tanımasındandır. “Convolutional Layer’a” kameradan gelen framelemler öncelikle siyah beyaz matris formatına çevrilmiştir. Ardından bu matrisler önce 5x5 boyutunda ve 3x3 boyutunda bir filtreleme işleminden geçirilmiştir. Yani Conv2D evrişim katmanında 5x5’lik pencere kullanıldığı için 25 nitelik vektörü öğrendiği anlamına gelmektedir. Böylelikle filtreleme sonucu elde edilen matrisler öznitelik haritasını oluşturmuştur. 5x5 ve 3x3 boyutundan elde edilen öznitelik matrislerine 2x2 boyutunda “Max. Pooling Layer” uygulanmıştır. Modelde iki adet 2x2 “Max. Pooling Layer” bulunmaktadır. En büyükleri biriktirme katmanı olarak da bilinen bu katmanda

öznitelik haritası üzerinde pencereler çıkartılmaktadır ve bu çıkarılan pencerelerdeki kanallardaki en büyük değeri almamızı sağlar. Bu boyut küçültme işleminin yapılma sebebi daha az katsayıyla işlem yapabilmek içindir. Katmanlar arasında overfitting olarak da isimlendirilen modelin ezberleme yapması sorunsalını aşmak için “Dropout” katmanı kullanılmıştır.

Son olarak elde edilen matrislerin “Fully Connected” katmanında kullanılabilmeleri için “Flatten” düzleştirme işlemi gerçekleştirilmiştir çünkü sinir ağı, giriş verilerini tek boyutlu bir diziden almaktadır. Modelin “fully connected” katmanında iki adet katman bulunmaktadır. İlk katmanda 500 adet düğüm vardır ve aktivasyon fonksiyonu için ReLU kullanılmıştır. İkinci ve son katmanımız yapay sinir ağımızın çıktı katmanıdır. Çıktı katmanında 40 adet düğüm bulunmaktadır ve bu katmanda softmax aktivasyon fonksiyonu kullanılmıştır. Çıktı fonksiyonumuzun 40 adet düğümden oluşmasının sebebi mevcut datamızda 40 adet trafik işaretinin bulunması ve bunların girdi olarak kullanılmasıdır. Yani bu durum bir girdi değeri için 40 boyutlu bir vektörün oluşacağı anlamına gelmektedir. Katmanlar oluşturulduktan sonra modelin derlenmesi gerekmektedir. Derleme işlemi “model.compile()” fonksiyonu çağırılarak yapılmıştır. Optimizasyon olarak “Adam” kullanılmıştır. Tüm bu adımların Python yazılım kodu Ek 1’de verilmiştir.

Çizelge 4.1 CNN modeli trafik işareti tanıma algoritması test sonuçları.

Sınıf	Trafik İşareti	Tahmin Olasılığı
1	Hız limiti (20 km/h)	%90-95
2	Hız limiti (30 km/h)	%95-97
3	Hız limiti (50 km/h)	%95-100
4	Hız limiti (60 km/h)	%98
5	Hız limiti (70 km/h)	%90
6	Hız limiti (80 km/h)	%88-95
7	Hız limiti (100 km/h)	%95-100
8	Hız limiti (120 km/h)	%95-100
9	Geçiş izni yok	%89-95
10	3,5 ton üzeri araçlara geçiş izni yok	%90-95

11	Anayol-Tali yol kavşağı	%85-90
12	Anayol	%90-95
13	Yol ver	%95-100
14	Dur!	%100
15	Taşıt trafiğine kapalı yol	%95
16	Kamyon giremez!	%85-90
17	Girişi olmayan yol	%98-100
18	Dikkat!	%93-98
19	Sola tehlikeli viraj	%90
20	Sağa tehlikeli viraj	%90-95
21	Sola tehlikeli devamlı virajlar	%90
22	Kasisli yol	%95-100
23	Kaygan yol	%95-97
24	Sağdan daralan kaplama	%86-92
25	Yolda çalışma	%99
26	Işıklı işaret cihazı	%90-95
27	Yaya geçidi	%87-90
28	Okul geçidi	%85-90
29	Bisiklet giremez	%96-98
30	Gizli buzlanma	%98
31	Vahşi hayvanlar geçebilir	%96-98
32	Bütün yasaklamaların ve kısıtlamaların sonu	%89-95
33	İleriden sağa mecburi yön	%90-94
34	İleriden sola mecburi yön	%90-96
35	İleri mecburi yön	%95-98
36	İleri ve sağa mecburi yön	%92-95
37	İleri ve sola mecburi yön	%90-95
38	Sağdan gidiniz	%83-90
39	Soldan gidiniz	%85-95
40	Ada etrafında dönünüz	%86-93

4.4 Şerit Algılama Uygulaması

Far pozisyonunun ayarlanabilmesi için şerit çizgilerinin merkezden uzaklığı hesaplanmıştır ve bu değer seri port üzerinden Arduino'ya iletilmiştir. Şerit çizgilerinin merkezden uzaklığını hesaplayabilmek için bir takım görüntü işleme prosesleri uygulanmıştır.

4.4.1 Görüntü İşleme Proses Süreci

Öncelikle görüntüde iyi bir kenar tespiti yapabilmek için parlaklık ve kontrast ayarlanarak görüntü gri tonlamaya dönüştürülmüş ve beyaz renk maskelenmiştir. Şerit çizgilerini doğru tespit edebilmek için görüntüdeki gürültüleri ve ayrıntıları ortadan kaldırmak gerekmektedir. Bundan dolayı görüntüdeki gürültüler ve ayrıntılar Gaussian Blur filtresi uygulanarak elimine edilmiştir. Tüm bu işlem adımlarının program çıktısı Şekil 4.8'de gösterilmiştir.



Şekil 4.8 Görüntü filtreleme proses adımları.

Canny kenar tespitinin matematiksel temeli Denklem (3.4) ve Şekil 3.11'de algoritma akış diyagramı kullanılarak Şekil 4.9 'da görünen uygulama çıktısı elde edilmiştir.



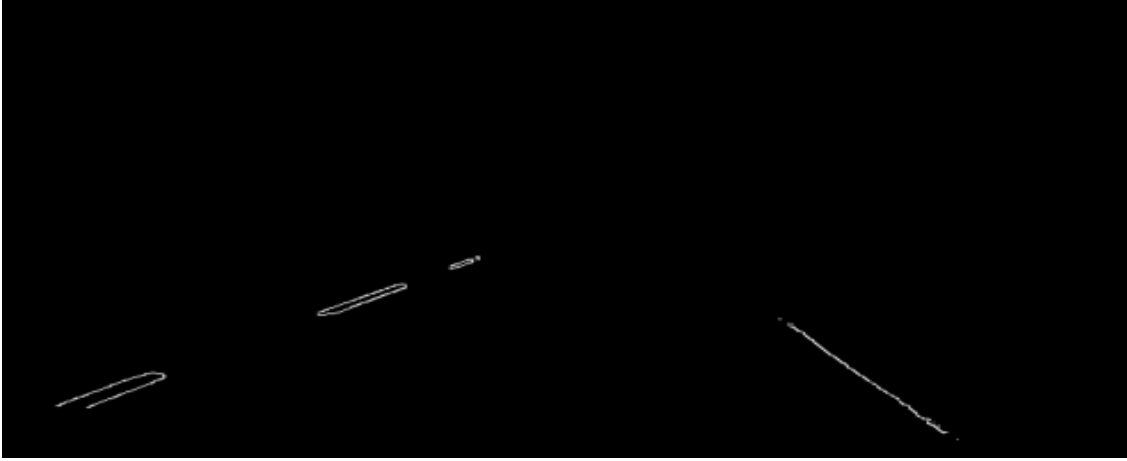
Şekil 4.9 Canny ile kenar tespiti.

Canny kenar tespit algoritmasını uyguladıktan sonra görüntüde sürüş halinde iken bir sürü beyaz renkli nesnenin kenarı algılanmıştır. Bundan dolayı şerit çizgileri haricindeki kenarları görüntüden uzaklaştırabilmek için kameranın görüş açısı sadece aracın önünü gösterecek şekilde bir kuş bakışı alan Şekil 4.10'da belirlenmiştir. Böylelikle görüntünün yol olmayan kısmında çizgi arama işlemi engellenmiştir.



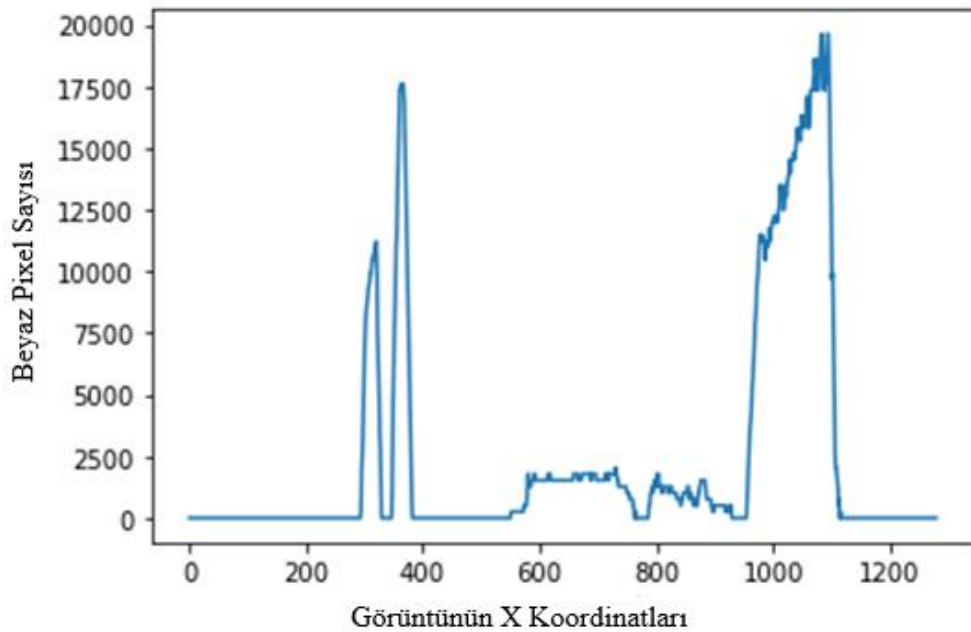
Şekil 4.10 Soldaki ROI, sağdaki kuşbakışı görüntü.

Hough dönüşümü Denklem (3.5) ve (3.6) matematiksel ifadeleri temel esas alınıp ve Şekil.3.11'de gösterilen akış diyagramının algoritması kullanılarak Şekil 4.11 elde edilmiştir.

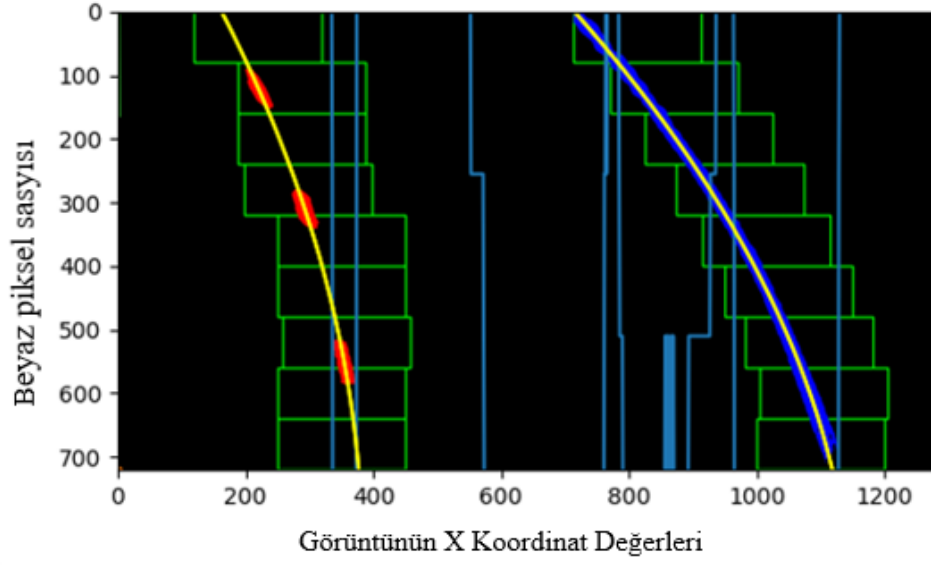


Şekil 4.11 Görüntünün Hough dönüşüm uygulanmış çıktısı.

Bu işlemlerin ardından binary olarak bir görüntü elde edilmiştir. Şerit çizgilerinin pikseller üzerindeki konumunu tespit etmek için görüntüye histogram uygulanmıştır. Histogram grafiği Şekil 4.12’de gösterilmiştir. Böylelikle histogramdaki en yüksek iki tepe şerit çizgilerinin x konumunu ifade etmektedir ve şerit çizgileri görüntüde bu konumlar referans alınarak aranmıştır. Bu başlangıç noktalarından, çerçevenin tepesine kadar olan çizgileri bulmak ve takip etmek için Şekil 4.13’deki çizgi merkezlerinin etrafına yerleştirilmiş bir kayar pencereler kullanılmıştır.



Şekil 4.12 Görüntüdeki tespit edilen beyaz piksellerin histogram grafiği.



Şekil 4.13 Kayan pencereler uygulaması gösterimi.

Kayar pencereler fonksiyonu çalıştırıldıktan sonra genel arama fonksiyonu algılayan şeritlerle oldukça doğru bir şekilde örtüşen sarı bir çizgi çizmektedir. Bu çizgi, direksiyon açılarını tahmin etmede önemli olan eğrilik yarıçapını ölçmek için kullanılmıştır. Bu eğri Denklem (3.7e) ile ifade edilmiştir.

Yukarıdaki iki işlem tamamlandıktan sonra elde edilen değerler piksel cinsinden olduğundan bu değerler metre birimine dönüştürülmelidir. Histogramdan elde edilen sağ ve sol şeritlerin x koordinatların arasındaki mesafe piksel biriminden şerit genişliğini vermektedir. Gerçekte olan metre biriminden olan şerit genişliği, histogramdan elde edilen ve piksel biriminden olan şerit genişliğine bölünerek piksel başına metre birimi elde edilmiştir. Şekil 4.14'te görüldüğü gibi aynı şekilde y eksenini için gerçekte olan şerit uzunluğu çerçeve yüksekliğine bölündüğünde y koordinatında piksel başına metre birimi elde edilmiştir.



Şekil 4.14 Reel şerit genişliği ve uzunluğu.

X eksenini boyunca ortalama metre/piksel değeri X /Çerçeve genişliği ve y eksenini boyunca ortalama metre/piksel değeri Y /çerçeve yüksekliği olarak hesaplanmaktadır. Uygulanan tüm bu işlem adımları sonucunda şerit çizgilerinin tespiti Şekil 4.15’de görüldüğü gibi gece yapılabilir.



Şekil 4.15 Şerit çizgilerin tespit edilmesi.

Şerit tespiti yapıldıktan sonra araç tespiti, şerit tespiti, şeritin merkezden uzaklığı ve trafik işareti tespiti aynı ekran üzerinde Şekil 4.16’daki gibi sürücüye gösterilmektedir.



Şekil 4.16 Sürücü multimedia ekranı.

Kaur ve Kumar (2015) şerit tespiti ile alakalı çalışmalarında mevcut yöntemlerin yüksek kaliteli görüntülerde, kıvrımlı olmayan yani düz şerit çizgilerine sahip yollarda, gece vaktinde yüksek doğruluk oranı yakalandığını gözlemlerken virajlı yollarda ve gece vakitlerinde şerit algılama tekniklerinin yüksek doğrulukla tespit yapamadığını ifade etmişlerdir. Ancak bu çalışmada Şekil 4.15 ve 4.16'da görüldüğü üzere gece vaktinde ve virajlı yolda şerit algılama algoritmasının doğru çalıştığı görülmektedir.

5. TARTIŞMA ve SONUÇ

Gece sürüşleri sürücüler için zorlayıcı ve yorucu olmaktadır ve bu sebepler anlık dikkatsizliğe neden olduğundan kaza riskini de arttırmaktadır. Bu çalışmada gece sürüşlerinde sürücünün görüş alanını arttıracak, anlık dalgınlıklarını ortadan kaldıran tanıma, tespit etme ve uyarma sistemi tasarlanmıştır. Bu teknolojiler otonom olarak bir yazılım vasıtasıyla görevlerini yürütmektedirler. Uzun ve kısa farlar LED teknolojisiyle donatılmıştır. LED matrisin 8 kolunu uzun far 8 kolunu kısa far için kullanılmıştır. Her bir hücre, tanıma ve tespit etme algoritmalarına göre kontrol edilmektedir. Geceleri yüksek doğruluk oranıyla araçları algılayabilen bu modeli eğitmek için gece çekilen far ve arka lambalara ait 1000'den fazla görüntünün olduğu bir veri seti kullanılmıştır. Bu

veri setindeki görsellerin %80'i modelin eğitimi için %20'si ise test için ayrılmıştır. Bu veri seti vasıtasıyla da Python yazılım dilinin sağlamış olduğu Keras kütüphanesi sayesinde gece vaktinde araç tespit eden bir model oluşturulmuştur. Eğitilen modelin tespit edilen görüntüyü tahmin edebilmesi için kameradan alınan gerçek zamanlı görüntüler görüntü işleme proseslerinden geçirilmiştir. Model gece vaktinde bir otoyolda, sokak lambalarının olduğu bir yolda denenmiştir. Bu deneme sonucunda sokak lambalarının aydınlatmasının algoritmayı başarısızlığa uğratmadığı aksine araçların ışık hüzmeleri eğitilen algoritma sayesinde %85 oranında başarıyla tespit edebildiği gözlemlenmiştir. Ancak İkidenden fazla araç tespit edildiğinde, mevcut 8 kolon fazla parçaya bölünmesi gerekmektedir ve bu durumun aydınlatma verimini düşürdüğü anlaşılmıştır. Bundan dolayı da uzun hüzmeler tamamen kapatılmaktadır. Her bir hücrenin kontrol edilmesi, karşıdan gelen araçların ışık hüzmelerinin; sürücünün anlık körlüğe sebep olmasını ortadan kaldırmasına avantaj sağlamıştır. Günümüz orta segment otomobillerin birçoğunda olan uzun far asistanı araç tespiti yaptığında uzun far otomatik olarak açıp kapatmaktadır. Oysa çalışmamızda maliyeti çok da arttırmadan anlık körlüğü önlemenin mümkün olabildiği görülmüştür.

Çalışmada far hüzmelerinin kontrolü eğimli ve virajlı yollarda görüş doğrultusunda aydınlatma sağlamasına imkan tanımadığı için far sistemine pozisyonlama yeteneği de kazandırılmıştır. Eğimli ve virajlı yollarda, tasarlanan far sistemi görüş doğrultusunda aydınlatma sağlayabilmek için servo motorlar yardımıyla sağa-sola ve aşağı-yukarı dönebilmektedir. Direksiyon döndürme açısı ve yönü aynı anda farların yatay düzlemdeki hareketini tayin etmektedir. Direksiyonun dönme açısının farın dönme açısına oranı 24:1 oranında ayarlanmıştır. Böylelikle farın dönme açısı 15° ile sınırlandırılmıştır. Yani direksiyon 360° derece döndürüldüğünde bile farlar maksimum 15° dönebilmektedirler. Yokuş aşağı ve yokuş yukarı gidildiğinde ise sisteme entegre edilen eğim sensörü sayesinde eğimin yönü tespit edilmiştir. Bu yön sayesinde de farlar belirlen sabit açı değerinde aşağı yukarı hareket ettirilmektedir. Sonuç olarak araç tespit edildiğinde hem uzun hüzmelerini tamamen kapatmayan ve aynı zamanda iki eksenle pozisyonlama yaparak aydınlatma sağlayan bu tasarım sürücülerin gece yolculuğunda sürüş konforunu arttırmıştır.

Gece görüş zorluğuna ek olarak diğer bir sorun ise gece vaktinde trafik işaretlerini net olarak görememek ya da yorgunluğun dikkat dağınıklığına yol açması sebebiyle sürücünün bu işaretleri es geçmesidir. Gece karşıdan gelen aracı tespit eden kamera derin öğrenme yöntemiyle tasarlanmış bir model yardımıyla yol kenarındaki trafik işaretlerini de tespit edip sürücüyü araç içinde bulunan ekranda uyarmaktadır. Tasarlanan trafik işareti tanıma modeli 34.800 adet görsel ile eğitilmiş ve test edilmiştir. Eğitilen modelin tespit edilen görüntüyü tahmin edebilmesi için kameradan alınan gerçek zamanlı görüntüler görüntü işleme proseslerinden geçirilmiştir. Eğitimler sonucunda 0,97 test verisinde doğruluk oranı yakalanmıştır. Günümüzde kullanılan bazı otomobil üreticilerinin hız limiti işaretlerini tespit edip aracın anlık hızına müdahale ettiği teknolojiler mevcuttur. Ancak bu çalışmada sadece hız limitleri değil tüm işaretler tespit edilmektedir ve araç içi multimedya ekranında görüntülenmektedir.

Çalışmada tasarlanan farın pozisyonlamasına destek olan son uygulama şerit tespit algoritmasıdır. Şerit tespiti sayesinde yolun eğriliği tespit edilmiş ve farın pozisyonlamasına ek destek sağlanmıştır. Şerit tespitinin yapılamadığı yollarda açısı sensörü devreye girmektedir. Sağ ve sol şerit çizgileri aynı anda tespit edildiği sürece açısı sensörünün veri göndermesi engellenmiştir. Şerit tespiti için bazı görüntü işleme prosesleri uygulanmıştır.

Şerit çizgilerini doğru tespit edebilmek için görüntüdeki gürültüleri ve ayrıntıları ortadan kaldırmak gerekmektedir. Bundan dolayı görüntüdeki gürültüler ve ayrıntılar Gaussian Blur filtresi uygulanarak elimine edilmiştir. Bu bulanıklaştırma tekniğinin görsel etkisi, bir görüntüye yarı saydam bir ekrandan bakmaya benzerdir. Bu özellik de kenar algılama algoritmalarında kullanılma sebebidir. Blurlanan görüntünün Canny ile görüntüdeki tüm kenarların tespiti yapılmıştır. Ancak bu aşamada şeritlerin dışında başka cisimlerin de kenarlarını tespit ettiği görülmüştür. Sadece şeritlerin tespit edildiği alanı sınırlayarak yani görüntüyü kırparak bu alan üzerinde proses yürütülmesi şerit tespiti açısından daha doğru olduğu saptanmıştır. Şerit çizgilerinin tam olarak nerden başladığını tespit edebilmek için histogram çizmek önemlidir. Histogram grafiği incelendiğinde beyaz piksellerin iki farklı tepe oluşturduğu görülmüştür. Sağ ve sol şerit çizgilerinin nerde başlayıp nerde biteceği bu grafikte net olarak anlaşılmaktadır. Bu

başlangıç noktalarından, çerçevenin tepesine kadar olan çizgileri bulmak ve takip etmek için çizgi merkezlerinin etrafına yerleştirilmiş bir kayar pencereler kullanılmıştır. Kayar pencereler fonksiyonu çalıştırıldıktan sonra genel arama fonksiyonu algılanan şeritlerle oldukça doğru bir şekilde örtüşen sarı bir çizgi çizmektedir. Bu çizgi, direksiyon açılarını tahmin etmede önemli olan eğrilik yarıçapını ölçmek için kullanılmıştır.

Otomobil üreticilerin birçoğu çalışmada bahsedilen teknolojilerin en az birini ürettikleri araçlarda kullanmaktadırlar. Volvo, BMW, Audi vb. üst segment otomobil markaları direksiyon açısını ve dikey eksen etrafındaki sapmayı bulabilmek için pahalı sensörler kullanmaktadırlar ve arabalarının farlarını döndürebilmek için küçük elektrik motorları kullanmaktadırlar. Bu arabalarda bu teknolojilerin maliyeti 1000\$'a kadar çıkmaktadır (Shereyas S vd. 2014). Ancak yapılan bu çalışma eklenti olarak araçlara sonradan eklenebilmektedir ve maliyet açısından da daha müşteri için ulaşılabilir seviyede olduğu anlaşılmıştır.

6. KAYNAKLAR

- Andrei M A, Boianjiu C A, Tarbă N, Voncilă M L, 2022, Robust Lane Detection and Tracking Algorithm for Steering Assist Systems, *Machines*,10,1-19.
- Ang J C, Mirzal A, Haron H, Hamed H N A, 2015, Supervised, Unsupervised and Semi-supervised Feature Selection: A Review on Gene Selection, *IEEE/ACM Transactions on Computational Biology and Bioinformatics*,13, 971-989.
- Balaji R D, 2020, A Case study on Automatic Smart Headlight System for Accident Avoidance, *International Journal of Computers Communications & Informatics*, 2, 70-77.
- Bullough J D, Donnell E T, Rea M S, 2013, To Illuminate or not to Illuminate: Roadway Lighting as It Affects Traffic Safety at Intersections, *Accident and Prevention*, Vol 3 ,65-77.
- Büthorn R, Tadjine H H, Auerbach B, Schulze K, 2013, Advanced Microsystems for Automotive Applications, pp 77-88.
- Chen Y, Chiu S, 2018, New method of automatic control for vehicle headlights, *Optik*, 157, 718-723.
- Ciresan D, Meier U, Gamberdella L M, Schmidhuber J, Convolutional Neural Network Committees for Handwritten Character Classification, 2011 International Conference on Document Analysis and Recognition, 18-21 September 2011, Beijing.
- Connell J H, Herta B W, Pankanti S, Hess H, Pliefke S, 2011, A Fast and Robust Intelligent Headlight Controller for Vehicles, 2011 IEEE Intelligent Vehicles Symposium (IV), June 5-9, Baden-Baden Germany.
- Cunningham P, Cord M, Delany S J, 2008, Supervised Learning in Machine Learning Techniques for Multimedia, Springer, 49p, Heidelberg.
- De J, Universal Adaptive Headlight System, 2014, IEEE International Conference on Vehicular Electronics and Safety, Hyderabad, India, December 16-17.
- Fokkinga M, 2011, The Hough Transform, Cambridge University Press, 21, 129-133.

- Goodfellow I, Bengio Y, Courville A, 2016, Deep Learning, MIT press Cambridge, 800p, Montreal.
- Gowrishankar M, Gowtham G, Kumaravel K, Vijai A K, 2019, Enhancement of Adaptive Headlight System, Asian Journal of Applied Science and Technology, Volume 3, Issue 1, Pages 335-343.
- Kaur G, Kumar D, 2015, Lane Detection Techniques: A Review, International Journal of Computer Application, Volume 112, 4-8.
- Kelleher J D, 2019, Deep Learning, MIT press, 296p, Cambridge.
- Kher S, Bajaj P, Sharma P C, 2000, A Novel Fuzzy Control of the Intensity of Headlights for Night Driving, Intelligent Vehicles Symposium IV 2000, February 2000, 76-80.
- LeCun Y, Bengio Y, Hinton G, 2015, Deep Learning, Nature, 521, 436-444.
- Lin T Y, Maire M, Belongie S, Hays J, Perona P, Ramanan D, Dollar P, Zitnick C.L, 2014, Microsoft COCO: Common Object in context, ECCV 2014, 8693, 740-755.
- Long X, He J, Zhou J, Fang L, Zhou X, Ren F, Xu T, 2015, A Review on Light-Emitting Diode Based Automotive Headlights, Renewable and Sustainable Energy Reviews, 29-41.
- Magar S G, 2015, Adaptive Front Light Systems of Vehicle for Road Safety, 2015 International Conference on Computing Communication Control and Automation, 26-27 February 2015, Pune.
- Poornima G R, Harih V, 2019, Vehicle Headlight Automation With Smart Energy Management System, 2019 4th International Conference on Recent Trends on Electronics, Information, Communication & Technology (RTEICT-2019), May 17-18, Auckland University of Technology.
- Scherer D, Müller A, Behnke S, 2010, Evaluation of Pooling Operations in Convolutional Architectures for Object Recognition, Springer, 92-101.
- Sekehravani E A, Babulak E, Masoodi M, 2020, Implementing Canny Edge Detection Algorithm For Noisy Image, Bulletin of Electrical Engineering and Informatics,

pp.1440-1410.

- Shashikant S, Mechkul M A, 2019, Adaptive front Light System, IOSR Journal of Electronics and Communication Engineering, 14, 5-9.
- Shreyas S, Raghuraman K, Padmavathy AP, Prasad S A, Devaradjane G, 2014, Adaptive Headlight System for Accident Prevention, 2014 International Conference on Recent Trends in Information Technology, Chennai, India, 1-6.
- Subhi A B, Hasoon F N, Fadhil H A, Manic S, Biju R, Smart Vehicle Headlights Control System, AIP Conference Proceedings 2137, 3-4 July 2019, Amsterdam.
- Aydın F, Ünalı M, Otomobillerde Yeni Aktif Güvenlik Sistem: Adaptif Far Sistemi, Technological Applied Sciences 5 (2010), 5, 288-298.
- Wolf C, Jolion J M, Chassaing F, 2002, Text Localization, Enhancement and Binarization in Multimedia Documents, IEEE, 1037-1040.
- Zhang J, Wang W, Lu C, Wang J, Sangaiah A K, 2020, Lightweight Deep Network for Traffic Sign Classification, Annals of Telecommunications, 75, 369-379.

İnternet Kaynakları

- 1- <https://headlights.com/the-history-of-headlights/>, 21.12.2021
- 2- <https://carfromjapan.com/article/car-maintenance/history-of-automotive-headlights/>, 21.12.2021
- 3- <http://trafik.gov.tr/kurumlar/trafik.gov.tr/04-Istatistik/Aylik/Ocak22.pdf>, 05.01.2022
- 4- <https://webthesis.biblio.polito.it/14693/1/tesi.pdf>, 20.01.2022
- 5- <https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6>, 20.07.2022
- 6- <https://www.intmath.com/applications-differentiation/8-radius-curvature.php>, 24.08.2022
- 7- <https://developer.nvidia.com/blog/deep-learning-nutshell-history-training/>, 30.09.2022

EKLER

EK 1. Çalışmanın Arduino ve Python Yazılımına Ait Bazı Kod Örnekleri

```
#include "FastLED.h"
#include<Servo.h>

#define NUM_LEDS 32
#define DATA_PIN 4

CRGB led_matrix[NUM_LEDS];

//char ch[3] = "";
char ch[6]="";
char lane[3];
String lane_value;
Servo motor;
const byte potansiyometre_pin=A0;
int pot_deger;
int k;

/////TEK ARAC/////
////////////////////////////////////
void position0()
{
    for(int i = 0; i <= 24; i+=8)
    {
        led_matrix[i] = CRGB::White; //column 1
        led_matrix[i+1] = CRGB::White; //column 2
        led_matrix[i+2] = CRGB::White; //column 3
        led_matrix[i+3] = CRGB::White; //column 4
        led_matrix[i+4] = CRGB::White; //column 5
        led_matrix[i+5] = CRGB::White; //column 6
        led_matrix[i+6] = CRGB::White; //column 7
        led_matrix[i+7] = CRGB::White; //column 8
        FastLED.show();
    }
}

***
***
```

```

***
***

void position9()
{
    for(int i = 0; i <= 24; i+=8)
    {
        led_matrix[i]    = CRGB::Black; //column 1
        led_matrix[i+1]  = CRGB::Black; //column 2
        led_matrix[i+2]  = CRGB::Black; //column 3
        led_matrix[i+3]  = CRGB::Black; //column 4
        led_matrix[i+4]  = CRGB::Black; //column 5
        led_matrix[i+5]  = CRGB::Black; //column 6
        led_matrix[i+6]  = CRGB::Black; //column 7
        led_matrix[i+7]  = CRGB::Black; //column 8
        FastLED.show();
    }
}

////////////////////////////////////
////END ////

////////////////////////////////////
void position01()
{
    for(int i = 0; i <= 24; i+=8)
    {
        led_matrix[i]    = CRGB::Black; //column 1
        led_matrix[i+1]  = CRGB::Black; //column 2
        led_matrix[i+2]  = CRGB::White; //column 3
        led_matrix[i+3]  = CRGB::White; //column 4
        led_matrix[i+4]  = CRGB::White; //column 5
        led_matrix[i+5]  = CRGB::White; //column 6
        led_matrix[i+6]  = CRGB::White; //column 7
        led_matrix[i+7]  = CRGB::White; //column 8
        FastLED.show();
    }
}

```

```

***
***
***
***
***
***
***

void position78()
{
    for(int i = 0; i <= 24; i+=8)
    {
        led_matrix[i]    = CRGB::White; //column 1
        led_matrix[i+1]  = CRGB::White; //column 2
        led_matrix[i+2]  = CRGB::White; //column 3
        led_matrix[i+3]  = CRGB::White; //column 4
        led_matrix[i+4]  = CRGB::White; //column 5
        led_matrix[i+5]  = CRGB::White; //column 6
        led_matrix[i+6]  = CRGB::Black; //column 7
        led_matrix[i+7]  = CRGB::Black; //column 8
        FastLED.show();
    }
}

void serit_servo()
{
    //lane_value=lane;
    //if (lane_value.toFloat()> 0.3)
    //motor.write(lane_value.toFloat()*100);
    motor.write(k);

}
//////////////////////////
//////END //////////

```

```

import tensorflow as tf
import tensorflow.compat.v1 as tf
import time
import pyautogui, sys
import serial
from object_detection.utils import label_map_util
from object_detection.utils import visualization_utils as vis_util

#####
#### START - TEK ARAC #####
def drawCurtainSingle(inpFrame, inpX, inpSer):

    ##Get the frame width
    fWidth = inpFrame.shape[1]
    print("Frame", fWidth)
    print("Single Car Detected")

    *****
    *****
    *****
    *****

    if(inpX < col1):
        posX1 = "0"
        print("POSX1:", posX1.encode())

    elif(inpX >= col1 and inpX < col2):
        posX1 = "1"
        print("POSX1:", posX1.encode())

    elif(inpX >= col2 and inpX < col3):
        posX1 = "2"
        print("POSX1:", posX1.encode())

    elif(inpX >= col3 and inpX < col4):
        posX1 = "3"
        print("POSX1:", posX1.encode())

```

```

ser = serial.Serial('COM4', 9600, timeout=0)
#ser2=serial.Serial('COM3', 9600, timeout=0)

#####TRAFFIC SIGN
def traffic_sign(imgOriginal):
    def getCalssName(classNo):
        if classNo == 0:
            return 'Hiz Limiti 20 km/h'
        elif classNo == 1:
            return 'Hiz Limiti 30 km/h'
        elif classNo == 2:
            return 'Hiz Limiti 50 km/h'
        elif classNo == 3:
            return 'Hiz Limiti 60 km/h'
        elif classNo == 4:
            return 'Hiz Limiti 70 km/h'
        elif classNo == 5:
            return 'Hiz Limiti 80 km/h'
        elif classNo == 6:
            return '80 km/h Hiz Limiti Sonu'
        elif classNo == 7:
            return 'Hiz Limiti 100 km/h'
        elif classNo == 8:
            return 'Hiz Limiti 120 km/h'
        elif classNo == 9:
            return 'Gecis izni Yok!'

    *****
    *****
    *****

    img = np.asarray(imgOriginal)
    img_size = cv2.resize(img, (32, 32))
    img_gray = cv2.cvtColor(img_size, cv2.COLOR_BGR2GRAY)
    img_equal = cv2.equalizeHist(img_gray)/255
    img_last = img_equal.reshape(1, 32, 32, 1)

```

```

#video = cv2.VideoCapture(0,cv2.CAP_DSHOW)
video=cv2.VideoCapture("viraj.mp4")
video=readVideo()
while(True):

    ret, frame = video.read()
    frame_expanded = np.expand_dims(frame, axis=0)

    height, width = frame.shape[:2]

    (boxes, scores, classes, num) = sess.run(
        [detection_boxes, detection_scores, detection_classes, num_detections],
        feed_dict = {image_tensor: frame_expanded})

    traffic_sign(frame)

    vis_util.visualize_boxes_and_labels_on_image_array(
        frame,
        np.squeeze(boxes),
        np.squeeze(classes).astype(np.int32),
        np.squeeze(scores),
        category_index,
        use_normalized_coordinates=True,
        line_thickness=4,
        min_score_thresh=0.40)

    medX1 = None    #ilk arac pozisyonu
    medX2 = None    # ikinci arac pozisyonu
    first = True
    final_score = np.squeeze(scores)
    count = 0       # tespit edilen araç sayısı

```

```

def myModel():
    no_Of_Filters=60
    size_of_Filter=(5,5)
    size_of_Filter2=(3,3)
    size_of_pool=(2,2)
    no_Of_Nodes = 500
    model= Sequential()
    model.add((Conv2D(no_Of_Filters,size_of_Filter,input_shape=(imageDimesions[0],imageDimesions[1],1),
    model.add((Conv2D(no_Of_Filters, size_of_Filter, activation='relu'))))
    model.add(MaxPooling2D(pool_size=size_of_pool))

    model.add((Conv2D(no_Of_Filters//2, size_of_Filter2,activation='relu'))))
    model.add((Conv2D(no_Of_Filters // 2, size_of_Filter2, activation='relu'))))
    model.add(MaxPooling2D(pool_size=size_of_pool))
    model.add(Dropout(0.5))

    model.add(Flatten())
    model.add(Dense(no_Of_Nodes,activation='relu'))
    model.add(Dropout(0.5))
    model.add(Dense(noOfClasses,activation='softmax'))

    model.compile(Adam(lr=0.001),loss='categorical_crossentropy',metrics=['accuracy'])
    return model

```

