



PYTHON İLE PLAKA TANIMA SİSTEMİ

Yazılım Mühendisliği Ana Bilim Dalı

Bitirme Projesi

Efe ÖNİER

Tez Danışmanı: Doç Dr. Vahide BULUT

Ocak 2023

İzmir Kâtip Çelebi Üniversitesi Fen Bilimleri Enstitüsü öğrencisi **Efe Önier** tarafından hazırlanan **Python ile Plaka Tanıma Sistemi** başlıklı bu çalışma tarafımızca okunmuş olup, yapılan savunma sınavı sonucunda kapsam ve nitelik açısından başarılı bulunarak jürimiz tarafından YÜKSEK LİSANS TEZİ olarak kabul edilmiştir.

ONAYLAYANLAR:

Tez Danışmanı: **Doç. Dr. Vahide BULUT**
İzmir Kâtip Çelebi Üniversitesi

Python İle Plaka Tanıma Sistemi

ÖZ

Günümüzde artan popülasyon buna paralel artış gösteren araç kullanımıyla birlikte, trafik problemleri, park problemleri ve güvenlik problemleri gibi sorunlara çözüm bulabilmek için teknolojik gelişmeler ışığında çeşitli yazılımlar geliştirmektedir. Trafik denetleme, otopark giriş çıkış kontrolleri, gişe otomasyonları gibi uygulamalar ile bu problemlere çözüm sunulabilir.

Bu çalışmada, python ile görüntü işleme algoritmaları kullanılarak plaka tanıma sistemi geliştirmesi hedeflenmiştir.

Anahtar Sözcükler: Görüntü işleme, Plaka tanıma sistemi, Karakter tanıma

License Plate Recognition System With Python

Abstract

Constantly increasing population has a direct affect on the amount of vheicles in the traffic. Parallel to this fact, a lot of software is being developed with the light of technology in order to find a solution for the increasing traffic, insufficient parking lots and security related issues.

These kind of problems can be solved/reduced with aps like traffic control/inspection, simplifying parking entry and exit, toll automation.

This study aims to develop a license plate recognition system by using image processing algorithms with Python.

Keywords: Engineering, Image processing, License plate recognition system, Character recognition

İçindekiler

Öz	ii
Abstract	iii
Şekiller Listesi.....	vi
Tablolar Listesi.....	vii
Kısaltmalar Listesi	viii
Giriş	1
Derin Öğrenme	3
2.1 Derin Öğrenme Mimarileri	4
2.1.1 Evrimsel Sinir Ağları	4
2.1.2 Tekrarlayan Sinir Ağları.....	4
2.1.3 Uzun/Kısa Dönemli Bellek.....	5
2.1.4 Kısıtlanmış Bozaltman Makineleri.....	6
2.1.5 Derin İnanç Ağı	7
2.1.6 Derin Oto-Kodlayıcılar.....	7
2.2 Görüntü İşleme Kütüphaneleri.....	8
2.2.1 Tensorflow	8
2.2.2 Keras	8
2.2.3 Numpy	9
2.2.4 Python Image Library	9
2.2.5 İmutils.....	9
2.2.6 PyTesseract.....	9
2.2.7 Skimage	10
2.2.8 Open CV	10
Plaka Tanıma Sistemi	11
3.1 Plaka Tanımlama Adımları	11

3.1.1 Görüntü Üzerinden Plakanın Ayrıştırılması.....	12
3.1.2 Görüntü Ön İzleme	12
3.1.3 Rgb'den Gri Seviyeli Görüntüye Dönüştürme İşlemi	12
3.1.4 Gri Seviyeli Resim Üzerinde Gürültü Temizleme	13
3.1.5 Histogram Eşitleme İşlemi	13
3.1.6 Morfolojik İşlem.....	14
3.1.7 Pixel Çıkartma İşlemi	14
3.1.8 Görüntü Eşikleme İşlemi	14
3.1.9 Kenar Algılama İşlemi.....	15
3.1.10 Genişletme İşlemi	15
3.1.11 Plaka Bölgesinin Maskelenmesi.....	16
3.2 Plaka Üzerinde Karakter Ayrıştırma.....	16
3.2.1 Eğimin Düzeltilmesi	16
3.2.2 Karakter Parçalama.....	17
3.2.3 Bölgelerin Belirlenmesi	18
3.2.4 Şablon Eşleştirme Yöntemi	19
3.2.5 Yapay Sinir Ağları Yöntemi.....	19
Sonuç	23
Kaynaklar	25
Özgeçmiş	26

Şekiller Listesi

Şekil 1	ABD, Türkiye ve Bulgaristan Plaka Örnekleri	2
Şekil 2	Derin Öğrenme	3
Şekil 2.1	RNN Mimarisi	5
Şekil 2.2	LTSM Mimarisi	6
Şekil 2.3	Boltzman Makineleri	6
Şekil 2.4	Derin İnanç Ağı	7
Şekil 2.5	Oto Kodlayıcı Şeması	8
Şekil 3.1	Rgb Uzayında Bulunan Görüntü	13
Şekil 3.2	Gri Seviyeye Dönüştürülmüş Görüntü	13
Şekil 3.3	Histogram Eşitleme Filtresi Uygulanan Görüntü	14
Şekil 3.4	Görüntü Eşikleme İşlemi Yapılmış Görüntü	15
Şekil 3.5	Canny Edge ve Geniştletme Sonrası Ortaya Çıkan Görüntü	16
Şekil 3.6	Kenar Sınır Belirleme Formülü	17
Şekil 3.7	Karakter Ayırıştırma İşlemi	18
Şekil 3.8	Bölgelerin Belirlenmesi	19
Şekil 3.9	Yapay Sinir Ağları	20
Şekil 3.10	İleri Beslemeli Ağ	20
Şekil 3.11	Plaka Karakter Matris Tanımı	21
Şekil 3.12	Yapay Sinir Ağları İle Karakter Tanımlama Blok Diagramı	21
Şekil 3.13	Yapay Sinir Ağı Eğitimi	22

Tablolar Listesi

Tablo 1	Türkiye Plaka Şablonu	18
---------	-----------------------------	----

Kısaltmalar Listesi

Python	Yüksek seviyeli bir programlama dili.
CNN	Convolutional Neural Networks
RNN	Recurrent Neural Network
LTSM	Long Short-Term Memory
RBM	Restricted Boltzmann Machine
Rgb	Red – Green – Blue Renk Kodları
Canny Edge	Canny Kenar Algılama Algoritması
YSA	Yapay Sinir Ağları

Bölüm 1

1. Giriş

Günümüzde artan popülasyon ve gelişen teknolojiyle beraber, yaşamımızı daha güvenli hale getirmek için teknolojiden yararlanarak güvenlik stratejileri geliştirilmeye başlanmıştır. Bunların ilk aklına gelen kameralar vasıtasıyla yüz tanıma sistemleri, araç takibi yapabilmek için plaka tanıma sistemleri vb. sistemler geliştirilmeye çalışılmaktadır. Bu tarz girişimlerin önlem olarak isimlendirilmesinin nedeni hayatımızda büyük öneme sahip bu araçların kontrolünün sağlanmasıdır. Örnek vermek gerekirse hırsızlık, gasp vb. kaçış durumları, köprü ve otoyol geçişleri, otopark giriş-çıkış esnasında kontrolü sağlayabilmek ve seyir halindeki tüm motorlu araçların bilgilerinin kaydedildiği bir veri tabanı üzerinden kontrol sağlanmasıdır. Bu örnek gibi birçok eklenebilecek pozitif ya da negatif durumlarda ihtiyaç duyulan, bu tarz güvenlik sistemlerinin çalışma prensibi, araçların belirli özellikleri üzerine kuruludur. Bu temel bilgiler araçların, marka, model, model yılı, renk ve plaka vb. ayırt edici özellikleridir.

Teknolojideki gelişmelerle beraber takip sistemleri üzerine farklı çalışmalar sürekli yapılmaktadır. Görüntü işleme sistemleri sayesinde aracın görüntüsünden, model, renk ve plaka gibi bilgiler hızlı ve doğru bir şekilde tespit edilebilmektedir. Plaka tanıma sistemi insan gücünün en aza indirgenebilmesine fayda sağlamaktadır. Plaka tanıma sisteminde ilk olarak araç tespiti yapılır. Tespit edilen araç üzerinde plaka yeri tespiti yapılmasıyla beraber bulunan plaka üzerindeki karakterler çıkartılıp plakanın belirlenmesi aşamalarından oluşmaktadır. Araç plakaları ülkeden ülkeye farklılık göstermektedir. Şekil 1’de görüleceği üzere, Türkiye, Bulgaristan ve Amerika ülkelerine ait plaka örnekleri sergilenmektedir.



Şekil 1: ABD, Türkiye ve Bulgaristan Plaka Örnekleri

Gelişen derin öğrenme algoritmalarıyla beraber daha hızlı ve daha doğru sonuçlar veren sistemler geliştirilebilmektedir. Derin Öğrenme, derin sinir ağlarını ifade eden bir makine öğrenme yöntemidir. Verilen bir veri kümesini yapay sinir ağlarını kullanarak çıktıları tahmin edecek yapay zekâyı eğitmemize imkân veren bir sistemdir. Derin öğrenmenin uygulanabilmesi için çok fazla veriye ihtiyaç duyulmaktadır. Derin öğrenme makine öğrenme yönteminin daha gelişmiş şeklidir. En temel fark veri içerisinden kendi kendine öğrenerek yeni özellikler oluşturmasıdır. Belirli özelliklerin makine öğrenmesinde kullanıcılar tarafından doğru bir şekilde tanımlanması ve oluşturulması gerekirken derin öğrenmede bu özellikleri sistemin üretip etiketlemesi ve sonucunda yapay sinir ağlarını kullanarak bir çıktı üretmesi beklenir.

Bölüm 2

2. Derin Öğrenme

Derin öğrenme, yapay sinir ağlarının büyük miktarda veriden öğrendiği makine öğreniminin bir alt kümesidir. İnsan beyninin çalışma şekli örnek alınarak genel hatlarıyla modellenen algoritmalar olan sinir ağları katmanları tarafından desteklenir. Büyük veriler ile eğitim, sinir hücrelerini konfigüre etmektedir. Sonuç olarak, eğitilen sinir ağı yeni verileri işleyen derin öğrenme modelidir. Özetle, verilen bir veri seti ile sonuçları tahmin edebilen, birden çok katmandan oluşan bir makine öğrenmesidir. Derin öğrenme, makine öğrenmesinin, makine öğrenmesi ise yapay zekanın bir alt dalı olarak adlandırılabilir.



Şekil 2: Derin Öğrenme

2.1. Derin Öğrenme Mimarileri

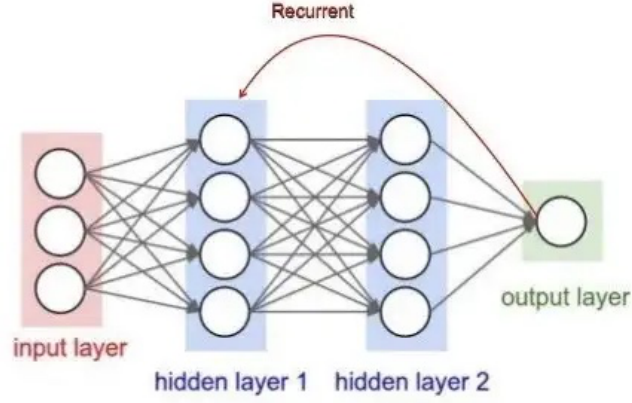
2.1.1. Evrişimsel Sinir Ağları

Evrişimsel sinir ağı (convolutional neural networks- CNN) bir girdi görüntüsünü alıp, görüntüdeki çeşitli nesnelere birbirinden ayırabilen bir derin öğrenme algoritmasıdır. Evrişimsel sinir ağları, temel olarak görüntüleri sınıflandırmak, benzerliklerle kümelemek ve sahnelerde nesne tarama yapmak için kullanılan derin öğrenme yapay sinir ağlarıdır. Evrişimsel sinir ağları yapısı gereği input olarak resim ya da video alır. Resimleri alırken uygun formatta olması gerekir [1].

2.1.2. Tekrarlayan Sinir Ağları

İnsanlar düşüncelerini öğrendiği deneyimlere dayanarak oluşturur. Bir cümleyi okuduğunuz zaman okuduğunuz her kelimeyi bir önceki kelimeyi referans alarak anlarız. Önceki kelimeyi hatırlayarak sonraki kelimeleri anlamlandırırız. Geleneksel sinir ağları böyle bir şey yapamaz. RNN ise bu tür sorunları ele alır. Tekrarlayan sinir ağları fikri 1980lerde Basit Tekrarlayan Ağ (Simple Recurrent Network) olarak atılmıştır. Sonraki zamanlarda Jeff Elman tarafından 1990larda RNN önerilmiştir. Tekrarlayan sinir ağlarındaki temel prensip sıralı bilgileri kullanmaktır. Geleneksel bir sinir ağında tüm girdiler ve çıktılar birbirinden bağımsız olduğu varsayılmaktadır.

Örneğin bir cümle içerisinde bir sonraki kelimeyi tahmin edebilmek için, hangi sözcüklerin o anki kelimedenden önce geldiğini bilmek gerekmektedir. RNN mimarisinin yinelenen olarak adlandırılmasının temel sebebi, bir dizi girdisinin her dizi ögesi için bir önceki çıktıyı referans alarak kontrolü yerine getirmesidir.

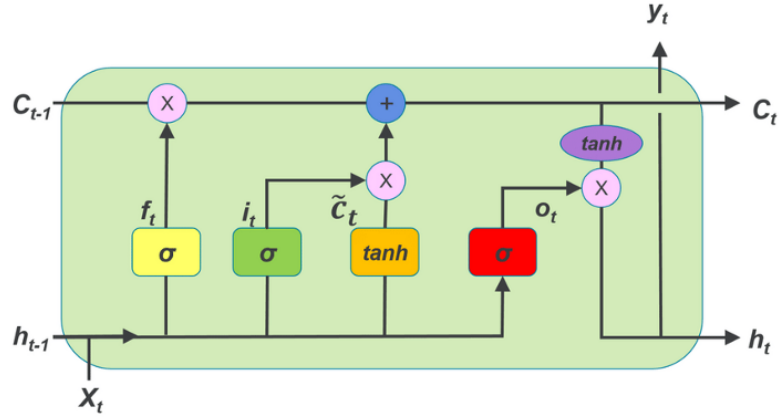


Şekil 2.1: RNN Mimarisi

RNN mimarisinin çalışma şekli şu yapıdadır. Bir giriş katmanına, iki gizli katmana ve bir çıkış katmanına sahiptir. Bu katmanların hepsi bağımsız şekilde çalışmaktadır. Dolayısıyla her katmanın bir görevi vardır. Ve her katman farklı bir görevi yerine getirir. Her katmandaki yapıların ağırlıkları ve katmana özgü eşik değeri vardır. Bu sistemin daha olası sonuçlar vermesini amaçlar. Tekrarlanan bu aşamalar bir önceki giriş durumunu depolar ve yeni elde edilen giriş değeriyle birleştirir. Bu sayede yeni elde edilen girişin bir önceki girişle ilişkisi sağlanır [2].

2.1.3. Uzun/Kısa Dönemli Bellek

LTSM ağlarının RNN ağlarından bir farkı yoktur. RNN'ler de dezavantaj olarak bilinen bağlam boşluklarının tahmin edilmesi ihtiyacından dolayı 1997 yılında LTSM'ler ortaya çıkmıştır. Gizli durumları hesaplamak için LTSM ağlarında bir yapı kullanılır. LTSM içerisinde hafıza hücreleri yer alır. Bu hafıza hücreleri önceki durum ile girdi bilgisi tutan hücrelerdir. Ağ mimarisi içerisinde yer alan bu hücreler hangi verinin tutulacağına veya hangi verinin silineceğine karar verirler. Devam eden aşamada ise önceki durumu mevcut bellek ile giriş verisini birleştirirler. Bu yaklaşımla uzun vadeli bağılıkları ortadan kaldırarak veri dizilerinin devam ettirilmesi sağlanmıştır.

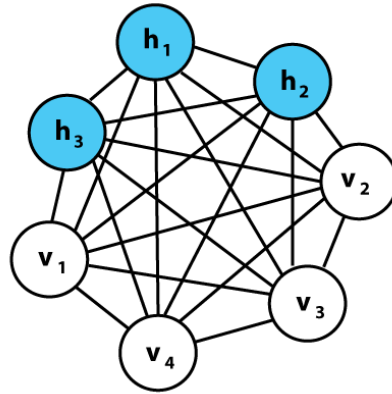


Şekil 2.2: LSTM Mimarisi

LSTM yapısı bilginin hücrenin içine ya da dışına akışını denetleyen kapılardan oluşur. Girdi olarak verilen kapı yeni gelen bilginin hafızaya ne zaman girmesi gerektiğini denetler. Unutma- hafıza kapısı mevcutta olan bilginin unutulmasını ve yeni verilerin hatırlanmasını sağlar. Çıkış aşaması ise hücredeki bilginin hücrenin çıkışına ne zaman kullanılacağını denetler[3].

2.1.4. Kısıtlanmış Boltzmann Makineleri

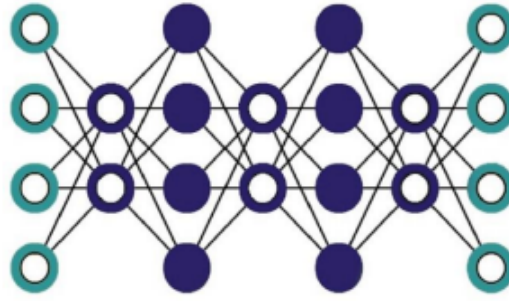
Kısıtlanmış Boltzmann makinesi, Harmonium adıyla ilk olarak 1986 yılında ortaya çıkmıştır. 2006 yılında da Geoffrey Hinton ve arkadaşları bir öğrenme algoritması ortaya çıkartmıştır. RBM aslında Boltzmann makinelerinin bir çeşididir. Görünür ve gizli noktalar olmak üzere birbirine bağlantılı graflardan oluşur. Sınıflandırma, regresyon ve özellik öğrenimi işlemlerini yapan Boltzman makinesi giriş veri seti üzerinden olasılık dağılımlarını öğrenebilen bir sinir ağıdır [4].



Şekil 2.3: Boltzman Makineleri [5]

2.1.5. Derin İnanç Ağı

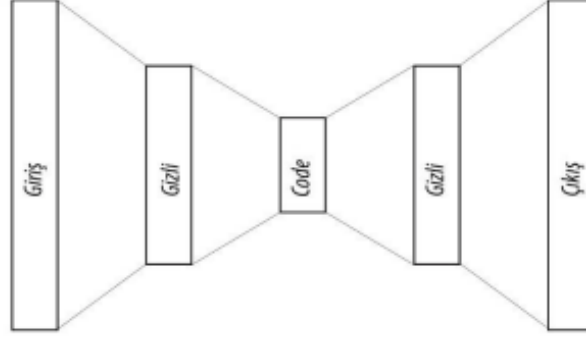
Hinton RBM'i kullanarak Derin İnanç Ağları yığımını oluşturmuş ve bu ağın eğitilebileceğini göstermiştir. Derin inanç ağları veri seti hiyerarşik temsilini çıkartmayı amaçlayan grafiksel modellerdir. Derin inanç ağları art arda gelen kısıtlı boltzman makineleri katmanlarından oluşan bir sinir ağı modelidir. Kısıtlı boltzman makinelerinin eğitilerek öğrenmesiyle gerçekleşir. Grafiksel model katmanından oluşan hem yönlendirilmiş hem de yönsüz kenarlı bir sinir ağı sınıfıdır. Örüntü tanıma ve üretme konularında yetkindir. Denetimsiz bir sinir ağıdır [6].



Şekil 2.4: Derin İnanç Ağı

2.1.6. Derin Oto-Kodlayıcılar

Derin oto kodlayıcılar denetimsiz öğrenme tabanlı makine öğrenmesi sistemidir. Bu sinir ağı diablo ağı olarakta adlandırılmaktadır. Oto kodlayıcılar giriş veri kümesini en az kayıpla en iyi öğrenmeyi hedefler. İleri beslemeli bir sinir ağıdır. Temel olarak 3 bölümden oluşur. Giriş ve çıkış katmanındaki nöron sayıları eşit olmakla birlikte gizli katmandaki nöron sayısı değişkenlik gösterir. Gizli katman içerisindeki nöronların sayısı giriş ve çıkış katmanında yer alan nöronlardan daha az olduğunda veri kümesi sıkıştırılır. Böylece daha az veri ağ içerisinde kalır. Bu da ağın performansını etkiler.



Şekil 2.4: Oto kodlayıcı şeması

2.2. Görüntü İşleme Kütüphaneleri

2.2.1. Tensorflow

Google geliştirici ekibi tarafından 2015 yılında ortaya çıkan açık kaynaklı bir derin öğrenme kütüphanesidir. Geliştiricilerin makine öğrenmesi, derin öğrenme modellerini tasarlamasında büyük kolaylık sağlamaktadır. Tensorflow, görüntü tanıma, doğal dil işleme, diferansiyel denklemler, metin tanımlama vb. sınıflandırma uygulamaları için modelleri eğitimeyi ve çalıştırmayı sağlayan bir kütüphanedir. Bir veya daha fazla işlemci üzerinde çalışmanızı sağlayan yapısı karmaşık hesaplamaları veya sınıflandırma işlemlerini kolaylıkla gerçekleştirmenizi sağlar. Python kullanarak geliştirilmiş olan bu framework günümüzde Javascript, R ya da Swift gibi bir çok programlama dilince de desteklenmektedir.

Tensorflow, kullanıcısı tarafından oluşturulan veri akışlarının grafiğini hesaplar, grafikteki her düğümü matematiksel bir işlem ile ifade ederken aynı zamanda düğümler arasındaki bağlantıyı çok boyutlu veri dizisi olarak ifade eder. Bu diziler tensörler olarak temsil edilirler.

2.2.2. Keras

Keras, Python'da yazılmış açık kaynaklı bir yapay sinir ağı kütüphanesidir. Keras katmanlar, aktivasyon fonksiyonları gibi parametreleri python dilini kullanarak yeni modeller oluşturmaya yarayan bir kütüphanedir. En temel mantıkla tüm yapı katmanlara dayanır. Her yapılacak olan işlem katmanlar arasında gerçekleşmektedir.

İlk katmanda veri giriři saęlanırken son katmanda veri ıkıřı saęlanır. Giriř ve ıkıř katmanları dahil bu yapı model olarak adlandırılır. Modelin eęitimi de bu aralıktta gerekleřtirilir.

2.2.3. Numpy

Numpy, Python'da bilimsel hesaplamalarda kullanılan temel matematik kütüphanesidir. Temelde ok boyutlu numpy dizileri bulunan bu yapı bir dizi içinde diziler ve matrisler barındıran, matematiksel işlemler veya mantıksal işlemler yapılırken ihtiyaç duyulur.

2.2.4. Python Image Library

Python'da bir görüntü üzerinde bir işlem yapabilmek için kullanılan grafik işleme kütüphanesidir. Birok fonsiyonu içinde barındıran kütüphane izim, düzenleme, filtreleme gibi işlemleri hazır fonsiyonlar sayesinde kullanıcıya hızlı bir görüntü işleme sunar.

2.2.5. İmutils

Python' da bir görüntü üzerinde evirme, döndürme, yeniden boyutlandırma, kontur¹ ıkarma, kenar algılama vb. gibi morfolojik işlemleri yapabilmemizi saęlayan bir kütüphanedir.

2.2.6. PyTesseract

Python-tesseract, python' da optik karakter tanıma aracıdır. Pillow tarafından desteklenen görüntü uzantılarının tamamını destekler. Örneęin jpeg, png, tiff, bmp, gif gibi uzantılı görüntüleri okuyup karakter olarak ıktısını sunan bir kütüphanedir.

¹ **Kontur:** Görüntü işlemede, bir rengin veya yoğunluęunun sahip olduęu tüm sürekli noktaları birleřtiren kapalı bir eğridir.

2.2.7. Skimage

Python’ da bir görüntü üzerinde segmentasyon, geometrik dönüşümler, renk alanı manipülasyonu, analiz, filtreleme, morfoloji, özellik algılama gibi işlemleri gerçekleştirebilmek için kullanılır.

2.2.8. Open CV

Open Source Computer Vision yani açık kaynak kodlu görüntü işleme kütüphanesidir. Temelinde C programlama diline uyumlu olarak geliştirilen bu kütüphane, kullanılan tüm programlama dillerine entegre edilebilecek şekilde geliştirilmiş ve geliştirilmeye devam ediyor.

Temel olarak OpenCV kütüphanesi içerisinde görüntü işlemeye yarayan ve makine öğrenmesi gibi çalışmalara katkı sağlayacak 2500’ den fazla algoritması olduğu biliniyor. Bu algoritmalar sayesinde yüz tanıma, nesnelere sınıflandırma, insan hareketlerini tespit etme, plaka tanıma, 3B resimler üzerinde işlem yapabilme, görüntü karşılaştırma, OCR (Optical Character Recognition) Optik Karakter Tanıma gibi birçok işlemin yapılabilmesine imkân veren bir image processing(görüntü işleme) kütüphanesidir. Beş temel bileşenden oluşmaktadır. Bunlar;

- **CV (Computer Vision-Bilgisayarla Görme) Bileşeni:** Bu bileşen görüntü işleme fonksiyonlarını ve algoritmalarını içerir.
- **MLL (Machine Learning Library) Bileşeni:** Bu bileşen adımla anlaşılacağı üzere makine öğrenimi için gerekli olan fonksiyonları ve algoritmaları içerir.
- **HighGUI Bileşeni:** Bu bileşen resim veya videoların kaydedilip silinmesi işlemlerini yerine getiren fonksiyonları içerir.
- **CXCore Bileşeni:** Resim üzerinde çizim yapmamızı sağlayan ve XML desteği veren bileşendir.
- **CvAux Bileşeni:** Şablon eşleştirme, yüz tanıma veya dudak hareketlerini izleme gibi algoritmalar içeren bir bileşendir.

Bölüm 3

3. Plaka Tanıma Sistemi

Plaka tanıma sistemleri ücretli yol girişleri, ışık ihlalleri, otopark veya site giriş çıkışları gibi yerlerde kullanılmaktadır. Temel olarak aracın ön ya da arka tarafından kamera ile yakalanan görüntü morfolojik işlemlerden geçtikten sonra plaka kısmı kırpılır. Kırpılan görüntü üzerinde daha önceden hazırlanmış bir çok karakter üzerinde eğitilmiş ve test edilmiş karakter okuma algoritması kullanılarak plaka okunur ve ilgili ekrana yazdırılır.

3.1. Plaka Tanıma Adımları

Plaka tanıma işlemini yapmak için belirlediğimiz temel adımlar vardır. Bu adımlar ile plaka bulma işlemi yapılmaktadır. Bu adımlar şunlardır;

- Plaka yerinin tespiti
- Plakanın sonraki adımlara uygun bir biçimde yeniden konumlandırılması ve boyutlandırılması
- Zıtlık ve parlaklık gibi görüntü özelliklerinin normalizasyonu
- Karakter ayırma ile görüntü üzerinden karakterlerin çıkarılması
- Optik karakter tanıma
- Ülkeye özgü kelime dizimi ve control işlemi
- Daha güvenilir bir sonuç elde etmek için tanınmalanan değer birden fazla görüntüde ortalamasının alınması.

3.1.1. Görüntü Üzerinden Plakanın Ayırıştırılması

Görüntü üzerinde plakanın doğru bir şekilde ayrılması gerekmektedir. Bu adımda plakanın görüntü üzerinden ayrılmasında belirli koşullar vardır. Sistem tasarlanırken hava koşulları, parlaklık gibi kriterler önemlidir. Tüm şartlarda en optimum düzeyde çalışabilecek sistemi tasarlamak için burada yapılacak işlemler önemlidir. Işığın az olduğu veya ışığın çok olduğu zamanlar olabilir. Bu dengesizlikler alınan görüntüler üzerinde bozulmalara neden olabilir. Doğru bir sonuç elde etmek için gerekli olan işlemlerin yapılması gerekmektedir. İşlemler sırasıyla; alınan görüntü griye çevrilir. Griye çevrilen resim üzerinde gürültü temizleme işlemi yapılır. Morfolojik işlem uygulanır. Histogram eşitleme yapılan görüntüden, morfolojik işlem yapılmış işlem piksel piksel çıkartılır. Görüntü eşikleme işlemi yapılır. Kenar algılama uygulanır. Kontur¹ işlemi yapılır ve son olarak plaka bölgesi maskeleyerek görüntü üzerinden ayrıştırılır.

3.1.2. Görüntü Ön İzleme

Bir araç üzerinde ilk olarak plaka yeri tespiti yapmak için bazı görüntü işleme algoritmalarından geçirmek gerekmektedir. Kameradan alınan görüntü ham bir şekilde gelmekte ve Rgb uzayında bulunmaktadır. Rgb (Red,Green,Blue) uzayında bulunan resimin düzgün bir şekilde işlenebilmesi için gerekli olan bazı dönüşümler yapmak gerekir. Bu adımlar aşağıda sırası ile açıklanmıştır.

3.1.3. Rgb'den gri seviyeli görüntüye dönüştürme işlemi

Resim işleme alındığı zaman ilk olarak RGB uzayındadır. Görüntü üzerinde işlemlerimizi daha basit ve daha kolay bir şekilde plaka yerini tespit etmek için görüntü Rgb'den gri seviyeli görüntüye dönüştürme işlemi yapılır. Rgb modeli üç ana renk olan kırmızı, yeşil ve mavi renklerini içermektedir. Gri seviyeli bir resme sahip olmak için kırmızı kanalın %30 'u, yeşil kanalın %59' u ve mavi kanalın %11'lik kısmı alınmaktadır.

Renkli bir resim ile gri seviye bir resim arasındaki fark, renkli resmin bir pikselinde üç farklı renk vardır. Gri seviyede sadece tek bir değer vardır. Bu üç farklı 0 ile 255 arasında değişen değerleri tek bir 0 ile 255 arasında değişen değer haline getirmek

gerekir. Şekil 3.1’de rgb uzayında bulunan resim şekil 3.2 ‘de gri seviyeli görüntüye dönüştürme işlemi yapılmıştır.



Şekil 3.1: Rgb Uzayında bulunan görüntü



Şekil 3.2: Gri seviyeye dönüştürülmüş görüntü

3.1.4. Gri Seviyeli Resim Üzerinde Gürültü Temizleme

Gürültü temizlemek için ortalama alma, medyan ortalama, bileteral filtreleme gibi çeşitli algoritmalar bulunmaktadır. Bu algoritmalar içerisinde görüntü üzerinde en güçlü gürültüyü azaltan filtre bileteral filtreleme işlemidir. Bileteral filtreleme işlemi kenarları keskin bir hale getirerek gürültünün giderilmesinde oldukça etkilidir. Fakat diğer filtrelere göre biraz daha yavaştır. Plaka tanıma sistemde plaka kenarlarının net bir biçimde ortaya çıkmasını sağlamak ve kaybolan kenarları netleştirmek ortaya çıkarmak için bilaterale filtre kullanılmıştır.

3.1.5. Histogram Eşitleme İşlemi

Plakalar tanınırken ortam, ışık, renk gibi farklılıklar olabilir. Sistemin bu tarz çevresel etkenlerden etkilenmesini minimize etmek için görüntü üzerine histogram işlemi yapılmalıdır. Bir görüntünün histogramı, o görüntü hakkında önemli bilgiler sağlar. Karanlık bir resim grafiğinin düşük gri seviye bölgesinde birikmektedir. Parlak bir

görüntüde ise büyük gri seviyesine birikmektedir. Alınan görüntü üzerinde histogram eşitleme işlemi gerçekleştirilerek iyileştirme işlemi yapılabilir.



Şekil 3.3: Histogram Eşitleme Filtresi Uygulanan Görüntü

3.1.6. Morfolojik İşlem

Morfolojik işlem ile görüntü üzerinde aşındırma ve genişletme işlemleri yapılabilir. Bu filtre ile görüntü üzerindeki gürültünün giderilmesi ile daha net sonuçlar elde edilmektedir. Aşındırma işlemi ile küçük parçalar yok edilerek görüntü tekrar genişletilir.

3.1.7. Pixel Çıkartma İşlemi

Pixel çıkartma veya görüntü çıkartma işlemi bir görüntünün piksel sayısı değerlerinin başka bir görüntüden çıkarılması ile ortaya çıkan bir işlemdir. Bir görüntü üzerinde düzensiz olan bölgeleri dengelemek ve iki resim arasında değişiklikleri saptamak için kullanılmaktadır. Plaka tanıma sisteminde amacımız plaka bölgesini net bir şekilde ortaya çıkartmaktır.

3.1.8. Görüntü Eşikleme İşlemi

Görüntü üzerinde belirli bir eşik değeri altında olan kısımlar 0 üstünde kalan kısımları ise 1 yapmak suretiyle ikili bir görüntü oluşturma işlemi yapılmasıdır. Bu sistemde, eşik değeri her görüntü için farklı olabilir. Elle eşik değeri belirlemek gömülü sistem üzerine çalışan uygulamalar ile mümkün olmaz. Bunu belirlemek için belirli algoritmalar kullanılması gerekir.

Eşik değeri belirleme de Otsu modelinde varyans denilen bir değeri hesaplanır ve bu değerin en düşük olduğu indeks döndürülmektedir. Pixel çıkartma işlemi uygulanmış ve plaka bölgesi net bir biçimde ortaya çıkmış görüntü üzerinde eşikleme işlemi yapılarak plaka bölgesi daha net olarak ortaya çıkması sağlanmıştır.



Şekil 3.4: Görüntü Eşikleme İşlemi Yapılmış Görüntü

3.1.9. Kenar Algılama İşlemi

Canny Edge algoritması kenar algılama için kullanılan bir algoritmadır [7]. Görüntü üzerinde plaka tanıma işlemi yapmak için kenar belirlenmesi gerekmektedir. Canny edge çeşitli görme sistemlerinde kenar algılama uygulaması olarak kullanılmış ve başarılı bir şekilde görüntü üzerindeki kenarları ortaya çıkartmayı başarmıştır. Aşamaları maddeleyecek olursak; Görüntünün gürültülerini azaltmak amacıyla Gaussian çekirdekle konvolüsyon alınarak azaltılır. Gaussian filtre dışında Mean ya da Medyan filtrede kullanılabilir. Gradyan operatörü uygulanır. Bu şekilde görüntünün Gradyan büyüklüğü ve yönü hesaplanır. Bu işlem için Sobel filtresi en çok kullanılan yöntemdir. Bunun dışında Prewitt ve Robert kenar bulma metotları da mevcuttur. Kenarlar Non Maxima baskılama kullanılarak incelemeye alınır. İkili eşikleme uygulanır bu şekilde istenmeyen ayrıntılardan arındırılma işlemi gerçekleştirilir. Güçlü-zayıf ayrımı yapıldıktan sonra baskılama uygulanır ve asıl kenarlarla görüntüye son hali verilir [8].

3.1.10. Genişletme İşlemi

Kenar algılama uygulanan görüntü üzerine uygulanarak kenarları yüksek oranda algılama işlemi gerçekleştirilmiştir. Genişletme işlemi yapmak için bir çekirdek matris oluşturulur. Bu projede 3x3'lük birler matrisi ile işlem gerçekleştirilir. Bir merkezi

nokta seçilir ve oradan başlayarak görüntü tüm pikseller üzerinde belirlenen iterasyon sayısı kadar bu işlem gerçekleştirilir. Bu sayede görüntü üzerinde darbeler ve noktalar bulunur. Maksimum pixel değerini hesaplar ve görüntü üzerinde bulduğu pikseli maksimum pikselle değiştirir. Canny edge uygulandıktan sonra kenarlar parlak bir biçimde gözükmetedir. Genişletme işlemi kullanıldıktan sonra plaka bölgesi daha net şekilde görüntülenmektedir.



Şekil 3.5: Canny Edge ve genişletme sonrası ortaya çıkan görüntü

3.1.11. Plaka Bölgesinin Maskelenmesi

Görüntü üzerinden plakayı ayırtırmak için maskeleme işlemi yapılır. Maskeleme yaparak bundan sonraki işlemlerin sadece plaka üzerinde olan işlemler gerçekleştirilir.

3.2. Plaka Üzerinde Karakter Ayırıştırma

Karakter ayırma işlemi plaka tanıma işlemi yapıldıktan sonra plaka üzerindeki lisans numaralarını yakalamak için kullanılan algoritmadır. Görüntü üzerinde sayıları ve harfleri vurgulayarak görüntüde bulunan diğer nesnelere ayırmaya çalışır. Karakter tanıma işlemi ile otomatik plaka tanıma, verileri düzenlenebilir hale getirme, aranabilir ve kolayca saklanabilir bilgilere dönüştürme işlemlerini gerçekleştirebiliriz. Plakayı bulmak için gerekli filtrelerin ve algoritmaların görüntü üzerine uygulanması gerekir. Bu adımlar karakterlerin daha net bir şekilde gözükmesini sağlayacaktır.

3.2.1. Eğimin Düzeltilmesi

Karakter ayırıştırma işleminin başarılı olması için x eksenine göre eğim tespiti yapılması gerekir. Eğim bulunan resimde x eksenine paralel bir biçimde gelmesi

sağlanmalıdır. Eğim tespiti için karakterlerin x eksenine göre başlangıç ve bitiş noktaları tespit edilmelidir.

Eğik plaka üzerinde karakterlerin başlangıç ve bitiş noktaları bulunduktan sonra bu karakterlerin üst ve alt sınırlarının birbirine göre artma veya azalma oranını bulmak gerekir. Karakterlerin üstünde ve altında bulunan plaka çerçevesi, sınırların belirlenmesinde sorun çıkartmaktadır. Plaka yerini öğrenme yönteminde, yanyana olan piksel değerlerinin farkı 20'nin üzerinde olması durumunda, bu pikseller kenar bölgesi kabul edilir.

3.2.2. Karakter Parçalama

Eğim düzeltme işleminden sonra tüm karakterlerin sınırları aynı x ve y düzlemine gelmektedir. Bu aşamadan sonra plaka resminde yatay düzlemde tarama işlemi yapılarak x ekseninde bulunan siyah piksel noktalarının sayısı bulunur. Görüntünün üst ve alt bölgesinde en az miktarda siyah piksel değerine sahip x eksenini karakterlerin üst ve alt sınırını oluşturmaktadır.

Genişlik: Plaka genişliğinin
X1: x noktası taramanın başlayacağı nokta
X2: x noktası taramanın bitirileceği nokta
X1: (Genişlik / 8) * 3
X2: (Genişlik / 8) * 5;

Şekil 3.6: Kenar Sınır Belirleme Formülü

Şekil.3.6' da belirtilen formül ile x ve y ekseninde kenarlarda bulunan en küçük siyah nokta değerine göre karakter kenarları belirlenir. Siyah değeri bulunmayan alanlar da karakter sınırlarını belirler. Plakanın sol başından ve sağ başından belirlenen üst ve alt sınır noktalarına birleşik olan bölgeler, dikey tarama yapılarak silinir.

Karakter belirlenme işlemi yaparken karakterin en ve boy oranı vardır. Bundan önce yapılmış ve denenmiş çalışmalara göre bir karakterin eni minimum 5 piksel yüksekliği ise minimum 10 pikseldir. En ve boy altında bulunan her görüntü parazit olarak kabul edilmektedir. Şekil 3.7 üzerinde karakter olarak tanınmış plaka örneği bulunmaktadır.



Şekil 3.7: Karakter Ayırma İşlemi

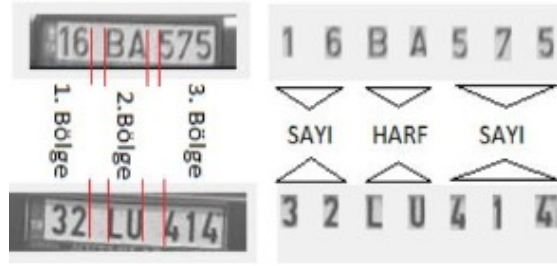
3.2.3. Bölgelerin Belirlenmesi

Ayrıştırılan plakalar üzerinde karakterleri daha hızlı bir şekilde tanımak için plaka kodlaması üzerinde bulunan harf ve karakterleri daha önceden tanımlamak doğru sonuç alınması açısından fayda sağlayacaktır. Türkiye’de kullanılan plaka şablonu Tablo 1 de gösterilmiştir.

7 Karakterli	8 Karakterli
SSHSSS	SSHSSSS
SSHHSS	SSHHSSS
SSHHSS	SSHHSSS

Tablo 1: Türkiye Plakaları Şablonu

Plaka şablonuna bakıldığı zaman plaka bölgesinin 3 bölümden oluştuğu görülebilir. Birinci bölge illeri ifade etmekte ve sayılardan oluşmaktadır. İkinci bölüm ise harflerden oluşmaktadır. Üçüncü bölüm ise sayılardan oluşmaktadır. Plaka üzerinde bulunan üç bölge arasında da karakterler arasında bulunan boşluktan daha çok boşluk bulunmaktadır. Bu yüzden plaka üzerinde kalan boşluklar hesaplama işlemi yapılır. Bulunan boşluklara göre plaka üzerinde bölgelendirme işlemi yapılmaktadır. Bölge işlemi bulunmasıyla birlikte hangi bölgenin karakter hangi bölgenin ise sayı olduğu ortaya çıkmış olacaktır. Karakter ayırma işlemi bir plaka üzerine uyguladığı zaman ayrılacak bölgeler belirlenmiştir. Örnek olarak şekil 3.8 da gösterilmektedir.



Şekil 3.8: Bölgelerin Belirlenmesi

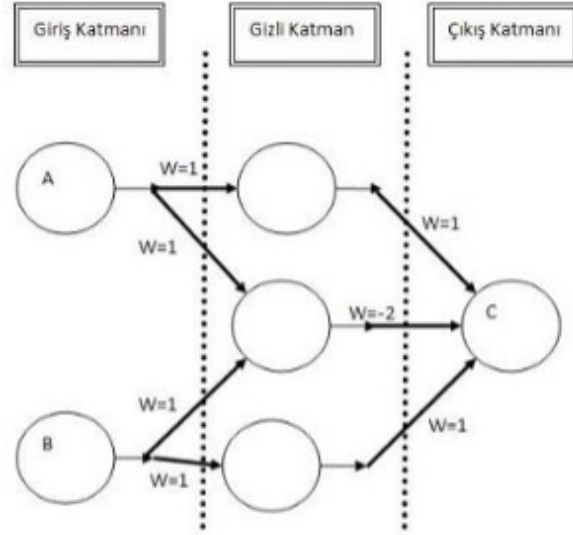
3.2.4. Şablon Eşleştirme Yöntemi

Şablon eşleştirme yöntemi ile karakter tanıma işlemi her bir karakterin piksel değerlerinin önceden programa tanıtılması gerekir. Plaka üzerinde bulunan karakterler bu şablondaki karakterlerin karşılaştırılması ile yapılır. Karakter tanıma işlemine başlamadan önce şablon çıkarılmalıdır. Şablon eşleştirme yönteminin doğru bir sonuç vermesi için karakterlerin her biri sabit bir boyuta getirilmesi gerekmektedir. Sabit olarak kullanılan karakter boyutu 15x32'dir. Boyutlandırılmış karakterlerin yüksekliği 32 piksel genişliği ise 15 piksel olur. Önceki çalışmalar sonucu en yoğun boyut olarak bu boyutlar belirlenmiştir.

Şablon eşleştirme yöntemi görüntü üzerindeki tüm pikselleri dolaşarak önceden tanıtmış olduğumuz karakterlerin tıpatıp aynısını bulmaya çalışır. Bu yöntemde karakter üzerinde bozukluk çok az bile olsa işlemin başarısız olma oranı çok yüksektir.

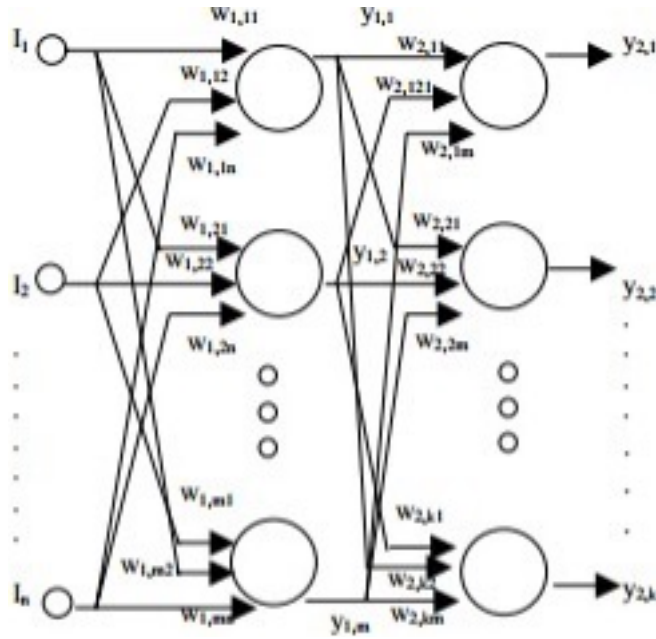
3.2.5. Yapay Sinir Ağları Yöntemi

Yapay sinir ağları karakterleri tanıma işleminde son zamanlarda sıkça kullanılan bir tekniktir. Yapay sinir ağlarında ana amaç insan beyninden yararlanılarak bir olayı öğrenebilmektir. Yapay sinir ağları farklı bir hesaplama yöntemi ile bulunduğu ortama uyum sağlayan, yetersiz bilgi ile karar verebilen bir sistemdir. Öğrenme işlemi tamamlandığında giriş farklı bir giriş olsa bile doğru bir cevap verebilecek yapıya gelir. İnsan beyninin öğrenme, ilişkilendirme, sınıflandırma, genelleme, ve özellik belirleme gibi konularda uygulanmaktadır. Yapay sinir ağlarında uygulamaya girdiler ve çıktılar verilir. Programa problemin nasıl çözüleceği öğretilir. Yapay sinir ağları üç kısımdan oluşmaktadır. Yapay sinir ağlarının yapısı şekil 3.9 de gösterilmiştir.



Şekil 3.9: Yapay Sinir Ağları

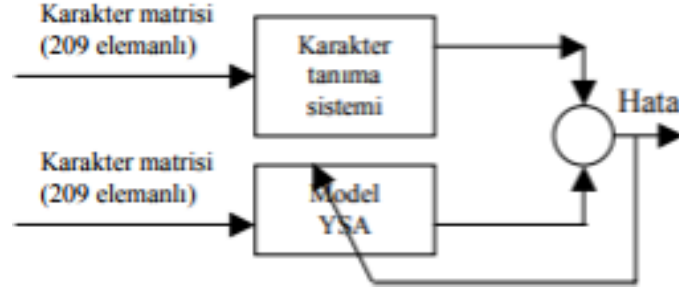
Çok katmanlı bir ileri besleme ağında katmanlar giriş katmanından çıkış katmanına doğru $L=0, L=1, L=2$ olarak belirlenmiştir. Çok katmanlı ileri beslemeli ağ yapısı ise şekil 3.10 deki gibidir



Şekil 3.10: İleri Beslemeli Ağ

Görüntü işleme teknikleri kullanarak plaka bölgesi gerekli morfolojik işlemlerden geçirilmesi ile siyah beyaz bir görüntü haline getirilir. Siyah beyaz bir görüntü 0 ve 1 den oluşmaktadır. Belirlenen bir eşik seviyesinden küçük olanlar 0, büyük olanlar ise 1 olarak belirlenmektedir. Plaka üzerinde bulunan nümerik ve alfa nümerik

Plaka üzerinde karşılaşılabilecek karakterler matrisler halinde elde edildikten sonra sonra yapay sinir ağı kullanarak tanınabilmesi için YSA modeli oluşturulur[8][9]. Bu model dosyasında tüm karakterler için girdi bilgileri bulunmaktadır. Model dosyasındaki veriler kullanılarak ağı eğitimi işlemi yapılmaktadır. Şekil 4.1 de yapay sinir ağlarında ağı eğiterek karakter tanıma işlemi gösterilmiştir.



Şekil 3.13: Yapay sinir ağları eğitimi

Yukarıda tanımlanan giriş, çıkış, karakter tanımlama işlemlerini gerçekleştirdikten sonra eğitim işlemi sonrası vermiş olduğumuz bir karakteri başarı ile tanıyabilir.

Plaka üzerinde bulunan karakterler parçalama işlemlerinden sonra ileri beslemeli yapay sinir ağları uygulanmasıyla karakterleri tanıyabilir. Plaka üzerindeki tüm karakterler bitene kadar bu işlemleri devam ettirir. Bu işlem sonucunda bir araç üzerindeki plakayı tanıma işlemi başarıyla gerçekleştirilir [7].

Bölüm 4

4. SONUÇ

Yapılan bu uygulamada plaka tanıma sisteminde kullanılan algoritma 3 aşamadan oluşmaktadır. Birinci aşamada alınan görüntü üzerinde plakanın koordinatlarının bulunması, ikinci aşamada bulunan plakanın üzerindeki karakterlerin parçalanması ve son aşamada ise elde edilen karakterlerin tanıma işlemi yapılmaktadır.

Plaka sistemleri üzerine yapılan çalışmaların genelinde hazır resim filtreleme yöntemleri kullanılarak plaka yeri tespit edilmiştir. Bu filtreler rgb uzayında bulunan bir görüntüyü griye çevirme, medyan alma, bileteral filtre uygulama, histogram eşitleme, kenar bulma filtresi, eşik değeri belirleme, kontur işlemi, maskeleme işlemi gibi filtrelerdir. Bu projede kullanılan filtreler plaka tanımlama işlemi hazırlanmış filtreler değildir. OpenCV kütüphanesi içinde hazır bulunan filtrelerdir. Gerçek zamanlı bir plaka tanıma sisteminde kameradan alınan görüntüde ilk amaç hızlı bir şekilde plaka yerini tespit etmektir. Bunun için tüm görüntüyü genel olarak kullanılan opencv filtrelerinden geçirmek, ve görüntü üzerinde komşu pikselleri incelemek plaka tanıma işlemi için uzun süre almaktadır. Plaka tanıma sisteminde ana amaç plaka bölgesini tespit etmek olduğu için tüm görüntü üzerinde işlem yapmak süreyi uzatmaktadır. Çalışma yapılırken filtreler uygulandığında önceki filtrelerle kontrol ederek görüntü üzerinde bozulma olup olmadığı kontrolleri yapılmıştır. Görüntü üzerinde piksel çıkarma işlemleri yapılarak bozulmalar kontrol edilmiştir. Bu sayede plaka yerinin tespiti işleminin hatasız bir biçimde yapılması için gerekli önlemler alınmıştır. Gece alınan görüntü üzerinde çalışma işlemi gerçekleştirilmemiştir. Çözünürlük oranı arttıkça plaka bölgesini bulma işlemindeki başarı oranı doğrusal olarak artış göstermiştir.

Karakter tanıma işlemlerinde şablon eşleştirme ve yapay sinir ağları bulunmaktadır. Şablon eşleştirme yönteminde adından anlaşılacağı gibi önceden bir şablon oluşturmak gerekir. Elimizde bulunan şablonla plaka üzerinde plaka üzerinde kontrol işlemi

gerçekleştirilir. Şablon eşleştirme yönteminin dezavantajı eğer plaka üzerindeki bir karakter hafif bir bozulma yaşamışsa şablon eşleştirme metodu ile karakterleri tanıma işlemi başarısız olacaktır. Şablon eşleştirme yöntemi plaka tanıma sistemi gibi yüksek derecede doğruluk bilgisi gereken yerlerde kullanılması sorun yaşatabilir. Şablon eşleştirme yerine yapay sinir ağı yöntemi kullanarak daha yüksek oranda başarı oranına ulaşılabilir.

Kaynaklar

- [1] Convolutional Neural Networks (Evrşimsel Sinir Ağları) [İnternet]; 2019 [erişim tarihi 10.01.2023]. <https://medium.com/@rabiakumus96/convolutional-neural-networks-evri%C5%9Fimsel-sinir-a%C4%9Flar%C4%B1-cceb887a2979>
- [2] Tekrarlayan Sinir Ağı – Recurrent Neural Networks(RNN) [İnternet];2020 [Erişim tarihi 10.01.2023] <https://medium.com/@batincangurbuz/tekrarlayan-sinir-a%C4%9F%C4%B1-recurrent-neural-networks-rnn-17b517dd0b3e>
- [3] Uzun Kısa Dönemli Bellek – LTSM [İnternet];2020 [Erişim Tarihi:10.01.2023] <https://medium.com/@batincangurbuz/uzun-k%C4%B1sa-d%C3%B6nemli-bellek-long-short-term-memory-lstm-c526980c28b1>
- [4] Ağgöl, Burak. DERİN ÖĞRENME KULLANILARAK SAHTE PLAKALI ARAÇ TESPİT SİSTEMİ GELİŞTİRİLMESİ (yüksek lisans tezi). İstanbul: Sabahattin Zaim Üniversitesi; 2021. <https://tez.yok.gov.tr/>
- [5] Boltzman Makinesi [İnternet]; 2020 [Erişim Tarihi: 10.01.2023] <https://medium.com/@batincangurbuz/boltzman-makinesi-boltzmann-machines-bm-ve-k%C4%B1s%C4%B1tlanm%C4%B1%C5%9F-boltzmann-makineleri-restricted-boltzmann-6f2e0f839b05>
- [6] Doğan F, Türkoğlu İ. Derin Öğrenme Modelleri ve Uygulama Alanlarına İlişkin Bir Derleme. DÜMF Mühendislik Dergisi 10:2 (2019) : 409-445
- [7] Plaka Id Tanıma Sistemi [İnternet]; 2018 [Erişim Tarihi: 10.01.2023] https://github.com/ertugruldeniz/plakatanimasistemi/blob/master/ErtugrulDeniz_Plaka_Tanima_Sistemi.pdf
- [8] Canny Kenar Algılama [İnternet]; [Erişim Tarihi: 10.01.2023] <http://mavienginberk.blogspot.com/2017/06/opencv-dersleri-ders16-canny-kenar.html>

Özgeçmiş

Adı Soyadı: Efe Önier
E-mail (1): y210234002@ogr.ikc.edu.tr
E-mail (2): efe_onier@windowslive.com

Eğitim:

2008–2010 Süleyman Demirel Üniversitesi, Bilgisayar Programlama
2011–2019 Anadolu Üniversitesi. İşletme ve Yönetim
2021–2022 Katip Çelebi Üniversitesi; Yazılım Müh.Tezsiz Yüksek Lisans

İş Deneyimi:

2022 – Devam Sendeo A.Ş
2021 – 2022 GKG Grup
2020 – 2021 DefineX Consulting
2016 – 2020 Koçsistem
2014 – 2016 Probel Yazılım
2014 – 2014 Netle Yazılım
2012 – 2014 MaviPiksel Yazılım