Package inputenc Error: Invalid UTF-8 byte "F6See the inputenc package documentation for explanation.The document does not appear to be in UTF-8 encoding.Try adding as the first line of the fileor specify an encoding such as [latin1]inputencin the document preamble.Alternatively, save the file in UTF-8 using your editor or another toolrntileme1 Package inputenc Error: Invalid UTF-8 byte "9DSee the inputenc package documentation for explanation.The document does not appear to be in UTF-8 encoding.Try adding as the first line of the fileor specify an encoding such as [latin1]inputencin the document preamble.Alternatively, save the file in UTF-8 using your editor or another toolTE21

PAMUKKALE ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ

OTOPARKLARDA PLAKA TANIMA SİSTEMİ İLE PARK ÜCRETİ HESABI

LİSANS TEZİ

Seval Naz KARAHAN
(17253510)

Bilgisayar Mühendisliği

Tez Danışmanı: Doç. Dr.Öğr.Üyesi Tufan Turacı

HAZİRAN 2022

Pamukkale Üniversitesi Bilgisayar Mühendisliği Bölümü, 17253510 numaralı Lisans Öğrencisi Seval Naz KARAHAN, ilgili yönetmeliklerin belirlediği gerekli tüm şartları yerine getirdikten sonra hazırladığı "OTOPARKLARDA PLAKA TANIMA SİSTEMİ İLE PARK ÜCRETİ HESABI" başlıklı tezini aşağıdaki imzaları olan jüri önünde başarı ile sunmuştur.

Tez Danışmanı : **Doç. Dr.Öğr.Üyesi Tufan Turacı** .....

Jüri Üyeleri : **Prof. Dr. Serdar İPLİKÇİ** .....
Pamukkale Üniversitesi

Öğr. Gör. Dr. Gökhan UÇKAN .....
Pamukkale Üniversitesi

Yük. Müh. Vasfi TATAROĞLU .....
Pamukkale Üniversitesi

Teslim Tarihi : **13 HAZİRAN 2022**
Savunma Tarihi : **31 MAYIS 2022**

Önsöz

Kendisinden tez almam vasıtasıyla tanışıp ,takıldığım her noktada yardımını esirgemeyen başta Doç.Dr.Öğr.Üyesi Tufan Turacı hocamıza, üniversitenin ilk senesinden beri tüm soru işaretlerimi gidermeme yardımcı olup her mailime aynı gün içerisinde dönüş yapan Öğr. Gör. Şevket Umut ÇAKIR hocamıza, sonrasında da aileme ve arkadaşlarıma gönülden teşekkürlerimi sunuyorum.

Haziran 2022 Seval Naz KARAHAN

İçindekiler

Sayfa

ÖNSÖZ ..... **v**
İÇİNDEKİLER ..... **vii**
KISALTMALAR ..... **xi**
ÇİZELGE LİSTESİ ..... **xiii**
ŞEKİL LİSTESİ ..... **xv**
ÖZET ..... **xvii**
SUMMARY ..... **xix**
1. GİRİŞ **1**
1.1 Tezin Amacı 1
1.2 Görüntü İşleme Neden Önemlidir? 2
1.3 Görüntü İşlemenin Kullanım Alanları Nelerdir? 2
1.3.1 Görüntü İyileştirme 3
1.3.2 Cisim Tanıma 3
1.3.3 Sağlık Sektörü 3
1.3.4 Savunma Sanayi 3
1.3.5 Diğer Kullanım Alanları 4
1.4 LİTERATÜR TARAMASI 5
2. KULLANILABİLECEK TEKNOLOJİLER **9**
2.1 PYTHON GÖRÜNTÜ İŞLEME KÜTÜPHANELERİ 9
2.1.1 Numpy Kütüphanesi 9
2.1.2 Matplotlib Kütüphanesi 9
2.1.3 PIL(Python Imaging Library) 10
2.1.4 OpenCV Kütüphanesi 10
2.2 GEOMETRİK DÖNÜŞÜMLER 10
2.2.0.1 Bilineer İnterpolasyon 11
2.2.0.2 Affin Dönüşümü 11
2.2.0.3 Öteleme 11
2.2.0.4 Döndürme 11
2.2.0.5 Aynalama 11
2.2.0.6 Ölçeklendirme 11
2.3 HİSTOGRAM EŞİTLEME 11
2.4 RENK UZAYLARI 12
2.4.1 HSV Renk Uzayı 12
2.4.2 RGB Renk Uzayı 12
2.5 GÖRÜNTÜ İŞLEMEDE KULLANILAN TEMEL FİLTRELER 12
2.5.1 Ortalama (Mean) Filtreleme Yöntemi 12

2.5.2 Medyan Filtreleme Yöntemi ... 13
2.5.3 Gaussian Filtreleme Yöntemi ... 13
2.5.4 Sobel Filtreleme Yöntemi ... 13
2.5.5 Laplasian Filtreleme Yöntemi ... 13
2.5.6 Konservatif Filtreleme Yöntemi ... 13
2.5.7 Fourier Dönüşüm Filtreleme Yöntemi ... 14
2.6 EŞİKLEME İŞLEMLERİ ... 14
2.6.1 Statik Eşikleme ... 14
2.6.2 Dinamik Eşikleme ... 14
2.7 MORFOLOJİ İŞLEMLERİ ... 14
2.7.1 Çekirdek ... 14
2.7.2 Erozyon ... 15
2.7.3 Dilation-Genişleme ... 15
2.7.4 Opening-Açılış ... 15
2.7.5 Closing-Kapanış ... 15
2.8 PLAKA TANIMA ADIMLARI ... 15
2.8.1 Görüntü Üzerinden Plakanın Ayrıştırılması ... 16
2.8.2 Görüntü Ön İşleme ... 16
2.8.2.1 Rgb'den gri seviyeli görüntüye dönüştürme işlemi ... 16
2.8.2.2 Gri seviyeli resim üzerinde gürültü temizleme ... 17
2.8.2.3 Histogram Eşitleme İşlemi ... 17
2.8.2.4 Morfolojik İşleme İşlemi ... 17
2.8.2.5 Pixel Çıkarma İşlemi ... 18
2.8.2.6 Görüntü Eşikleme İşlemi ... 18
2.8.2.7 Kenar algılama işlemi ... 19
2.8.2.8 Genişletme İşlemi ... 20
2.8.2.9 Görüntü üzerine kontur uygulanması ... 20
2.8.2.10 Plaka bölgesinin maskelenmesi ... 21
2.9 KARAKTERLERİN PLAKA BÖLGESİNDEN AYRIŞTIRILMASI ... 21
2.9.1 Plaka üzerinde eğimin düzeltilmesi ... 22
2.9.2 Plaka üzerindeki karakterleri parçalama işlemi ... 23
2.9.3 Karakterleri tanıma işlemi ... 23
2.9.4 Şablon eşleştirme yöntemi ... 24
2.9.5 Yapay sinir ağları yöntemi ... 25
2.10 NEDEN SQLITE VERİTABANI KULLANIYORUM? ... 25
2.10.1 Veritabanı Nedir? ... 25
2.10.2 Neden SQLite ? ... 25
2.10.3 SQLite Python Kodlarına Nasıl Dahil Edilir? ... 26
2.11 PROJE KODLARI VE ÇIKTILARINA AÇIKLAMALI ŞEKİLDE YER VERİLMESİ ... 31
2.11.1 Kullanılacak Kütüphanelerin dahil edilmesi ... 31
2.12 PROJEDEN ELDE EDİLEN ÇIKTILARIMIZ ... 33
2.12.1 Projenin Çalıştırılması ... 33
2.12.2 Kullanıcıdan Resim Inputunun Alınması ... 34
2.12.3 Kullanıcıdan Alınan Resim Inputuna Göre Plaka Tanımlamasının Yapılması ... 34

2.12.4Kullanıcıdan Konum Bilgisinin Alınması 34
2.12.5Aracın Otoparktan Ayrılması 34
3. HİPOTEZ 37
3.1 Tezin Hipotezi 37
KAYNAKLAR..... 41
ÖZGEÇMİŞ 43

KISALTMALAR

RGB **:** Red , Green , Blue
CSV **:** Hue, Saturation ve Value

Tablo Listesi

Sayfa

Şekil Listesi

Sayfa

Şekil 1.1 : Görüntü işleme ile Araç plakası tanıma 2
Şekil 2.1 : RGB renk uzayında plaka görünümü 17
Şekil 2.2 : Gri seviyeye dönüştürülmüş plaka..... 17
Şekil 2.3 : Histogram eşitleme işleminden geçmiş plaka görüntüsü 18
Şekil 2.4 : Morfolojik işlem sonrası meydana gelen plaka resmi 18
Şekil 2.5 : Pixel çıkarma işlemi uygulandıktan sonra elde edilen plaka görüntüsü..... 19
Şekil 2.6 : Görüntü eşikleme işlemi yapılmış olan plaka..... 19
Şekil 2.7 : Kenarları belirgin hale getiren plaka görüntüsü 20
Şekil 2.8 : Genişletme işlemi yapıldıktan sonra ortaya çıkanyeni görüntü..... 20
Şekil 2.9 : Kontur işlemi plaka görüntüsü..... 21
Şekil 2.10 : Kontur ile plaka bölgesinim bulunması ve etrafının çerçevelenmesi 21
Şekil 2.11 : :Bulunan plaka yerinin fotoğraftan kırpılması işlemi 22
Şekil 2.12 : Plaka üzerinde tespit edilen eğimin örnek görüntüsü..... 23
Şekil 2.13 : Plaka üzerinde karakter ayrıştırması yapılması..... 24
Şekil 2.14 : Karakter tanıma işlemi yapılan plaka görüntüsü 24
Şekil 2.15 : Yapay sinir ağları ile karakter tanıma blok diyagramı..... 25
Şekil 2.16 : Python'a Sqlite bağlantısı kurma kodu..... 26
Şekil 2.17 : Python ile Sqlite içerisinde bir tablo oluşturma kodu..... 27
Şekil 2.18 : Python ile Sqlite içerisinde bir tabloya veri kaydı ekleme kodu 28
Şekil 2.19 : Python ile Sqlite içerisinde bir tablodan veri kayıtlarını çekme kodu 29
Şekil 2.20 : Python ile Sqlite içerisinde bir tablodan veri kayıtlarını güncelleme kodu..... 29
Şekil 2.21 : Python ile Sqlite içerisinde bir tablodan veri kayıtlarını temizleme kodu..... 30
Şekil 2.22 : Proje içerisine kütüphanalerin import edilmesi işlemi 31
Şekil 2.23 : Kullacıdan alınan resmin işlenme aşamaları 31
Şekil 2.24 : Plakanın araç resmi içeriisnden kırpılması..... 32
Şekil 2.25 : Araç resminde bulunan plaka bölgesinin dikdörtgen içerisinde gösterilmesi..... 32
Şekil 2.26 : Sqlite veritabanına bağlantı sağlama kodları..... 33
Şekil 2.27 : Plakası tanımlanan aracın konumunun belirlenmesi 33
Şekil 2.28 : Projenin çalıştırılması için gerekli terminal komutumuz 34
Şekil 2.29 : Kullanıcıdan istenen resim inputu 34
Şekil 2.30 : Kullanıcıdan istenen resim bilgisine göre plakanın işlenmesi 35

Şekil 2.31 : Kullanıcıdan tanımlanmış plakaya ait aracın konum bilgisinin alınması 35
Şekil 2.32 : Aracın otoparktan ayrılması 36

OTOPARKLARDA
PLAKA TANIMA SİSTEMİ İLE
PARK ÜCRETİ HESABI

Özet

Bu projede araç giriş-çıkış bilgileri ve konumları görüntü işleme kütüphaneleri kullanılarak araçların plakalarına bağlı olarak hafızada tutulmaktadır. Otopark ücreti, aracın hangi plakasının otoparkın hangi bölümüne park edildiğine ve giriş çıkış saatlerine göre hesaplanır. Peki bu plakaları nasıl hafızada tutmayı hedefliyorum?
Öncelikle her aracın bir plakası vardır. Plaka tanıma, Python görüntü işleme kitaplıkları yardımıyla gerçekleştirilir. Otoparktaki her aracın konumu plakaları ile eşleştirilir ve SQLITE veri tabanında saklanır. Aynı şekilde plakası tanımlanan aracın otoparka giriş saati de veritabanımızdaki kayıtlarda tutulmaktadır. Plaka tanımlandıktan sonra kullanıcıdan otoparkta bir konum girmesi istenir. Bu konumda zaten bir araç varsa, geri bildirim olarak araç için başka bir konum seçilmesi için bir mesaj gönderilir.
Doğru konum bilgisi girildikten sonra aracımızın otoparka girişinin tam saati ve dakikası bilgisayara göre veri tabanına kaydedilir.
Daha sonra aynı plaka ikinci kez okunmak isteniyorsa bu araç zaten veri tabanında kayıtlı yani otoparkta mevcut ve artık otopark yapılması gerekiyor.
Bu kısımda bilgisayara göre kesin çıkış saati ve dakikası hesaplanır ve buna göre giriş zamanı çıkarılır. Kalan saatlerimiz ücret ile çarpılır ve kullanıcının ödemesi gereken park ücreti olarak geri döner.
Aracımız otoparktan ayrıldıysa kaydı da veri tabanından silinmelidir.

Anahtar Sözcükler: Python,Görüntü İşleme,Görüntü Restorasyonu,Yüksek Çözünürlük, Karakter tanıma

CALCULATION OF PARKING FEE
WİTH PLATE RECOGNITION SYSTEM
IN PARKING AREAS

SUMMARY

In this project, vehicle entry-exit information and locations are kept in memory, depending on the license plates of the vehicles, using image processing libraries. The parking fee is calculated according to which plate of the vehicle is parked in which part of the parking lot and entry and exit times. So how do I aim to keep these plates in memory?
First of all, every vehicle has a license plate. License plate recognition is performed with the help of Python image processing libraries. The locations of each vehicle in the parking lot are matched with their license plates and stored in the SQLITE database. Likewise, the entrance time of the vehicle whose license plate is defined to the parking lot is also kept in the records in our database. After the license plate is defined, the user is asked to enter a location in the parking lot. If a vehicle already exists at this location, a message is sent to select another location for the vehicle as a feedback.
After the correct location information is entered, the exact time and minutes of our vehicle's entrance to the parking lot are recorded in the database according to the computer.
Later, if the same plate is wanted to be read a second time, this means that this vehicle is already registered in the database, that is, it is present in the parking lot, and now the parking lot must be done.
In this part, the exact exit hour and minute are calculated according to the computer, and accordingly the entry time is subtracted. Our remaining hours are multiplied by the fee and return as the parking fee that the user has to pay.
If our vehicle has left the parking lot, its registration should also be deleted from the database.

Keywords: Python,Image Processing,Image Restoration,High Definition, Character recognition

1. GİRİŞ

Python Görüntü İşleme kütüphaneleri ile gerçek zamanlı görüntü işlemeleri yapılabileceği gibi analizleri de gerçekleştirilir.

Görüntü işleme, genellikle estetik bir standart elde etmek veya tercih edilen bir gerçekliği desteklemek için bir görüntüyü keyfi olarak manipüle etmek olarak görülür. Bununla birlikte, görüntü işleme, insan görsel sistemi ile dijital görüntüleme cihazları arasında bir çeviri aracı olarak daha doğru bir şekilde tanımlanır.

İnsan görsel sistemi, ek gürültü ve bant genişliği kısıtlamaları getiren görüntüleme cihazlarıyla dünyayı dijital dedektörlerle aynı şekilde algılamaz. İnsan ve dijital detektörler arasındaki göze çarpan farklar, çeviriyi gerçekleştirmek için bazı temel işlem adımlarıyla birlikte gösterilecektir.

Görüntü işlemeye bilimsel yöntemle tutarlı bir şekilde yaklaşılmalıdır, böylece diğerleri yeniden üretebilir ve birinin sonuçlarını doğrulayabilir. Buna, işleme eylemlerinin kaydedilmesi ve raporlanması ve benzer tedavilerin yeterli kontrol görüntülerine uygulanması dahildir.

Böylelikle elde edilen görüntünün anlaşılabilirliği ,verimliliği ve çözünürlüğü en üst kalitede çıktı olarak elimize geçer. Görüntü işleme kütüphanelerinin izlediği yöntemler ve algoritmalarının taşıtlar üzerinde kullanılıp plaka tanımlamalarının yapılabilmesi hedeflenir.

Plakalar tanımlandıktan sonra proje içerisinde hedeflenilen, araçların otopark ücretleri ve otoparktaki konum bilgilerine erişilir.

1.1 Tezin Amacı

Python Görüntü İşleme Kütüphaneleri aracılığıyla bir otoparka giriş yapacak olan taşıtların gerçek zamanlı plaka görüntüleri saptanır ve bellekte tutulur. Hafızada tutulan taşıt plakalarına göre, her taşıtın park yeri konum bilgilendirmesi elde edilir. Otoparkta taşıtların kalma sürelerine ilişkin veriler, plakalarına bağlı olarak bellekte saklanır.Böylelikle her plakaya ait otopark ücreti ve otoparkta hangi konuma park

Şekil 1.1 : Görüntü işleme ile Araç plakası tanıma

edildiğini otomatik olarak öğrenebilmek amaçlanır.Otoparkı bulunan işletmelerde park ücretinin sistem üzerinden otomatik olarak hesaplanması hedeflenir. Hayatın günlük karmaşasında insanoğlu unutkanlık yaşayabileceği gibi aracını park ettiği katı veya bloğu otopark içerisinde unutabilir. Bu projede aracımızın otopark konum bilgisi kolaylıkla elde edilir.

1.2 Görüntü İşleme Neden Önemlidir?

[**?**]Görüntü İşleme, elimizde bulunan görüntüden anlamlı ifadeler çıkarmamıza yarayan işlemler bütünüdür. Bu işlemler, görüntüyü oluşturan pikseller üzerinde gerçekleştirilecek matematiksel işlemler sayesinde gerçekleştirilir. Görüntü elde edildikten sonra, yapılması istenen göreve göre bir algoritma tasarlanır ve görüntü bu aşamalardan geçerek istenen görevi yerine getirir. Görüntü işleme teknikleri kullanılarak yapılabilecek işlemlere göz atacak olursak şunları görebiliriz;

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

1.3 Görüntü İşlemenin Kullanım Alanları Nelerdir?

1.3.1 Görüntü İyileştirme

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

1.3.2 Cisim Tanıma

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

1.3.3 Sağlık Sektörü

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

1.3.4 Savunma Sanayi

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

1.3.5 Diğer Kullanım Alanları

Diğer alanlardan bazıları ise şu şekildedir; Uydu görüntüleri üzerinden nüfus yoğunluğu, çevre kirliliği gibi çevresel durumların tespiti, Hava Gözlem Ve Tahmin, Güvenlik Sistemleri, Kriminal Laboratuvarlar, Uzaktan Algılama Sistemleri.

1.4 LİTERATÜR TARAMASI

Anıl Güngördü - Araç Plaka Tanıma Sistemi(Verileri tanıtmak için bir xml dosyası oluşturmuştur.Bu xml dosyası 3 exe den meydana gelmektedir.Annotation.exe, createsamples.exe, traincascade.exe'dir. Verileri eğitme aşamaları anlatılmaktadır.)

Nisanur Bulut - Plaka Tanıma (Bu projede benim projemden farklı olarak araç plakası tanınması sırasında Aforge Kütüphanesi ve Damla Filtreleme İşlemi uygulanmıştır.)

Neeramitra Reddy - Digital Image Processing(Dijital görüntü işleme aşamalarından, bu aşamaların gerekliliklerinden, dijital görüntü işleme kavramının ne olduğundan , faydalarından ve kullanım alanlarından bahsedilmiştir.)

Rohit Kumar Jena -Automatic Watermark Detection (Dijital görüntü işleme projesidir. Otomatik filigran algılama anlamına gelir.Projede dijital damgalama yaşam döngüsü aşamaları kullanılmıştır.)

Data Carpentry - Image Processing(Skimage'de görüntü temsiline, çizim ve bitsel işlemlere, histogram oluşturmaya, resimleri bulanıklaştırmaya,eşik işlemlerine ve son olarak da bağlı bileşen analizine dair bilgilere yer verilmiştir.)

Muhammed Mücahid Enes YÜCEL- Görüntü İşleme ile Araç Takibi (Python'da OpenCv kütüphanesi içerisinden çekilen fonksiyonlar kullanılmıştır. Uygulamada görüntü işleme teknikleri ile araç takibi yapıp sonrasında araç sayısı gösterilerek proje gerçekleştiriliyor.)

Bhanu Prakash Poluparthi - Image Processing with Python (Python görüntü işleme kütüphaneleri kullanılmıştır.Görüntü işlemenin ek olarak; komşuluk işlemesine(neighbourhood processing) , segmentasyon analizine, çekirdek dışı görüntü işlemesine , renk manipülasyonuna değinilmiştir.)

Hasaan Majeed-Number Plate Recognition System Using OpenCV (Python'da

OpenCV kullanılarak yapılan araç plaka tanıma sistemidir. Görüntüden olduğu kadar videodan da plaka algılamak için kullanılabilecek bir projedir.)

Barroso J., Rafael A.,. Dagless E. L and BulasCruz J.-Number plate reading using computer vision, IEEE Barroso J., Rafael A.,Dagless E. L and BulasCruz J.(Bir plaka tanıma sistemi projesi yaparak görüntüdeki plaka alanının konumu, karakterlerin segmentasyonu ve karakterlerin tanımlanması gereçekleştirilir.)

Gonzalez R. and Woods R. - Morphology-based License Plate Detection from Complex Scenes(Bu proje, karmaşık sahnelerden renkli kenar kullanan bir plaka algılama algoritması sunmaktadır. Önerilen yöntem dört adıma ayrılabilir:orijinal görüntüden dikey renk kenarı elde etmek, yatay adaylar elde etmek,dikey ve yatay renk kenarlarını içeren LP'nin renk kenarını elde etmek,doğru satır ve çizgi konumlarını bulmak.)

Christos-Nikolaos E. Anagnostopoulos; Ioannis E. Anagnostopoulos; Ioannis D. Psoroulas-License Plate Recognition From Still Images and Video Sequences: A Survey (Görüntülerdeki veya videolardaki plaka tanıma (LPR) algoritmaları plaka formatlarının çeşitliliği ve görüntü elde etme sırasındaki eşit olmayan dış aydınlatma koşulları nedeniyle oldukça zordur. Hareketsiz görüntülerde veya video dizilerinde LPR için çok sayıda teknik geliştirilmiştir ve bu makalenin amacı bunları sınıflandırmak ve değerlendirmektir.)

B. Y. Amirgaliyev; C. A. Kenshimov; K. K. Kuatov; M. Z. Kairanbay; Z. Y. Baibatyr - License plate verification method for automatic license plate recognition systems (Otomatik plaka tanıma (ALPR), insan etkileşimi olmadan bir video akışından veya bir görüntüden plaka numaralarının tanımlanması için kullanılan teknolojidir. ALPR, kayıt veya muhasebe (ücretli yollar, kontrol noktaları ve otoparklar), arama ve takip (çalınan araçların kurtarılması, suçluların yakalanması ve trafik kanunlarının diğer düzenlemeleri) vb. gibi birçok uygulamaya sahiptir.)

Zhuang Liu; Yuanping Zhu - Vehicle License Plate Recognition In Complex Scenes (Bu makale, karmaşık arka plan ve plaka eğimi altında plaka tanıma

problemini incelemektedir. Mevcut yöntemler bu sorunları iyi çözemez. Bu makale, derin öğrenmeye dayalı uçtan uca bir düzeltme ağı önermektedir. Model üç bölümden oluşmaktadır: Plaka düzeltme, görüntü özelliği çıkarma ve plaka karakter tanımanın bozulmasından sorumlu olan düzeltme ağı, artık modül ve sıra modülü.)

2. KULLANILABİLECEK TEKNOLOJİLER

2.1 PYTHON GÖRÜNTÜ İŞLEME KÜTÜPHANELERİ

[2] Görüntü İşleme alanında programlama yapmak için günümüzde birçok farklı alternatif bulunmaktadır. En temel düzeyde iki sınıfa ayırmak gerekirse, bunlardan biri yapay zeka destekli (YOLO vb.) görüntü işleme çalışmaları, diğeri de belirli kütüphane ve frameworkleri kullanarak herhangi bir machine learning ya da deep learning tekniği kullanmaksızın kendi algoritmalarımızı geliştirmektir. Python zengin kütüphane ve framework seçenekleri ile adından söz ettirdiği gibi bu alanda da birçok alternatif bulundurmakta. Python Görüntü İşleme Kütüphaneleri arasından başı OpenCv çeker, bununla beraber SimpleCV gibi de frameworkler bulundurmaktadır. OpenCv 'nin yanı sıra Yapay Zeka ve Makine öğrenmesi üzerine büyük avantajları bulunan Python'da Deep Learning tabanlı kendi Image Processing algoritmalarınızı da geliştirebilirsiniz.

2.1.1 Numpy Kütüphanesi

Numpy, bilimsel hesaplama işlemleri,çok boyutlu diziler, çeşitli türetilmiş nesneler dahil olmak üzere diziler üzerinde hızlı işlemler yapılabilmesi için kullanılan önemli kütüphanelerden birisidir. Numpy, veri tipi olarak n boyutlu dizinin tanımlanmasını sağlar ve diziler üzerinde yeniden boyutlandırma, indeksleme gibi temel işlevlerin gerçekleştirilmesinde kullanılır.

2.1.2 Matplotlib Kütüphanesi

Matplotlib, çeşitli basılı formatlarda ve interaktif ortamlarda yüksek kalitede görüntüler üretebilen iki boyutlu çizim kütüphanesidir. Matplotlib, Python komut dosyalarında, Python ve IPython kabuklarında, Jupyter dizüstü bilgisayarında, web uygulama sunucularında ve kullanıcı grafik arabirim araçlarında sıklıkla

kullanılmaktadır.

2.1.3 PIL(Python Imaging Library)

PIL kütüphanesi basit noktasal işlemler, bir dizi yerleşik konvolüsyon çekirdekleri ile filtreleme ve renk alanı dönüşümleri gibi temel görüntü işlevlerini gerçekleştirmek için kullanılan bir kütüphanedir. Python programlama dilinde bu kütüphane kullanılarak görüntü dosyaları üzerinde değiştirme ve dönüştürme işlemleri gerçekleştirilebilmektedir.

2.1.4 OpenCV Kütüphanesi

OpenCV, bilgisayarlı görü uygulamaları ve ticari ürünlerde makine öğrenme algısını hızlandırmak için hazırlanmış bir kütüphanedir. OpenCV, 1999 yılında Intel'de Gary Bradski tarafından dünyadaki bilgisayar görüsünün araştırılması ile ticari uygulamaların hızlandırılması amacıyla grafiksel olarak daha güçlü bilgisayarların oluşturulması için tasarlanmıştır.Açık kaynak kodlu ve platformdan bağımsız bir kütüphanedir.

2.2 GEOMETRİK DÖNÜŞÜMLER

Görüntü işleme sırasında sıklıkla geometrik görüntü dönüşümlerine ihtiyaç duyulmaktadır. Geometrik dönüşümler, pikseller üzerinde işlemler yaparak görüntü döndüren,daraltan veya büyüten fonksiyonlardır.

2.2.0.1 Bilineer İnterpolasyon

Görüntü üzerinde eski piksel değeribaz alınarak farklı bir noktadaki yeniğ pikselin olası değerinin hesaplanması yöntemidir. Başka bir ifadeyle örnekleme olarak ifade edilir.

2.2.0.2 Affin Dönüşümü

•üç temel işlem:Görüntüler üzerinde koordinaqt dönüşümlerine dayalı üç temel işlem olan ölçekleme, döndürme ve öteleme işlemleridir. **•affin dönüşüm bağıntısı**:Birbirine paralel olmayan (dik veya eğik) iki düzlemden biri, diğeri üzerine izdüşümlendirilirse görüntü ve esas şekil arasında bir affin dönüşüm bağıntısı elde edilir.

2.2.0.3 Öteleme

Görüntüyü öteleme işlemi, piksellerin değerleri değişmeden koordinat sistemindeki konumunun değiştirilmesi olarak tanımlanmaktadır.

2.2.0.4 Döndürme

Görüntünün döndürülmesi, görüntüdeki piksellerin belirli bir açıda döndürülmesi ile piksel konumlarının değiştirilmesi işlemidir.

2.2.0.5 Aynalama

Görüntüyü yansıtma işlemi, belirli bir nokta ya da eksen etrafında orjinal görüntüdeki piksellerin yansıtılarak yeni konumlarına yerleştirilmesi işlemidir.

2.2.0.6 Ölçeklendirme

Görüntüyü ölçeklendirme işlemleri, görüntüler üzerinde belirli bir alanda ya da görüntünün tamamında görüntünün büyütülmesi veya küçültülmesi için kullanılmaktadır. Küçültme-Büyültme Piksel Değiştirme Piksel İnterpolasyonu Eğme-Kaydırma

2.3 HİSTOGRAM EŞİTLEME

[3] Histogram eşitleme işlemi bir resim üzerideki piksel değerlerinin arasındaki farkları en aza indirme işlemidir. Bir nevi map fonksiyonu gibi düşünülebilir. Histogram eşitleme işlemi piksel değerlerini sadece kaç adet olduklarını bulmak

için kullanırken adet değerlerinin cdf* ile hesaplanmış hallerinden yeni görüntüyü oluşturur. Bu işlemi gerçekleştirmek için işlem adımları aşağıda verilmiştir;

Her pikselin tekrar sayısını bul(histogram dizisi), Pikselleri Değerlerine göre sırala, Bu değerler üzerinden cdf toplam matrisi oluştur, Her bir cdf değeri için cdf fonksiyonunu işlet, Gerçek resim üzerindeki piksel değerlerini cdf piksel değerlerine göre değiştir.

2.4 RENK UZAYLARI

[4] Renk uzayları, görüntüde oluşan renk içerikleri üç veya dört farklı renk bileşenleriyle birlikte matematiksel model oluşturmaktadır.

2.4.1 HSV Renk Uzayı

[5] HSV, bir silindir şeklinde ifade edilir. Piksel değeri x için, H değeri x' in açısal konumunu ifade eder, S değeri x' in silindirin merkezine uzaklığını, V değeri ise x' in silindir yüzeyine uzaklığını ifade eder.HSV sisteminde H ve S değerlerinin ne olduğuna bakmaksızın eğer V değeri sıfıra eşitse 0 ise ortaya çıkan renk siyah olacaktır.

2.4.2 RGB Renk Uzayı

[6] RGB renk modelinin temel olarak doğada bulunan tüm renklerin yalnızca üç rengin referansı ile oluşturulabilmesine dayanır. Bu üç renk ise kırmızı, yeşil ve mavidir. Daha anlaşılır bir ifadeyle, RGB renkler ışık bazlı olduklarından dijital ekranlarda renkli görüntü elde etmeyi mümkün kılmaktadır. Bu nedenle eğer dijital ekranlarda görüntülenecek bir tasarımınız varsa, kullandığınız tasarım programında mutlaka RGB renkleri seçmeniz gerekmektedir. Bu sayede daha canlı tasarımlar elde edebilirsiniz.

2.5 GÖRÜNTÜ İŞLEMEDE KULLANILAN TEMEL FİLTRELER

2.5.1 Ortalama (Mean) Filtreleme Yöntemi

[7]Ortalama filtresi, görüntüleri yumuşatmanın basit ve uygulanması kolay bir yöntemidir. Diğer bir deyişle, bir piksel ile diğerleri arasındaki değişim miktarını azaltmaktır. Genellikle görüntülerdeki gürültüyü azaltmakiçin kullanılır.Ortalama filtresi, bir görüntünün her bir piksel değerini komşularının ve kendisinin dahil olduğu

ortalama değer ile değiştirmektir. Bu durum, çevresindekileri temsil etmeyen piksel değerlerinin ortadan kalkmasına yol açar. Ortalama filtresi bir konvolüsyon filtresidir. Konvolüsyon filtreleri çekirdek şablon (kernel)temeline dayanır.

2.5.2 Medyan Filtreleme Yöntemi

Medyan filtresi, normal olarak mean filtresi gibi bir resimdeki gürültüyü azaltmak için kullanılır. Ancak resim üzerindeki detayların kaybolmaması noktasında mean filtresinden çok daha iyi sonuç verir. Medyan filtre de mean filtresi gibi her pikselin değerini hesaplamak için yakınındaki komşularına bakar. Medyan filtresinde piksel değeri komşu piksel değerlerinin ortalaması ile değiştirmek yerine (mean filtresi), komşu pikselleri sıralayıp sıranın ortasındaki değeri alır. Eğer incelenen bölge (şablonun içerisi) çift sayıda piksel varsa,orta değer olarak, ortada bulunan iki pikselin ortalaması kullanılır.

2.5.3 Gaussian Filtreleme Yöntemi

Gauss yumuşatma operatörü, görüntüleri 'bulanıklaştırmak', ayrıntı ve gürültüyü ortadan kaldırmak için kullanılan 2 boyutlu konvolüsyon (çekirdek matris ile resim üzerindeki piksellerin çarpımı işlemi) operatörüdür. Bu anlamda, ortalama (Mean) filtreye benzer. Ancak Gauss, "çan şeklindeki" grafikle temsil edilebilecek farklı bir çekirdek şablon (matris) kullanır.

2.5.4 Sobel Filtreleme Yöntemi

Sobel filtre görüntülerdeki objelerin sınırlarını tespit etmek için kullanılan bir filtreleme yöntemidir.Görüntü üzerinde hem yatay hem de dikey olarak kenar belirleme yapılabilmektedir.

2.5.5 Laplasian Filtreleme Yöntemi

Laplacian operatörü, görüntü netleştirme, kenar algılama ve kenar hatlarını belirleme işlemlerinde sıklıkla kullanılmaktadır.

2.5.6 Konservatif Filtreleme Yöntemi

Görüntü üzerindeki gürültüleri azaltmak için kullanılan bir yöntemdir.Gürültü, görüntüdeki ayrıntıların görülme miktarını azaltırken konservatif filtreleme ile bu problem hızlı bir şekilde azaltılarak keskin kenarlar korunmaktadır.

2.5.7 Fourier Dönüşüm Filtreleme Yöntemi

Furier dönüşümünde filtreleme işlemi frekans boyutunda gerçekleşmektedir. Fourier dönüşümü frekans alanındaki görüntüyü ifade ederken, girdi görüntüsü uzaysal alanı ifade etmektedir. Kısacası Fourier bir görüntünün kosinüs ve sinüs bileşenlerini ayırmak için kullanılmaktadır.

2.6 EŞİKLEME İŞLEMLERİ

[8] Eşikleme genel olarak gri tonlu görüntüden ikili görüntü elde etmek için kullanılan bir filtreleme yöntemidir. Eşikleme işlmeleri statik ve dinamik eşitleme olmak üzere iki kategoriye ayrılır.

2.6.1 Statik Eşikleme

Statik eşiklemede ikili tip, ters ikili tip, sıfır tip, ters swıfır tip ve kırpmaq tip olmak üzere beş kategoriye ayrılır.

2.6.2 Dinamik Eşikleme

Otsu algoritması olarak da bilinir. Görüntüdeki piksellerin kümelenmesini piksel değerlerinin dağılımına göre sağlamaktadır. Görüntüdeki piksel sayılarını hesaplamada kullanılan matematiksel ifadeleri mevcuttur.

2.7 MORFOLOJİ İŞLEMLERİ

[9] Bazen, görüntünüzü eşledikten sonra, ikili görüntünüzde istenmeyen parazit olur. Morfolojik işlemler bu gürültüyü görüntüden çıkarmaya yardımcı olabilir.

2.7.1 Çekirdek

Çekirdek, başlangıç noktasının ikili görüntünün 1 değerinin her pikselinin üzerine yerleştirildiği basit bir şekildir. OpenCV, çekirdeği NxN matrisiyle sınırlar, burada N tek sayıdır. Çekirdeğin kökeni merkezdir.

2.7.2 Erozyon

Bilgisayarla görmedeki erozyon, topraktaki erozyona benzer. Ön plandaki nesnelerin sınırlarından uzaklaşır. Bu işlem arka plandaki gürültüyü ortadan kaldırabilir.

2.7.3 Dilation-Genişleme

Genleşme erozyonun tam tersidir. Sınırlardan uzaklaşmak yerine onlara katkıda bulunuyor. Bu işlem, daha geniş bir bölgedeki küçük delikleri kaldırabilir.

2.7.4 Opening-Açılış

Açılma erozyon ve ardından genişlemedir. Bu işlem, daha büyük özelliklerin şeklini etkilemeden gürültüyü ortadan kaldırır.

2.7.5 Closing-Kapanış

Kapatma genişlemedir ve ardından erozyon meydana gelir. Bu işlem, daha büyük unsurların şeklini etkilemeden küçük delikleri veya kırılmaları giderir.

2.8 PLAKA TANIMA ADIMLARI

[10]Plaka tanıma işlemi yapmak için yedi temel algoritma vardır. Bu algoritmalar ile plaka bulma işlemi yapılmaktadır. Bu algoritmaların hassasiyeti plaka bulma işleminde çok fazla etkilemektedir. [**?**] Plaka tanıma işlemlerinde kullanılan algoritmalar şunlardır;

- Plaka yerinin tespiti
- Plakanın sonraki algoritmalara uygun bir biçimde yeniden konumlandırılması ve boyutlandırılması
- Zıtlık ve parlaklık gibi görüntü özelliklerininnormalizasyonu
- Karakter ayırma ile görüntü üzerinden karakterlerinçıkarılması

- Optik karakter tanıma
- Ülkeye özgü kelime dizimi ve kontrol işlemi
- Daha güvenilir bir sonuç elde etmek için tanımlanan değerin birden fazla görüntüde ortalamasının alınması

2.8.1 Görüntü Üzerinden Plakanın Ayrıştırılması

[3] Görüntü üzerindeki plaka kısmının doğru bir şekilde ayrılması oldukça önemlidir. Bu adımda yapılacak bir yanlış sistemin tamamını etkilemektedir. Plakanın görüntü üzerinden ayrılmasında belirli koşullar vardır. Sistem tasarlanırken hava koşulları, parlaklık gibi kriterler önemlidir. Tüm şartlarda üst seviyede çalışabilecek sistem tasarlanması için burada yapılacak işlemler önemlidir. Sistem günün her saatinde çalışması gerektiği için alınan görüntler farklı kontrast değerlerine sahip olacaklardır. [12] İşlemler sırasıyla ilk olarak alınan görüntünün griye çevrilir. Griye çevrilen resim üzerinde gürültü temizleme işlemi yapılır. Görüntü üzerinde histogram eşitleme işleme işlemi yapılır. Morfolojik işlem uygulanır. Histogram eşitleme yapılan görüntüden, morfolojik işlem yapılmış işlem piksel piksel çıkartılır. Görüntü eşikleme işlemi yapılır. Canny edge uygulanır. Genişletme işlemi yapılır. Kontur işlemi gerçekleştirilir.Son olarak plaka bölgesi maskeleme yapılarak görüntü üzerinden ayrıştırılır.

2.8.2 Görüntü Ön İşleme

[13]

2.8.2.1 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

2.8.2.2 Gri seviyeli resim üzerinde gürültü temizleme

Şekil 2.1 : RGB renk uzayında plaka görünümü

Şekil 2.2 : Gri seviyeye dönüştürülmüş plaka

[14] 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 işlemdir. Bileteral filtreleme işlemi kenaraları 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 bileteral filtre kullanılmıştır.

2.8.2.3 Histogram Eşitleme İşlemi

[3] Plaka tanıma sisteminde plakalar tanınırken ortam, ışık, renk gibi değişiklikler olabilir. Sistemin bunun gibi çevresel etkenlerden etkilenmesini azaltmak için görüntü üzerinde histogram eşitleme işlemi yapılmaktadır. Bir görüntünün histogramı, o görüntü hakkında önemli bilgiler verir. Karanlık bir resim grafiğinin düşük gri seviye bölgesine yığılmaktadır. Parlak bir görüntüde ise büyük gri seviyesine yığılmaktadır. Alınan görüntü üzerinde histogram eşitleme işlemi gerçekleştirilerek görünütüyü iyileştirme işlemi gerçekleştirilir.

2.8.2.4 Morfolojik İşleme İşlemi

Şekil 2.3 : Histogram eşitleme işleminden geçmiş plaka görüntüsü


Şekil 2.4 : Morfolojik işlem sonrası meydana gelen plaka resmi

[15] Mofolojik işlem ile görüntü üzerinde aşındırma ve genişletme işlemleri yapılabimektedir. Bu filtre ile görüntü üzerindeki gürültünün giderilmesi ile daha doğru sonuçlar elde edilmektedir. Bu filtre ile görüntü üzerinde açma işlemi gerçekleştirilmiştir. Aşındırma işlemi ile küçük paralar yok edilerek görüntü tekrar genişletilmiştir.

2.8.2.5 Pixel Çıkarma İşlemi

[16] Pixel çıkarma veya görüntü çıkarma işlemi bir görüntünün veya piksellerinin sayısal değerlerinin başka bir görüntüden çıkarılması ile ortaya çıkan bir işlemdir. Bir görüntü üzerinde düzensiz olan bölgeleri dengelemek ve iki resim arasındaki değişiklikleri saptamak için kullanılmaktadır. Plaka tanıma sisteminde amacımız plaka bölgesinin net bir şekilde ortaya konularak , diğer bölgelerin görünürlüğünü azaltmaktır. Histogram eşitleme yapılmış bir resim üzerinden morfolojik işlem yapılan resim çıkartılarak bu adım gerçekleştirilmektedir.

2.8.2.6 Görüntü Eşikleme İşlemi

[14] Görüntü üzerinde belirli bir eşik değeri altında kalan kısımları 0 üstünde kalan kısımları ise 1 yapmak süretiyle ikili bir görüntü oluşturma işlemi yapılmasıdır. Bu sistemde ortam şartları sürekli değişmektedir. Eşik değeri her görüntü için farklı

Şekil 2.5 : Pixel çıkarma işlemi uygulandıktan sonra elde edilen plaka görüntüsü

Şekil 2.6 : Görüntü eşikleme işlemi yapılmış olan plaka

olmaktadır. Elle eşik değeri belirlemek gömülü sistem üzerinde çalışan uygulamalar için mümkün olmayacağı için görüntü üzerinde eşik değeri belirli algoritmalar sayesinde hesaplanmaktadır. Eşik değeri belirlemede Nobuyuki Otsu 1975 yılında yayınladığı yazıda Otsu modelini tanıtmıştır. Otsu modelinde varyans denilen bir değer hesaplanır ve değerin en düşük olduğu indeks döndürülmektedir. Görüntü üzerinde eşik değerini hesaplamak için Otsu modelinden yararlanılmaktadır.Pixel çıkarma işlemi uygulanmış ve plaka bölgesi net bir biçimde ortaya çıkmış resimde görüntü eşikleme işlemi yapılarak plaka bölgesinin daha net olarak ortaya çıkması sağlanmıştır.

2.8.2.7 Kenar algılama işlemi

[17] Canny edge algoritması kenar algılalama için kullanılan bir algoritmadır.Görüntü üzerindeki plaka tanıma işlemi yapmak için kenarlar belirlenmesi gerekmektedir. Canny edge çeşitli görme sistemlerinde kenar algılama uygulaması olarak kullanılmış ve başarılı bir şekilde görüntü üzerindeki kenarları ortaya çıkarmayı başarmıştır. Canny algoritması süreci adımlar halindedir. Canny kenar algılama algoritması düşük hata oranı ile kenarları bulması için gürültüyü gidermek için ilk olarak bileteral filtresi uygulanmıştır. Bileteral filtresi yerine gauss filtreside uygulanabilir.

Şekil 2.7 : Kenarları belirgin hale getiren plaka görüntüsü

Şekil 2.8 : Genişletme işlemi yapıldıktan sonra ortaya çıkanyeni görüntü

Bileteral uygulayarak canny edge ile kenar bulma işlemi daha başarılı bi şekilde gerçekleşecektir. Histogram eşitleme işlemi yapılır. Görüntü eşikleme işlemi yapılır. Canny adımları yapılan görüntüde kenarlar ortaya çıkarılmaya çalışılmıştır.

2.8.2.8 Genişletme İşlemi

Canny edge uygulanan görüntü üzerine uygulanmıştır. Canny edge uygulanarak kenarları yüksek bir oranda algılama işlemi gerçekleştirilmiştir. Kenarları genişletmek için daha belirgin hale getirmek için dilate işlemi yapılır. Genişletme işlemi yapmak için bir çekirdek matris oluşturulur. Bu uygulamada 3x3 lük birler matrisi ile işlemleri gerekleştirilir. Bir merkezi nokta seçilir oradan başlayarak görüntünün tüm pikselleri üzerinde belirlediğimiz iterasyon sayısı kadar bu işlem gerçekleştirilir. Bu sayade görüntüntü üzerinde darbeler ve noktalar bulunur. Maksimum pixel değerini hesaplar ve görüntü üzerinde bulduğu pikseli maksimum pikselle değiştirir. Pikselleri maksimum değerl ile değiştirmek parlak olan bölgelerin genişlemesini sağlar.

2.8.2.9 Görüntü üzerine kontur uygulanması

Genişletme işlemi ile kenarları belirgin bir görüntü elde edilir. Elde edilen görüntü üzerinde gereken işlem kontur uygulayarak aranan plaka bölgesini işaretlemek-tir.Konturlar, aynı renk veya yoğunluğa sahip olan tüm kesintisiz noktaları (sınır

Şekil 2.9 : Kontur işlemi plaka görüntüsü

Şekil 2.10 : Kontur ile plaka bölgesinim bulunması ve etrafının çerçevelenmesi

boyunca) birleştiren bir eğri olarak basitçe açıklanabilir. Konturlar, şekil analizi ve nesne algılama ve tanıma için yararlı bir araçtır.Kontur yapmadan önce konturun doğru sonuç vermesi için canny edge ve genişletme işlemi uygulanır. Sınır noktalarınızın olması koşuluyla herhangi bir şekli çizmek için de kullanılabilir. Buna göre istediğimiz alanları bulmak için bulması gereken parametreler önceden belirlenmiştir. Buna göre görüntü üzerindeki dikdörtgen bölgeyi arama işlemi yapılmıştır. Aradığı bölgeleri bulduktan sonra dört köşesini işaretleteme işlemi yapılmıştır. Bulunan bu bölgeleri çerçeve içerisine alınmışıtr.Görüntü üzerinde uygulanan kontur sonrasında plaka bölgesinin bulunması işlemi gerçekleştirilmiştir. Bulunan plaka bölgesi çerçeve içersine alınmıştır.

2.8.2.10 Plaka bölgesinin maskelenmesi

[10] Kontur uygulanan resim ile plaka çerçevesi buldurma işlemi gerçekleştirilmiştir. Görüntü üzerinden plakayı ayrıştırmak için maskeleme işlemi yapılır. Maskeleme yaparak bundan sonraki işlemlerde sadece plaka üzerinde işlemler gerçekleştirilir.

2.9 KARAKTERLERİN PLAKA BÖLGESİNDEN AYRIŞTIRILMASI

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 nesnelerden ayırmaya çalışır.Karakter tanıma işlemi ile otomatik plaka tanıma, verileri düzenlenebilir hale getirme, aranabilir

Şekil 2.11 : :Bulunan plaka yerinin fotoğraftan kırpılması işlemi

ve kolayca saklanabilir bilgilere dönüştürür. Görüntüye uygulanan belirlenmiş algoritmalardır.

[18] Plakayı bulmak için gerekli filtreler görüntü üzerinde uygulanması gerekir. Bu adımlar karakterlerin daha net bir biçimde gözükmesini doğruluk oranının artmasını sağlayacaktır. Plaka tanıma işleminde en kritik süreç karakter ayrıştırılması işlemidir. Görüntü üzerinden kesilen plaka bölgesi resim üzerinde düzgün bir biçimde olmayabilir. Karakterlerin tamamı aynı hizada değildir.Karakterlerin aynı hizada olması karakterin ayrıştırılması için önemli bir etkendir. Kaliteli bir biçimde ayrıştırma yapmak için eğim açısı tespit edilir.Aynı açıyla ters yönde döndürerek plaka x ekseni üzerinde paralel hale gelir. Plaka üzerindeki karakterler eğim işlemi sonucunda aynı hizaya gelecektir. Karakterlerin üst ve alt sınırları belirlendikten sonra karakter ayrıştırma işlemi gerçekleştirilir.

2.9.1 Plaka üzerinde eğimin düzeltilmesi

[15] Karakter ayrıştırma işleminin başarılı bir biçimde gerçekleştirmek için x eksenine göre eğim tespiti yapılması gerekmektedir. Eğim bulunan resmin x eksenine paralel bir biçimde gelmesi sağlanmalıdır.Eğim tespiti için karakterlerin x eksenine göre başlangıç ve bitiş noktaları tespit edilir. Y eksenine görede üst sınır ve alt sınır noktaları bulunur. Görüntü üzerinden alınmış karakterlerin yükseklikleri eşittir. Tespit edilen karakterlerin üst sınırları soldan sağa düzenli bir biçimde artma veya azalma gerçekleşiyorsa plaka üzerinde eğim vardır. Bu düzenli artma veya azama değeri tespit edilir, x eksenindeki ve y eksenindeki farkları hesaplanarak eğim açısı bulunur.

[19] Gerekli doğrulamalar yapılmaya başlanır .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, sınırlarının belirlenmesinde edilmesinde bazı

Şekil 2.12 : Plaka üzerinde tespit edilen eğimin örnek görüntüsü

durumlarda sorun çıkarmaktadır. Plaka yerini öğrenme yönteminde kullanılan, yan yana olan piksel değerlerinin farkı 20'nin üzerinde olması durumunda, bu pikseller kenar bölgesi kabul edilir. Bu sayede plaka çerçevesi oluşturulduğu için sorun ortadan kalkar.

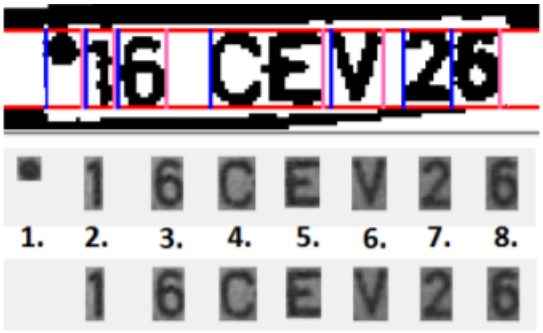
2.9.2 Plaka üzerindeki karakterleri parçalama işlemi

Plaka görüntüsünün eğimi düzeltildikten sonra tüm karakterlerin üst ve alt sınırları aynı y düzlemine gelmektedir. Fakat eğimi düzeltilmiş plaka görüntülerinde bile azda olsa x düzlemine göre eğim farklı bulunacağından karakterlerin y sınır eksenleri birebir uymaz, bu durum karakter parçalama için sorun teşkil etmemektedir. Bu aşamadan sonra plaka resminde karakterlerin üst ve alt sınırlarının tespit edilme işlemine gelinmiştir. 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 ekseni karakterlerin üst ve alt sınırını oluşturmaktadır. Hem hız bakımından hem de bulunan plaka bölgesinde, plaka çerçevesinin oluşturduğu bozulmadan kurtulmak için tüm x eksenini taramak yerine aşağıdaki denklemde gerekli değerler yerine konularak hesaplanır. Genişlik: Plaka genişliğinin (1) X1: x noktası taramanın başlayacağı nokta X2: x noktası taramanın bitirileceği nokta X1: (Genişlik / 8) * 3 X2: (Genişlik / 8) *5;

2.9.3 Karakterleri tanıma işlemi

[9] Plaka ayrıştırma sonucunda elde edilen sonuca göre karakterleri tanıma işleminin yapılacağı kısımdır. Plaka karakterlerini tanımlamak için yaygın yöntemler şablon yerleştirme ve yapay sinir ağları yöntemidir. Farklı yazı tipindeki karakterleri tanımlamak için genelde şablon eşleme yöntemi kullanılır. Şablon eşleme yöntemi ile karakter tanıma işlemi yaparken her bir karakterin yükseklik ve genişlllik boyutları

Şekil 2.13 : Plaka üzerinde karakter ayrıştırması yapılması

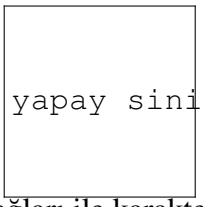


Şekil 2.14 : Karakter tanıma işlemi yapılan plaka görüntüsü

sabit biçime getirilmesi gerekir. Karakterler önceden belirlenerek tanıma işlemi gerçekleştirilir.

2.9.4 Şablon eşleştirme yöntemi

[4] Şablon eşleştirme yöntemi ile karakter tanıma işlemi her bir karakterin piksel değerlerinin önceden programa tanılması ile gerçekleştirilmektedir. Plaka üzerinde bulunan karakterler bu şablondaki karakter ile karşılaştırılması ile yapılır. [14] Karakter tanıma işlemine başlamadan önce şablon çıkarılmalıdır.Şablon eşleştirme yöntemi 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 uygun boyut 15x32 olarak belirlenmiştir. Farklı ebatta şablonlar bulunmaktadır. Şablon eşleştirme yöntemi görüntü üzerindeki tüm pikselleri dolaşarak önceden tanıtmış olduğumuz karakterlerin tıtatıp aynısını bulmaya çalışır.



Şekil 2.15 : Yapay sinir ağları ile karakter tanıma blok diyagramı

2.9.5 Yapay sinir ağları yöntemi

[20] 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.[15] 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. Karakteri belirleme işlemi yaparken karakterin en ve boy oranı vardır. Bundan önce yapılmış ve denenmiş çalışmalara göre bir karakterin eni minumum 5 piksel yüksekliği ise minumum 10 pikseldir. En ve boy altında bulunan her görüntü parazit olarak kabul edilmektedir. Ayrıştırılan plakar üzerinde karakterleri daha hızlı bir şekilde tanımak için plaka kodlaması üzerinde bulunan harf ve karakterleri daha önceden tanımlamak doğru sonuç alınması açısından fayda sağlayacaktır.

2.10 NEDEN SQLITE VERİTABANI KULLANIYORUM?

[**?**]

2.10.1 Veritabanı Nedir?

Bilgisayar terminolojisinde, sistematik erişim imkânı olan, yönetilebilir, güncellenebilir, taşınabilir, birbirleri arasında tanımlı ilişkiler bulunabilen bilgiler kümesidir. Bir başka tanımı da, bir bilgisayarda sistematik şekilde saklanmış, programlarca işlenebilecek veri yığınıdır.

2.10.2 Neden SQLite ?

```
#!/usr/bin/python

import sqlite3

conn = sqlite3.connect('test.db')

print "Opened database successfully";
```

Şekil 2.16 : Python'a Sqlite bağlantısı kurma kodu

Python'da veritabanı işlemleri için kullanabileceğiniz pek çok alternatif bulunur. Ama ben bütün bu alternatifler içinde Sqlite'ı tercih edeceği.m Peki neden Sqlite?

Sqlite'ın öteki sistemlere göre pek çok avantajı bulunur. Gelin isterseniz Sqlite'ın bazı avantajlarına şöyle bir göz gezdirelim :

- Sqlite herhangi bir yazılım veya sunucu kurulumu gerektirmez. Bu sayede, bu modülü kullanabilmek için öncelikle bir sunucu yapılandırmanıza da gerek yoktur. Bazı veritabanlarını kullanabilmek için arka planda bir veritabanı sunucusu çalıştırıyor olmanız gerekir. Sqlite'ta ise böyle bir şey yapmazsınız.
- Sqlite özgür bir yazılımdır. Bu yazılımın baştan aşağı bütün kodları kamuya açıktır. Dolayısıyla Sqlite kodlarının her zerresini istediğiniz gibi kullanabilir, değişikliğe uğratabilir, satabilir ve ticari olan/olmayan bütün uygulamalarınızda gönül rahatlığıyla kullanabilirsiniz.
- Her şeyden önce Sqlite Python'un 2.5 sürümlerinden bu yana bu dilin bir parçasıdır. Dolayısıyla eğer kullandığınız Python sürümü 2.5 veya üstü ise Sqlite'ı Python'daki herhangi bir modül gibi içe aktarabilir ve kullanmaya başlayabilirsiniz.

2.10.3 SQLite Python Kodlarına Nasıl Dahil Edilir?

Aşağıdaki Python kodu, mevcut bir veritabanına nasıl bağlanılacağını gösterir. Veritabanı yoksa, oluşturulacak ve son olarak bir veritabanı nesnesi döndürülecektir.
Daha önce oluşturulan veritabanında bir tablo oluşturmak için aşağıdaki Python programı kullanılacaktır. Yukarıdaki program çalıştırıldığında, test.db'nizde COMPANY tablosunu oluşturacaktır.

```
#!/usr/bin/python

import sqlite3

conn = sqlite3.connect('test.db')
print "Opened database successfully";

conn.execute('''CREATE TABLE COMPANY
         (ID INT PRIMARY KEY     NOT NULL,
         NAME           TEXT    NOT NULL,
         AGE            INT     NOT NULL,
         ADDRESS        CHAR(50),
         SALARY         REAL);''')
print "Table created successfully";

conn.close()
```

Şekil 2.17 : Python ile Sqlite içerisinde bir tablo oluşturma kodu

Aşağıdaki Python programı, yukarıdaki örnekte oluşturulan COMPANY tablosunda nasıl kayıt oluşturulacağını gösterir.

Yukarıdaki program çalıştırıldığında, COMPANY tablosunda verilen kayıtları oluşturacaktır.

Aşağıdaki Python programı, yukarıdaki örnekte oluşturulan COMPANY tablosundan kayıtların nasıl getirilip görüntüleneceğini gösterir.

Aşağıdaki Python kodu, herhangi bir kaydı güncellemek için UPDATE ifadesinin nasıl kullanılacağını ve ardından COMPANY tablosundan güncellenmiş kayıtları getirip görüntüleyeceğinizi gösterir.

Aşağıdaki Python kodu, herhangi bir kaydı silmek için DELETE ifadesinin nasıl kullanılacağını ve ardından COMPANY tablosundan kalan kayıtları alıp görüntüleyeceğinizi gösterir.

Sonuç olarak projemde Python ve Sqlite bağlantısı kurarak , aracın plakasını , otoparka giriş saatini ve aracı bıraktığı konumu veritabanı içerisinde tutuyorum. Daha sonrasında Python içerisinden Sqlite veritabanının barındırdığı verileri çekerek ücret hesaplanması ve aracın konumunun bulunması gibi işlevleri yerine getirebiliyorum.

```
#!/usr/bin/python

import sqlite3

conn = sqlite3.connect('test.db')
print "Opened database successfully";

conn.execute("INSERT INTO COMPANY (ID,NAME,AGE,ADDRESS,SALARY) \
      VALUES (1, 'Paul', 32, 'California', 20000.00 )");

conn.execute("INSERT INTO COMPANY (ID,NAME,AGE,ADDRESS,SALARY) \
      VALUES (2, 'Allen', 25, 'Texas', 15000.00 )");

conn.execute("INSERT INTO COMPANY (ID,NAME,AGE,ADDRESS,SALARY) \
      VALUES (3, 'Teddy', 23, 'Norway', 20000.00 )");

conn.execute("INSERT INTO COMPANY (ID,NAME,AGE,ADDRESS,SALARY) \
      VALUES (4, 'Mark', 25, 'Rich-Mond ', 65000.00 )");

conn.commit()
print "Records created successfully";
conn.close()
```

Şekil 2.18 : Python ile Sqlite içerisinde bir tabloya veri kaydı ekleme kodu

```
#!/usr/bin/python

import sqlite3

conn = sqlite3.connect('test.db')
print "Opened database successfully";

cursor = conn.execute("SELECT id, name, address, salary from COMPANY")
for row in cursor:
   print "ID = ", row[0]
   print "NAME = ", row[1]
   print "ADDRESS = ", row[2]
   print "SALARY = ", row[3], "\n"

print "Operation done successfully";
conn.close()
```

Şekil 2.19 : Python ile Sqlite içerisinde bir tablodan veri kayıtlarını çekme kodu

```
#!/usr/bin/python

import sqlite3

conn = sqlite3.connect('test.db')
print "Opened database successfully";

conn.execute("UPDATE COMPANY set SALARY = 25000.00 where ID = 1")
conn.commit()
print "Total number of rows updated :", conn.total_changes

cursor = conn.execute("SELECT id, name, address, salary from COMPANY")
for row in cursor:
   print "ID = ", row[0]
   print "NAME = ", row[1]
   print "ADDRESS = ", row[2]
   print "SALARY = ", row[3], "\n"

print "Operation done successfully";
conn.close()
```

Şekil 2.20 : Python ile Sqlite içerisinde bir tablodan veri kayıtlarını güncelleme kodu

```
#!/usr/bin/python

import sqlite3

conn = sqlite3.connect('test.db')
print "Opened database successfully";

conn.execute("DELETE from COMPANY where ID = 2;")
conn.commit()
print "Total number of rows deleted :", conn.total_changes

cursor = conn.execute("SELECT id, name, address, salary from COMPANY")
for row in cursor:
   print "ID = ", row[0]
   print "NAME = ", row[1]
   print "ADDRESS = ", row[2]
   print "SALARY = ", row[3], "\n"

print "Operation done successfully";
conn.close()
```

Şekil 2.21 : Python ile Sqlite içerisinde bir tablodan veri kayıtlarını temizleme kodu

main.py > ...

```
1  from datetime import datetime
2  import cv2
3  from matplotlib import pyplot as plt
4  import numpy as np
5  import imutils
6  import easyocr
7  import sqlite3
```

Şekil 2.22 : Proje içerisine kütüphanalerin import edilmesi işlemi

```
pic = input("Resim giriniz: ")
img = cv2.imread(pic)
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
cv2.imshow("Licence",gray) #griye çevrilmiş görüntü
bfilter = cv2.bilateralFilter(gray, 11, 17, 17) #Gürültü azaltma
edged = cv2.Canny(bfilter, 30, 200) #Kenar algılama
cv2.imshow("Licence2",edged) #Kenarları algılanmış görüntü
keypoints = cv2.findContours(edged.copy(), cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)
contours = imutils.grab_contours(keypoints)#plakadaki anahtar bölgelerin işaretlen
contours = sorted(contours, key=cv2.contourArea, reverse=True)
location = None
for contour in contours:
    approx = cv2.approxPolyDP(contour, 40, True)
    if len(approx) == 4:
        location = approx
        break
```

Şekil 2.23 : Kullacıdan alınan resmin işlenme aşamaları

2.11 PROJE KODLARI VE ÇIKTILARINA AÇIKLAMALI ŞEKİLDE YER VERİLMESİ

2.11.1 Kullanılacak Kütüphanelerin dahil edilmesi

Projenin bu kısmında kullanıcıdan input olarak alınan resim adı içerisinde farklı araç resimleri mevcuttur.Bu araç resimleri görüntü işleme kütüphaneleri yardımıyla tanımlanır ve nihai olarak tanımlanan aracın plakası text olarak geri dönüş yapar.

Kod satırlarında açıklamalara yer verilmiştir.

Şimdi de plakanın tanınmasının diğer aşamalarını inceleyelim. Aşağıdaki kod satırında arac resmi içerisinden plaka kırpılarak çıkarılıyor.

Kırpılmış plaka aşamasından sonra plakanın aracnın hangi konumunda olduğunu çevresine bir dikdörtgen çizerek gösteren kod satırına bakalım.

```
mask = np.zeros(gray.shape, np.uint8)
new_image = cv2.drawContours(mask, [location], 0,255, -1)
new_image = cv2.bitwise_and(img, img, mask=mask)
cv2.imshow("Licence3",new_image) #Sadece Plakayı gösterir.
(x,y) = np.where(mask==255)
(x1, y1) = (np.min(x), np.min(y))
(x2, y2) = (np.max(x), np.max(y))
cropped_image = gray[x1:x2+1, y1:y2+1]
cv2.imshow("Licence4",cropped_image) #Plakayı Keserek Gösteriyor.(kırpılmış hali)
reader = easyocr.Reader(['en']) #Plakada kullanılan dil
result = reader.readtext(cropped_image)
print(result) #kesilmiş plakanın döndürülmesi

text = result[1][-2]
```

Şekil 2.24 : Plakanın araç resmi içeriisnden kırpılması

```
text = result[1][-2]

font = cv2.FONT_HERSHEY_SIMPLEX
res = cv2.putText(img, text=text, org=(approx[0][0][0], approx[1][0][1]+60), fontFace=fon
lineType=cv2.LINE_AA)
res = cv2.rectangle(img, tuple(approx[0][0]), tuple(approx[3][0]), (0,255,0),3) #plaka bö
# dikdörtgenin şekillendirilmesi
```

Şekil 2.25 : Araç resminde bulunan plaka bölgesinin dikdörtgen içerisinde gösterilmesi

```
con = sqlite3.connect('plaka.db') #python sql bağlantısının kurulması
cur = con.cursor() #bağlantısı sağlanan veritabanın cursor yardımıyla tek tek gezi
konum = ""
length = 0
cur.execute("SELECT * FROM Arac WHERE Plaka=?",(text,)) #tablodan plakaların çekil

rows2 = cur.fetchall()#tüm plakaları çek
```

Şekil 2.26 : Sqlite veritabanına bağlantı sağlama kodları

```
if(len(rows2)==0): #eğer tablodan çekilen plakalar null ise yani araç henüz park ed
    while(len(konum)!=2 or length != 0 ): #konum null iken
        konum = input("Araçın Konumunu Giriniz:(Örn: A2) ")
        cur.execute("SELECT * FROM Arac WHERE Konum=?",(konum,)) #tablodan konumu ç
        rows = cur.fetchall() #tüm konumları çek
        length = len(rows)
        if length != 0: #tablodan çekilen konum null değilse yani o konumda bir ara
            print("Girmiş olduğunuz alan dolu. Başka alan giriniz.")
        if(len(konum)!=2):
            print("Lütfen doğru konum giriniz.")
cur.execute("SELECT * FROM Arac WHERE Plaka=?",(text,)) #tablodan plakaları seç
```

Şekil 2.27 : Plakası tanımlanan aracın konumunun belirlenmesi

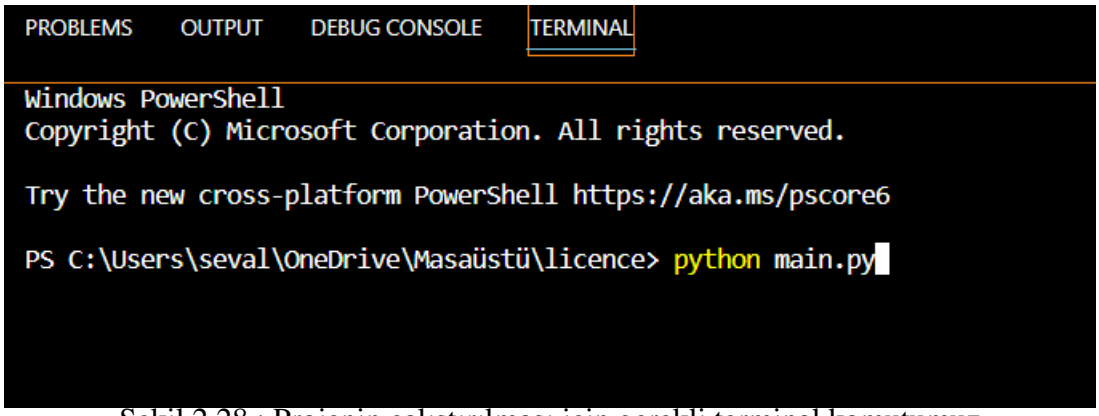
[22] Projenin bu kısmında SQLİTE bağlantısı gerçekleştiriyoruz ki tanıdığımız plakaların otoparka giriş saatlerini ve aracın konumunu bir veritabanı içerisinde tutabilelim. Böylelikle araç otoparktan çıkarken ne kadar otopark ücreti ödemesi gerektiğini bulabilir ve aracın otopark içerisindeki konumuna erişebiliriz.

[23] Bu aşamada ise plakası tanımlanan aracın konumunu kullanıcan girmesini istiyoruz. Eğer otoparkta bu konumda bulunan bir araç yok ise aracın otoparka giriş yapmasını sağlıyoruz. Araç otopark içerisine girdiğinde veritabanında da yer alır. Aracın veritabanında plakası konumu ve giriş zamanı tutulur.

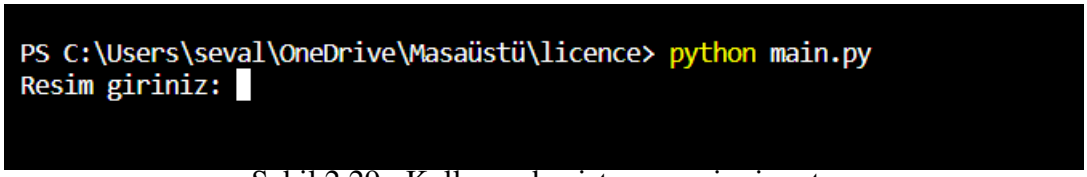
Son kısımda yer alan kodlarımda , aracın plakası daha önce veritabanına kayıtlı ise direkt olarak ücret hesabı ve aracın konumu bildirilip veritabanından aracın kaydının silinmesi işlevi gerçekleştirilirken , plakanın henüz veritabanında kaydı bulunmuyor ise , kaydı oluşturulmaktadır.

2.12 PROJEDEN ELDE EDİLEN ÇIKTILARIMIZ

2.12.1 Projenin Çalıştırılması



Şekil 2.28 : Projenin çalıştırılması için gerekli terminal komutumuz



Şekil 2.29 : Kullanıcıdan istenen resim inputu

2.12.2 Kullanıcıdan Resim Inputunun Alınması

2.12.3 Kullanıcıdan Alınan Resim Inputuna Göre Plaka Tanımlamasının Yapılması

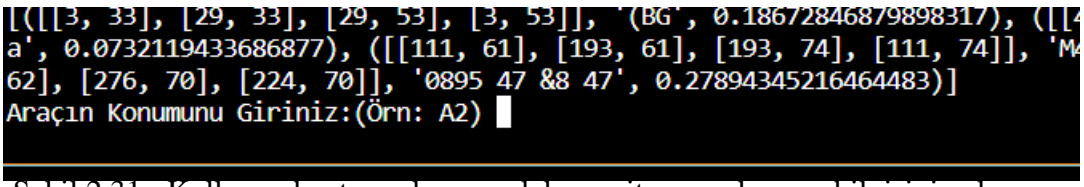
Bu kısımda kullanıcın girdiği resim inputu içerisinde bulunan araç resmindeki plaka tanımlanır .

2.12.4 Kullanıcıdan Konum Bilgisinin Alınması

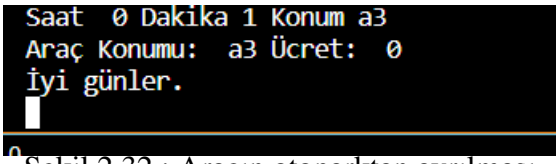
Bu kısımda hali hazırda tanımlanan plakaya bağlı aracın otoparkın hangi konumunda park edileceği bilgisi kullanıcıdan alınır.
Konum bilgisi yalnızca iki haneli olmak zorundadır.
Kullanıcının aracı bırakmak istediği konumda bir araç zaten bulunuyor ise kullanıcı başka bir konum girmek zorundadır.

2.12.5 Aracın Otoparktan Ayrılması

Şekil 2.30 : Kullanıcıdan istenen resim bilgisine göre plakanın işlenmesi



Şekil 2.31 : Kullanıcıdan tanımlanmış plakaya ait aracın konum bilgisinin alınması



Şekil 2.32 : Aracın otoparktan ayrılması

Eğer tanımlanmış plaka öncesinde veritabanında bulunuyor ise biz programı çalıştırdığımızda kullanıcı yine aynı aracın resim bilgisini girdiği zaman plaka ikinci kez veritabanına kaydedilmez.

Bu demektir ki araç artık otoparktan ayrılmak istiyor. Böylelikle veritabanından bu plakaya ait bilgilerin kayıtları temizlenir ve otopark ücreti ile aracın konum bilgisi kullanıcıya döndürülür.

3. HİPOTEZ

3.1 Tezin Hipotezi

Yapılan bu uygulamada bir otoparktaki araçların görüntü işleme ile plakaları tanınarak sqlite veritabanında saklanır. Saklanılan veriler ile plakalara bağlı otopark ücret hesabı sonuçları elde edilir.

[3] İlk olarak plaka tanıma işleminin nasıl gerçekleştiğine değineceğiz.

Plaka tanıma sisteminde kullanılan algoritma aşaması 3 kademe şeklindedir.

Bu aşamalardan ilki alınan görüntü üzerinde plaka yerinin bulunması, ikincisi bulunan plakanın üzerindeki karakterlerin parçalanması ve son olarak elde edilen karakterlerin tanıma işlemidir. Plaka sistemleri üzerine yapılan çalışmaların genelinde hazır resim filtreleme yöntemleri kullanılarak plaka yeri tespit edilmiştir. [7] 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 fitreler opencv kütüphanesi içinde bulunan hazır filtrelerdir. Bu proje boyunca kullanılan filtreler plaka tanıma işlemi için hazırlanmış filtreler değildir. Genel olarak görüntü işlemede kullanılan filtrelerdir.

[12] Gerçek zamanlı bir plaka tanıma sisteminde kameradan alınan görüntüde ilk amaç hızlı bir şekilde plaka yerini tespit etmektir. Bunun için tüm görünütüyü 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 sayade plaka yerinin tespiti işleminin hatasız bir biçimde yapılması için gerekli önlemler alınmıştır. [9]

Plaka bölgesinin bulunması sırasında aracın üzerinde bulunan plakaya benzeyen marka yazısı, logo veya buna benzer karakterlerin olması ile bu yerlerin plaka olarak belirlenmesi hatası ile bazen karşılaşılmıştır.Gece alınan görüntü üzerinde çalışma işlemi gerçekleştirilmemiştir. Plaka tanıma işleminde kullanılan görüntülerin çözünürlük oranı arttıkça plaka bölgesini bulma işlemindeki başarı oranı doğrusal olarak artış göstermiştir. Bu çalışmada farklı renklerdeki araçlar üzerinde denenmiştir. Araçların rengi farketmeksizin plaka bölgesini belirleme işlemi başarıyla gerekleşmiştir. Araç üzerindeki görüntüler hem ön tarfata hem de arka taraftan alınan görüntülerde denenmiştir.

Her iki açıdan da alınan görüntülerde de başarı oranı neredeyse aynıdır. [15] 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.

[8] Şablon eşleştirme yöntemi plaka tanıma sistemi gibi yüksek derecede doğruluk bilgisi gereken yerlerde kullanılması sorun yaşatabilir. Şablon eşleştirme yerine yapay sinir aları yöntemi kullanarak daha yüksek oranda başarı oranına ulaşılabilir. Sonuç olarak alışveriş merkezlerinde, sınır kapılarında, üniversite kampüsüne giriş çıkışlarda, otoparklarda, emniyet müdürlüğü gibi alanlarda aracın giriş, çıkış, çalıntı veya istatiksel gibi verilerin tutulması gereken yerlerde kullanılmaktadır.Plaka tanıma sistemi günümüzde neredeyse zorunlu hale gelmiştir. Plaka tanıma sistemi sayesinde güvenlik zafiyetleri engellenmesi için yardım etmekte ve insan gücünü en aza indirmektedir. İnsanların 24 saat boyunca kağıtta kimin girip çıktığını not tutması ihtiyacı ortadan kalkmıştır. Tüm önlemler alınarak yapılan bir plaka tanıma sistemi sayesinde 24 saat boyunca sürekli aynı verimde çalışabilir. Plaka tanıma sisteminin gelecekte aracın bulunduğu her noktada hayatımızda olacağı düşünülmektedir.

Plaka tanıma işlemi sonucunda araçların plakaların ,kullanıcıdan girilen konum bilgileri ve otoparka giriş saatleri SQLite veritabanında kayıt olarak tutulur. Bu kayıtlar

otoparktan araç çıkarken , python ile çekilir (select edilir) ve nihai olarak istediğimiz otopark ücreti ve plakaya sahip aracın konumunun bize döndürülmesine olasılık sağlar.

Kaynakça

[1] https://www.codecuit.com/goruntu-isleme-ve-python/.

[2] **Bulsari ve Saxen** (1991). A Feedforward Artificial Neural Network for System Identification of a Chemical Process,.

[3] https://github.com/NisanurBulut/LicensePlateRecognitionSystem/.

[4] https://datacarpentry.org/image-processing/.

[5] http://ibrahimdelibasoglu.blogspot.com/2016/11/hsv-renk-uzay.html/.

[6] http://ibrahimdelibasoglu.blogspot.com/2016/11/hsv-renk-uzay.html/.

[7] https://yavuzbugra.wordpress.com/2011/05/01/goruntu-islemede-filtreleme/.

[8] **Acosta, B.** (2004). Experiments in image segmentation for automatic US license plate recognition.

[9] https://github.com/BhanuPrakashNani/Image_Processing/.

[10] **Barroso J., Rafael A., .D.E.L. ve J., B.** (1997). Number plate reading using computer vision, IEEE - International Symposium on Industrial Electronics.

[11] https://github.com/anlgungordu/PlakaTanimlama/.

[12] **Christos-Nikolaos E. Anagnostopoulos, Ioannis E. Anagnostopoulo, I.D.P.** (2008). License Plate Recognition From Still Images and Video Sequences: A Survey.

[13] **R., G. ve R., W.** (1992). Digital Image Processing.

[14] **da Fontoura Costa, L. ve Jr., R.M.C.** (1992). Shape Analysis and Classification.

[15] **R., G. ve R., W.** (2002). Morphology-based License Plate Detection from Complex Scenes,.

[16] **S., C.J.** (1997). Applications Of Computer Vision to Road Traffic Monitoring.

[17] https://medium.com/analytics-vidhya/skeletonization-in-python-using-opencv-b7fa16867331/.

[18] **Zhuang Liu, Y.Z.** (2020). Vehicle License Plate Recognition In Complex Scenes.

[19] **B. Y. Amirgaliyev, C. A. Kenshimov, K.K.K.M.Z.K.Z.Y.B.** (2020). License plate verification method for automatic license plate recognition systems.

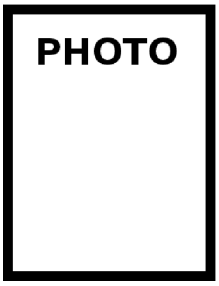
[20] **Sagiroglu S, L.M.** (1996). Metodunun Bir Robot Sensörün Yapay Sinir Ağı Modellenmesinde Kullanılması.

[21] `https://www.tutorialspoint.com/sqlite/sqlite_python.htm`.

[22] `https://docs.python.org/3/library/sqlite3.html`.

[23] `https://zetcode.com/python/sqlite/`.

ÖZGEÇMİŞ



Ad Soyad: Seval Naz KARAHAN

Doğum Tarihi ve Yeri: 13.02.1999-Muğla

E-Posta: skarahan171@posta.pau.tr

ÖĞRENİM DURUMU:

- **Lise:** 2016, Muğla Köyceğiz Fen Lisesi-Ortaca Açı Temel Lisesi
- **Lisans:** 2022, Pamukkale Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği

MESLEKİ DENEYİMLER VE ÖDÜLLER:

- Staj I:Teknokent Kodpit A.Ş, 2019
- Staj II:B firması, yıl
- Programmer: Hiera Soft A.Ş, 2017-2018
- Front-End Developer: Shaper System , 2021-2022
- Game Developer: Data Games, 2022- 2022
- Game Developer: Vulpis Yazılım, 2022- ...(STILL WORKING)