



# AKILLI OTOPARK SİSTEMİ HAFTALIK DOKÜMAN

FARUK ÇETİN

April 25, 2024

## Abstract

This report presents in detail a smart car parking system design that aims to bring an innovative approach to traditional car parking systems. Aiming to provide a solution to the increasing vehicular traffic and car parking problems in modern cities, this project is realised by integrating special cameras including number plate reading technology and visual recognition features. The design aims to enable drivers to automatically recognise number plates, quickly identify empty car parking spaces and provide more efficient parking management. This innovative system aims to improve traffic flow and optimise urban transport while enhancing users' parking experience. By focusing on the details of the design process, the report reveals the key elements of this project, which has the potential to provide an advanced solution for smart car parking systems.

# İçindekiler Tablosu

<b>1</b>	<b>Özet</b>	<b>3</b>
<b>2</b>	<b>Önemli</b>	<b>3</b>
<b>3</b>	<b>Yöntem</b>	<b>3</b>
3.1	İlk Proje için Yapılanlar . . . . .	3
3.2	İkinci Proje İçin Yapılanlar . . . . .	5
3.3	Bulgular ve Tartışma . . . . .	6
3.3.1	İşaretlenen Dataların Kayması . . . . .	6
3.3.2	Yanlış Gösterimler . . . . .	7
3.4	Veriseti Bilgilendirme . . . . .	7
3.5	Şuan Öğrenilmeye Çalışılan Kütüphaneler . . . . .	8
3.5.1	Opencv Kütüphanesi(CV2) . . . . .	8
3.5.2	Matplotlib Kütüphanesi(PLT) . . . . .	9
3.6	Hangi Yapay Zeka Yöntemi . . . . .	9
3.7	Sonuç . . . . .	9

# 1 Özet

Bu rapor, akıllı otopark sistemleri üzerine detaylı bir tasarımı sunmaktadır. Geleneksel otopark sistemlerine yenilikçi bir yaklaşım getirmeyi hedefleyen bu tasarım, modern şehirlerdeki artan araç trafiği ve otopark sorunlarına çözüm sağlamayı amaçlamaktadır. Projede, özel kameraların entegrasyonu ile plaka okuma teknolojisi ve görsel tanıma özelliklerini birleştiren bir yaklaşım benimsenmiştir. Sürücülere plakaları otomatik olarak tanıma, boş otopark alanlarını hızlı bir şekilde belirleme ve daha verimli bir otopark yönetimi sağlama amacı güdülmüştür. Bu yenilikçi sistem, trafik akışını iyileştirmeyi, kentsel ulaşımı optimize etmeyi ve kullanıcıların otopark deneyimini geliştirmeyi amaçlamaktadır. Tasarım sürecinin detaylarına odaklanarak, bu projenin akıllı otopark sistemleri için gelişmiş bir çözüm sağlama potansiyelini ortaya koymaktadır.

# 2 Önemli

Bu bölümde, ilk proje kodları yazılırken youtube kanalından yardım alınmıştır[1]. İkinci projede de kodlar yazılırken youtube kanalından yardım alınmıştır[2]. Son olarak, hata kodları çözümü, bilgi öğrenme ve kod optimizasyonu durumlarında chatgpt'den yardım alınmıştır[3].

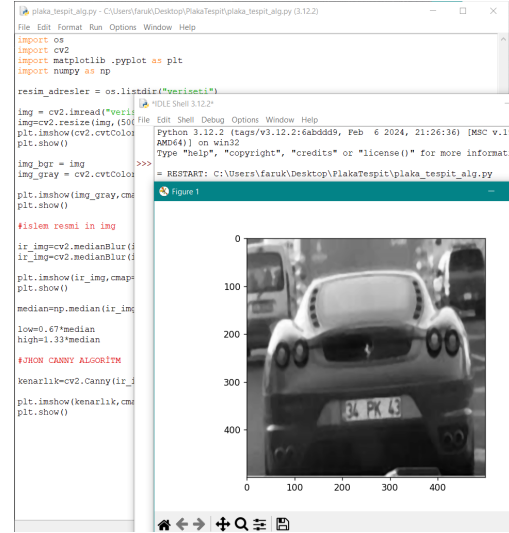
# 3 Yöntem

## 3.1 İlk Proje için Yapılanlar

Projenin ilk aşamasında, akıllı otopark sisteminin giriş bölümü için plaka tespit yapısının oluşturulması gerçekleştirildi. Python programlama dili kullanılarak veriler veri setinden çekilerek ekrana yansıtıldı. Ardından, otopark girişinde plakanın o an kamerada bulunduğu konumu algılayan algoritma yazılması için gerekli bilgiler toplandı. Bu bilgiler arasında dikdörtgen tespiti, (h,w) oranı ve boyut sınırlandırma gibi detaylar belirlendi. Daha sonra resim gri formata çevrilerek, gürültü ve medyan bulanıklaştırma işlemleri uygulandı.



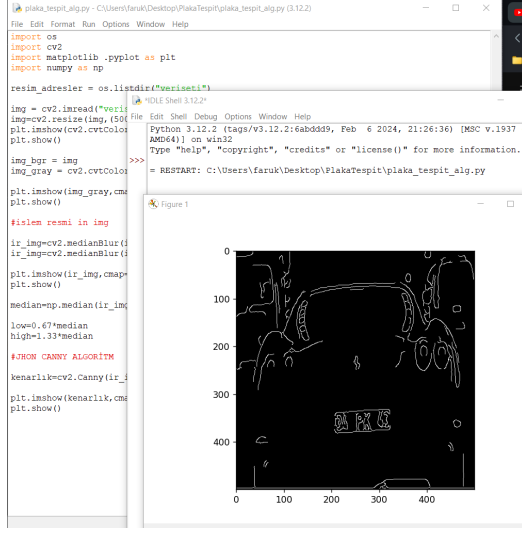
Şekil 1: a)



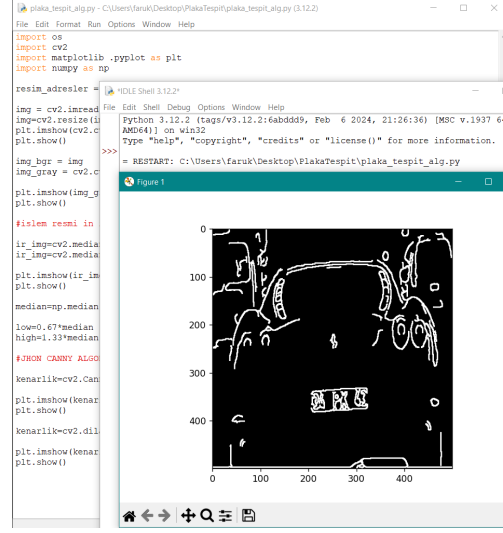
Şekil 2: b)

a) Gri Formata Çevrilmiş Resim b) Medyan Bulanıklaştırma

Jhon Canny Algoritması kullanılarak kenarların algılanması sağlandı ve kenarlıkların algılanmasının ardından ayrıştırma ve eşikleme işlemleri yapılarak plaka karakterlerinin daha iyi seçilmesi sağlanmaya çalışıldı.



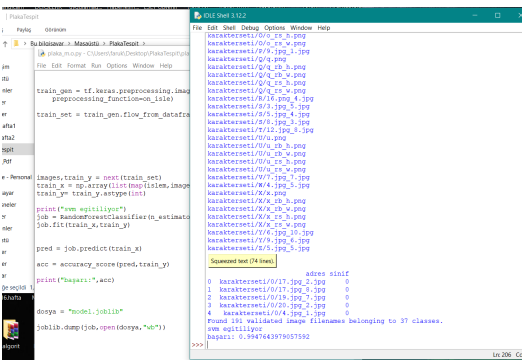
Şekil 3: a)



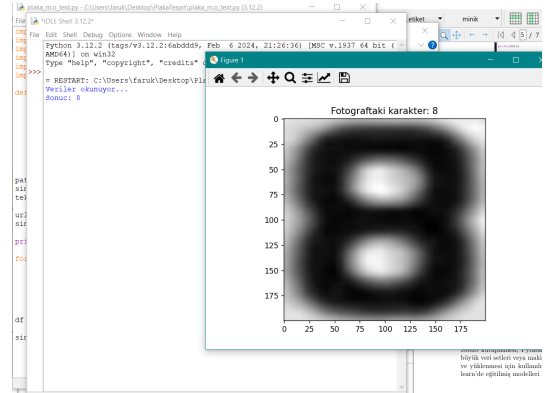
Şekil 4: b)

a) Canny Algoritması Sonuc Çıktısı b) Genişletilme Yapıldıktan Sonra Görüntü

Projenin sonraki aşamasında ise, akıllı otopark sisteminin son bölümü için veri setinden alınmış plaka görüntülerinden, plakaları tanımak için bir makine öğrenimi modeli kullanıldı. Bu aşamada, yukarıda belirtilen önemli kütüphaneler import edilerek bir model dosyası (model.joblib) ve sınıf etiketleri belirlendi. Plaka görüntülerinin işlenmesi için gerekli işlevler ve işlem adımları tanımlanarak plakaların ayıklanması, plaka karakterlerinin tanınması ve tanınan karakterlerin plaka olarak düzenlenmesi gibi işlemler gerçekleştirildi. Kod içerisinde plaka tanıma işlemlerini gerçekleştiren bir fonksiyon olan "plakaTani" işlevi ile yapılmaya çalışıldı. Bu işlev, girdi olarak bir görüntü ve plakanın koordinatlarını alır ve plaka içindeki karakterleri tanımakla görevli fonksiyon işlevini görmektedir. Son olarak ise modelimizin denemeleri ve test aşamaları yapılarak hatalar ve ileride ne gibi düzeltmeler yapılacağı belirlendi.



Şekil 5: a)



Şekil 6: b)

```
adres sinif
0 karakterseti/0/17.jpg_2.jpg 0
1 karakterseti/0/17.jpg_8.jpg 0
2 karakterseti/0/19.jpg_7.jpg 0
3 karakterseti/0/20.jpg_2.jpg 0
4 karakterseti/0/4.jpg_1.jpg 0
Found 191 validated image filenames belonging to 37 classes.
svm egitiliyor
başarı: 0.9947643979057592
>>
```

Şekil 7: c)

a) Makine Öğreniminin Sağlanması b) Test Verilerinin Denenmesi c) Başarı Oranının Çıktısı

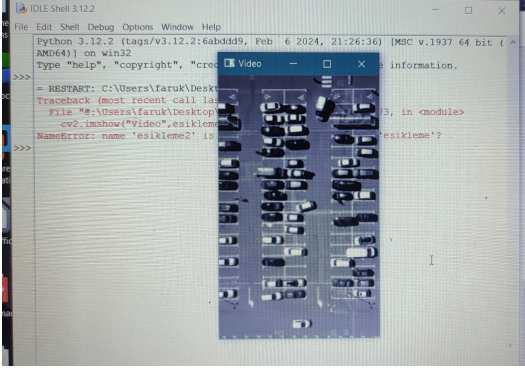
### 3.2 İkinci Proje İçin Yapılanlar

Projenin ikinci aşamasında, yoğun bir bilgisayarla görme ve veri işleme çalışması gerçekleştirildi. Ana odak noktası, bir video dosyasındaki nesnelerin tespit edilmesi ve takip edilmesi üzerineydi. İlk önce resim incelenmek üzere ekran yansıtmaya gerçekleştirildi. Koda işaretlemeler yapılabilmesi için gerekli kod satırları yazıldı ve hatalı işaretleme durumu için silme işlemi kodu entegre edildi. İşaretleme yapılarak hem pickle hem de csv dosyasına kaydedildi.

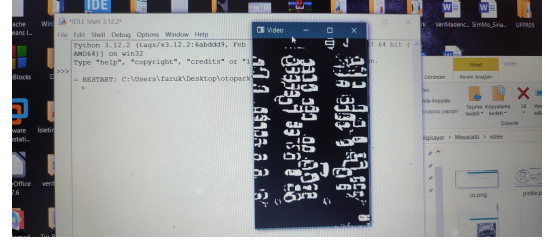


Şekil 8: Otopark Alanlarının İşaretlenmesi

Daha sonra video dosyası üzerinde nesne tespiti ve analizi yapmak ve doluluk boşluk durumunu öğrenmek için yeni dosya açılarak opencv kütüphanesi kullanılarak görüntü işleme teknikleri uygulandı; gri tonlamalı görüntüler elde etmek için renk dönüşümleri, median bulanıklığı ve adaptif eşikleme gibi yöntemler kullanıldı.



Şekil 9: a)



Şekil 10: b)

a) Resmin Gri Formata Dönüştürülmüş Hali b) Resmin Eşikleme Yapıldıktan Sonraki Hali

Ardından, nesneler tespit edildi ve bu nesnelerin etrafına dikdörtgenler çizilerek görselleştirildi. Ayrıca, kullanıcı etkileşimli bir arayüz üzerinden nesne tespiti için referans noktaların belirlenmesini sağlayan bir sistem geliştirildi. Son olarak ise model test aşamasına sokularak gerekli bilgiler toplandı ve hataların neler olduğu bulundu. Bu bilgilerden yararlanılarak projenin nasıl ilerlenmesi gerektiği hakkında çıkarımlarda bulunuldu.

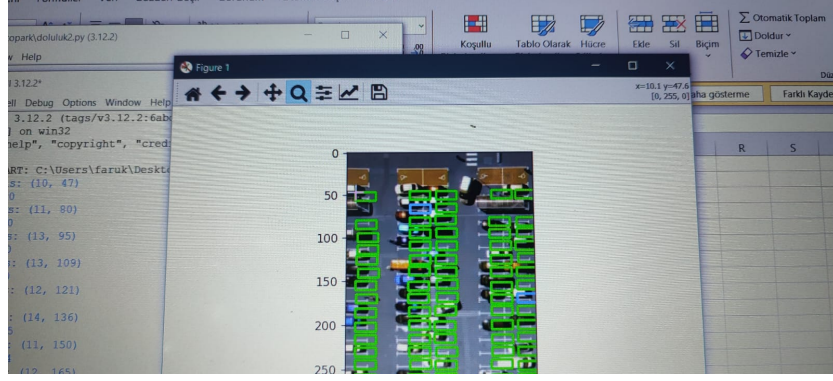


Şekil 11: Kurgulanan Modelin Çıktısı

### 3.3 Bulgular ve Tartışma

#### 3.3.1 İşaretlenen Dataların Kayması

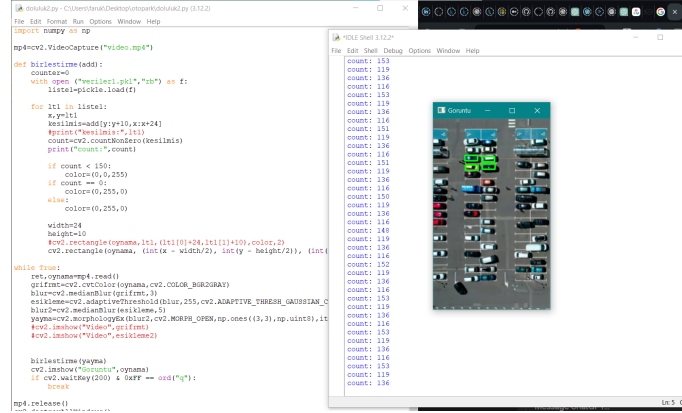
Bu hatada önceki kodda işaretlediğim dikdörtgenler belli konumdaydı. Mouse ile basıldığında, noktayı ortalayarak dikdörtgen çiziliyordu. Fakat ikinci kodda, mouse ile basılan nokta köşe olarak kabul edilip dikdörtgen böyle oluşturuluyordu. Bu da dikdörtgenlerin kaymasına ve otopark yerlerinin yanlış işaretlenmesine neden oluyordu. Bu hata birinci koddan aynı dikdörtgen hesabı kodu alınarak düzeltildi.



Şekil 12: Dataların hatalı bir şekilde kayması

### 3.3.2 Yanlış Gösterimler

Modelin büyük hatalarından birisi, yanlış gösterimdir. Makine öğrenimine sokulmadığından dolayı, oluşturulan model doluluk şeklini yoğunluğa göre belirlendi. Yani renk pikselleri 100'den fazla ise orada araba var ve otopark yeri dolu demektir fakat bu, gölge çıktığında ya da aşırı güneş vurduğunda yanlış sonuçlara neden olmaktadır. Ayriyeten de insanlar geçtikçe de ani tepki vererek çok kısa süreliğine kırmızı yanmasına ve yanlış bilgiler oluşmasına neden olmuştur. Hala düzeltilme aşamasındadır.

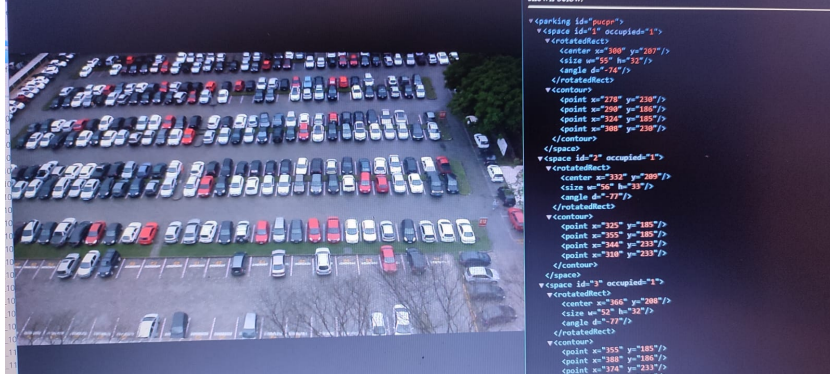


Şekil 13: Dataların hatalı bir şekilde kayması

## 3.4 Veriseti Bilgilendirme

Yapılan projelerden sonra konu hakkında bilgiler elde edilmiştir. Bundan sonra makine öğrenimi sürecine geçmek için hazırlıklar yapılmaktadır. İlk önce makine öğrenimi için dataset araştırıp bulunmuştur. 11gb olan datasette 3 farklı mekan için karlı, yağışlı ve güneşli gibi hava durumları olaylarında çekilmiş fotoğraflar vardır. Belli tarih ve günlerde çekilen fotoğraflarda, otoparkın dolu ve boş olduğu zamanlarda çekilen fotoğraflar sayesinde taglemeler yapılarak veriseti oluşturulmuştur. Boş otopark yerleri için 0 değeri verilmiş ve dolu otopark yeri için ise 1 değeri verilmiştir. Veri boyutu fazla olduğundan dolayı net bir şekilde ne kadar veri taglendiği bilinmemektedir.





Şekil 14: Kullanılacak Dataset

### 3.5 Şuan Öğrenilmeye Çalışılan Kütüphaneler

Bu bölümde Projenin ilerleyişinde kullanılan ve öğrenerek kullanmam gereken kütüphaneler anlatılacaktır.

#### 3.5.1 Opencv Kütüphanesi(CV2)

OpenCV (Open Source Computer Vision Library), bilgisayar görüşü ve görüntü işleme alanlarında yaygın olarak kullanılan açık kaynaklı bir kütüphanedir. Bu kütüphane, gerçek zamanlı görüntü işleme, nesne tanıma, nesne izleme, yüz tanıma, video analizi gibi birçok görevi gerçekleştirmek için bir dizi işlev ve algoritma içerir.



Şekil 15: Opencv Kütüphanesi Kullanım Örneği

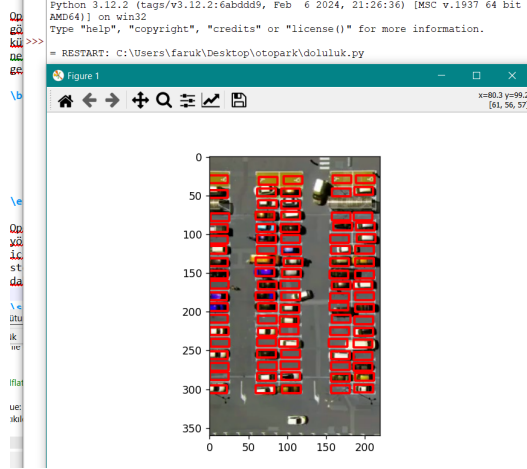
OpenCV'nin temel amacı, bilgisayarlı görüş ve görüntü işlemenin birçok yönünü desteklemek, araştırma yapmak ve gelişmiş uygulamalar oluşturmak için bir platform sağlamaktır. Yüz tanıma, nesne tanıma, nesne izleme, stereo görüş, hareket tespiti, görüntü segmentasyonu, derin öğrenme ve daha birçok görevi gerçekleştirebilen geniş bir işlev yelpazesi sunar.



### 3.5.2 Matplotlib Kütüphanesi(PLT)

Matplotlib, grafiklerin, histogramların, dağılım grafiklerinin, 3D grafiklerin ve daha birçok görsel öğenin oluşturulmasını sağlar. Ayrıca, interaktif grafikler oluşturmanızı da sağlayan çeşitli arayüzler ve grafik araçları sunar.

Genellikle `import matplotlib.pyplot as plt` şeklinde kısaltılarak kullanılır. Bu, Matplotlib'in en popüler alt modülü olan pyplot modülünü içe aktarır ve kısayol olarak plt adını kullanır. Bu şekilde, Matplotlib'in çeşitli grafik oluşturma fonksiyonlarına daha kolay erişebilirsiniz.



Şekil 16: Matplotlib Kütüphanesi

### 3.6 Hangi Yapay Zeka Yöntemi

Bu projede, ilk olarak makine öğrenimi ile basit bir şekilde sistemin boş ve dolu yerlerinin algılanması sağlanacaktır. Bunu da daha önce yaptığımız plaka tanıma sistemine benzer şekilde öğrenilenler uygulanarak yapılacaktır. Ayriyeten son yapılan opencv ile otopark yer boş dolu sistemi geliştirilerek makine öğrenimine hazır hale getirilecek tüm otoparklarda çalışabilecek bir sistem tasarlanması için geliştirilecektir. Bu işlem bittikten sonra ise makine öğrenimi istenilen etkiyi ve doğruluğu vermez ise bu projeyi geliştirmek açısından derin öğrenme yöntemine başvurulacaktır. Bunu yapmamızın nedeni ise makine öğreniminin modelimizde yeterince düzgün çalışmamasından olacaktır. Bu işlem de bittikten sonra genel güncellemeler yapılarak akıllı otopark sistemimiz tam olarak hazır hale gelecektir.

### 3.7 Sonuç

Bu rapor, akıllı otopark sisteminin tasarımı ve geliştirilmesi üzerine yapılan bir çalışmanın sonuçlarını sunmaktadır. Geleneksel otopark sistemlerinin mevcut zorluklarını aşmak ve modern şehirlerdeki artan araç trafiği sorunlarına çözüm getirmek amacıyla, plaka tanıma teknolojisi ve görsel tanıma algoritmalarının kullanımıyla bir sistem tasarlanmıştır. Çalışmanın sonuçları, mevcut sistemlerin iyileştirilmesine dair önemli ipuçları sunmaktadır. Ancak, derin öğrenme ve makine öğrenimi gibi daha ileri seviye tekniklerin uygulanmasına henüz geçilmemiştir.

## Kaynakça

- [1] umutkaanbaser, “plakatanimasistemi.” <https://github.com/umutkaanbaser/plakatanimasistemi.git>, 2022. 18.03.2024.
  - [2] S. K. Şahin, “Otopark boş alan tespiti bölüm - 1 ve otopark boş alan tespiti bölüm - 2.” <https://www.youtube.com/@SerhatKaganSahin>, 2023. 14.04.2024.
  - [3] OpenAI, “Chatgpt-3.5,” 2021. Bu kod ChatGPT 3.5 tarafından yazılmıştır.
- ...