KENAR BULMA YÖNTEMLERİ İLE PARK ALANI TESPİTİ

PARKING AREA DETECTION WITH EDGE FINDING METHODS

BUSE YENER, NİSA AKSOY

BİLİŞİM SİSTEMLERİ MÜHENDİSLİĞİ

KOCAELİ ÜNİVERSİTESİ

busee.yener@gmail.com

BİLİŞİM SİSTEMLERİ MÜHENDİSLİĞİ

KOCAELİ ÜNİVERSİTESİ

nisaaksoy08@gmail.com

Özet

Bu dokümanda, görüntü işleme teknikleri kullanılarak kenar bulma yöntemleri ile park alanı tespiti yapılmasının üzerinde durulmuştur.

Abstract

In this document, we focus on parking lot detection with edge detection methods using image processing techniques.

1.Giriş

Projemizi yaparken Görüntü İşleme kapsamındaki tekniklerden faydalanarak "Kenar Bulma Yöntemleri ile Park Yeri Tespiti" başlığını gerçekleştirdik. Öncelikle Kenar Bulma Yöntemleri ile Park Yeri Tespiti projesinden öneminden bahsedelim.

1.1 Akıllı Şehirler ve Teknolojik Gelişim

Akıllı Şehir Çözümlerinin Bir Parçası: Şehirleşmenin hızla arttığı bir dünyada, otopark yönetimi ve trafik düzeni, akıllı şehir çözümlerinin kritik bileşenleridir. Kenar bulma yöntemleri, kamera tabanlı otopark sistemlerinin temelini oluşturur.

Trafik Problemlerini Çözme: Araç sahiplerinin boş park yeri ararken trafikte daha fazla zaman harcaması, trafik sıkışıklığını artırır. Park alanlarını hızlı bir şekilde tespit etmek, bu problemleri azaltabilir.

1.2 Ekonomik Perspektif

Düşük Maliyetli Çözümler: Kenar bulma tabanlı sistemler, pahalı sensör teknolojilerinden bağımsız olarak kameralarla çalışabilir. Bu, özellikle geniş çaplı otoparklarda maliyetlerin düşürülmesini sağlar.

Manuel Kontrolün Azaltılması: Geleneksel otopark yönetimi, insan müdahalesi gerektirir. Otomatik sistemler, çalışan ihtiyacını azaltarak işletme maliyetlerini düşürür.

1.3 Çevresel ve Toplumsal Perspektif

Karbon Salınımını Azaltma: Araçların boş yer arayarak fazladan yakıt harcaması engellenir. Bu, çevre kirliliğini azaltır ve sürdürülebilir şehirleşmeye katkı sağlar.

Zaman Tasarrufu ve Kullanıcı Memnuniyeti: Araç sahipleri, park yerlerini hızlıca bulur. Bu hem kullanıcı memnuniyetini artırır hem de şehir içi zaman kaybını azaltır.

1.4 Otopark Yönetimi ve İşletme Perspektifi

Geniş Ölçekli Otoparkların Yönetimi: Alışveriş merkezleri, havaalanları ve büyük otoparklarda manuel kontrol neredeyse imkansızdır. Kenar bulma tabanlı sistemler, bu alanların otomatik yönetimini sağlar.

Dinamik Fiyatlandırma: Boş/dolu oranına göre fiyatlandırma yapmak, otopark işletmelerinin gelirlerini optimize edebilir.

Kenar Bulma Yöntemleri ile Park Alanı Tespiti projemiz için birçok kaynaktan literatür taraması gerçekleştirdik. İlk başta Kenar Bulma Yöntemleri kullanarak bulmuş olduğumuz otopark görüntülerinde çeşitli çıktılar aldık.

Görüntü İşleme kapsamındaki Kenar Bulma Yöntemlerinden bahsedelim.

1. Canny Kenar Algoritması: Çift eşikli bir yöntem kullanarak net ve keskin kenarları algılar. En popüler kenar algılama tekniklerinden biridir.

2. Sobel Kenar Algoritması: Görüntünün yatay ve dikey kenarları ayrı ayrı algılar ve sonra bunları birleştirir.

3. Prewitt Kenar Algoritması: Sobel algoritmasına benzer şekilde x ve y eksenlerinde kenarları algılar. Daha basit bir filtreden faydalanır. Kenarları belirlemek için Prewitt matrisleri kullanılır.

Canny algoritması, düşük hata oranı ve yüksek doğruluğu nedeniyle literatürde en çok tercih edilen yöntem olarak ön plana çıkmaktadır.

Literatür araştırmalarımızda görüyoruz ki çoğu çalışmada, yalnızca sabit görüntüler üzerinde analiz yapılmıştır. Gerçek zamanlı video işleme konusu sınırlı sayıda ele alınmıştır.
Araçların park durumunun (Örneğin, yarı dolu veya yanlış park) analizi genellikle ihmal edilmiştir.

Bu çalışma, kenar algılama yöntemleri ile park alanı tespiti konusunda literatürdeki eksiklikleri gidermeye yönelik tasarlanmıştır. Görüntü işleme tabanlı bu yöntemler, akıllı otopark sistemlerinde önemli bir potansiyele sahiptir ve gerçek zamanlı trafik yönetiminde etkili bir çözüm sunmaktadır.

2. Önerilen Yöntem

Kenar Bulma Yöntemleri ile Park Alanı Tespiti projemizde bu algoritmalardan faydalanabilmek için Python programlama dilini ve Jupyter Notebook geliştirme ortamını kullandık.

Python: Web uygulamaları, yazılım geliştirme, veri bilimi ve makine öğreniminde yaygın olarak kullanılan bir programlama dilidir. Geliştiriciler, etkili ve öğrenmesi kolay olduğu ve birçok farklı platformda çalıştırılabildiği için bu dili kullanır. Ücretsiz olarak indirilebilir, her türlü sistemle iyi bir entegrasyon sağlar ve geliştirme hızını artırır.

Avantajları:

- Geliştiriciler, basit ve İngilizceye benzer bir söz dizimine sahip olduğundan Python dilinde yazılmış programı kolayca okuyabilir ve anlayabilir.
- Geliştiriciler, diğer birçok dile kıyasla çok daha az kod satırıyla bir Python programı yazabildiğinden, geliştiricilerin daha üretken olmasını sağlar.
- Python, neredeyse her görev için yeniden kullanılabilir kodları içeren geniş bir standart kitaplığa sahiptir. Geliştiricilerin kodu sıfırdan yazmasına gerek kalmaz.
- Geliştiriciler Python'ı Java, C ve C++ gibi diğer popüler programlama dilleriyle kolayca kullanabilir.
- Bir sorunla karşılaşırsanız Pyhton topluluğundaki ekiplerden hızlıca destek alabilirsiniz.
- Python'ı öğrenmek için internette birçok yararlı kaynak bulunmaktadır. Kolayca videolar, dersler, belgeler ve geliştirici kılavuzları bulabilirsiniz.
- Windows, macOS, Linux ve Unix gibi farklı bilgisayar işletim sistemleri arasında taşınabilir.

Python dili içerisinde çeşitli kütüphanelerden faydalandık.

1. OpenCV(cv2): Görüntü işleme ve video oynatma işlemleri için kullanılır.

2. Pickle: Kullanıcı tarafından seçilen noktaların kaydedilmesi ve daha sonra yeniden yüklenmesi için kullanılır.

3. NumPy: Görüntü işleme ve dilatasyon işlemlerinde yardımcı olan matematiksel hesaplamalar için kullanılır.

Jupyter Notebook: Veri bilimi, makine öğrenimi, veri analizi ve bilimsel hesaplamalar gibi alanlarda yaygın olarak kullanılan, etkileşimli bir not defteri uygulamasıdır. Bu araç, programcıların ve veri bilimcilerin verileri keşfetmelerine, analiz etmelerine, sonuçlarını görselleştirmelerine olanak tanır. Hem kod hem de metin içeriğini bir arada sunabilmesi, Jupyter Notebook'u diğer geliştirme araçlarından ayıran en önemli özelliklerden biridir.

3. Geliştirilen Yazılımın Tanıtımı

Kenar Bulma Yöntemleri ile Park Alanı Tespiti projesi için aşağıdaki başlıklar uygulanmıştır.

- Grayscale Dönüşümü (cv2.cvtColor): Renkli bir görüntüyü gri tonlamaya dönüştürerek işlenmesi daha kolay bir hale getirir.
- Gaussian Blur (cv2.GaussianBlur): Görüntüyü yumuşatarak parazitleri azaltır.
- Adaptive Thresholdin (cv2.adaptiveThreshold): Görüntüyü ikili hale dönüştürerek (siyah-beyaz), belirgin kenarları vurgular.
- Median Blur (cv2.medianBlur): Görüntüdeki gürültüyü azaltır.
- Dilasyon (cv2.dilate): Görüntüdeki beyaz alanları genişleterek şekilleri daha belirgin hale getiriyor.
- Görüntü Kesme (frame[y:y + 15, x:x + 60]): İlgili alanları keserek bu alanlar üzerinde işlem yapılmasını sağlar.
- Piksel Sayma (cv2.countNonZero): Belirli bir alandaki beyaz piksellerin sayısını hesaplayarak alanın dolu ya da boş olduğunu tespit eder.
- Video Okuma (cv2.VideoCapture): Bir video dosyasını okuyarak kare kare işleme hazır hale getirir.
- Kare Yeniden Boyutlandırma (cv2.resize): Video karesini görüntünün boyutuna göre yeniden ölçeklendirir.
- Kare İşleme: Her kareye görüntü işleme teknikleri uygulanarak işlenir.
- Kare Görüntüleme (cv2.imshow): İşlenen her kare ekranda gösterilir.

- Fare Callback Kullanımı(cv2.setMouseCallback): Kullanıcının tıklamalarını algılayarak resim üzerinde işlem yapmasını sağlar:
    - Sol tık ile nokta ekleme.
    - Sağ tık ile nokta silme.
- Çizim (cv2.rectangle): Kullanıcının belirttiği noktalara dikdörtgen çizer.
- Dikdörtgen Çizme (cv2.rectangle): Park alanlarını belirginleştirilir.
- Bilgi Gösterimi (cv2.putText): Park alanlarının durumu ekrana yazılır.

Kenar Bulma Yöntemleri ile Park Alanı Tespiti projemizin adımları ise şu başlıklardan oluşmaktadır:

Başlangıçta Noktaların Yüklenmesi: İlk olarak, daha önce kaydedilmiş olan nokta bilgileri bir dosyadan (n) yüklenir. Eğer dosya bulunamazsa, liste boş olarak başlatılır.

Resim ve Video Yükleme: Program, bir görsel dosyasını yükler ve boyutlarını alır.

Kullanıcı Etkileşimi: Kullanıcı, resim üzerinde fare ile sol tıklayarak dikdörtgen bölgeleri ekler veya sağ tıklayarak bu bölgeleri siler. Bu noktalar, liste adında bir listede tutulur ve dosyaya kaydedilir.

Görüntü İşleme: Görüntüler, gri tonlama, bulanıklaştırma, adaptif eşikleme ve dilatasyon işlemleri gibi çeşitli ön işleme adımlarından geçirilir.

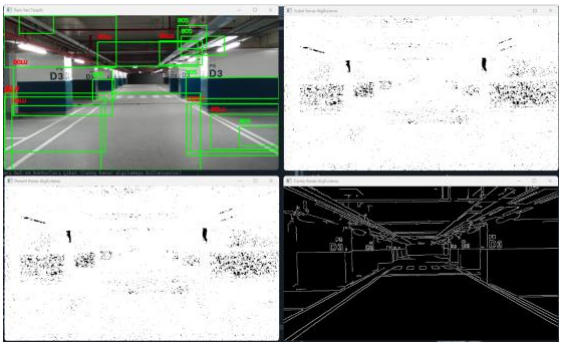
Alan Kontrolü: Her bir dikdörtgen bölgesi, cv2.countNonZero() fonksiyonu ile kontrol edilir. Eğer bölge içindeki yoğunluk değeri 150'den düşükse bu alan "boş" olarak kabul edilir, aksi takdirde "dolu" olarak işaretlenir.

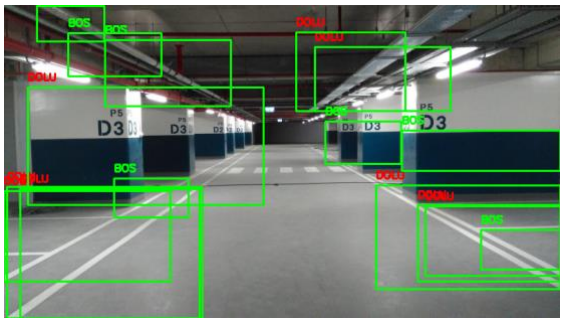
Sonuçların Görüntülenmesi: Her bir dikdörtgenin üzerine, boş ya da dolu olduğu bilgisi renk ile gösterilir: Yeşil renk boş, kırmızı renk ise dolu anlamına gelir. Ayrıca, ekranın sol üst kısmında boş alanların sayısı görüntülenir.

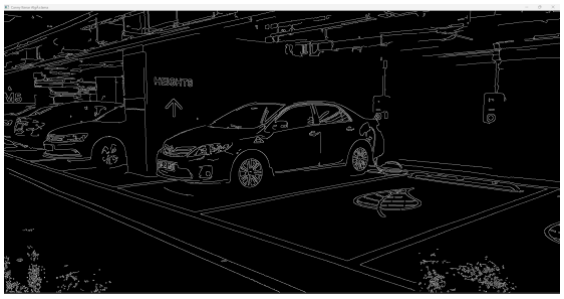
Program Sonlandırma: Kullanıcı, programı q tuşuna basarak sonlandırabilir.

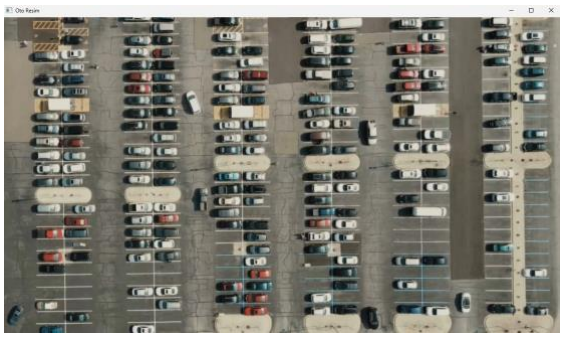
4. Deneysel Sonuçlar

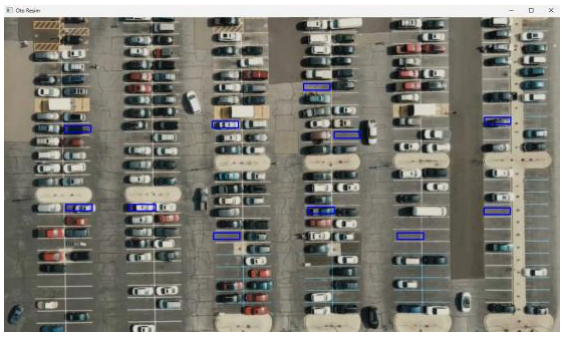
Kenar Bulma Yöntemleri ile Park Alanı Tespiti projemizin çeşitli çıktıları aşağıdadır.

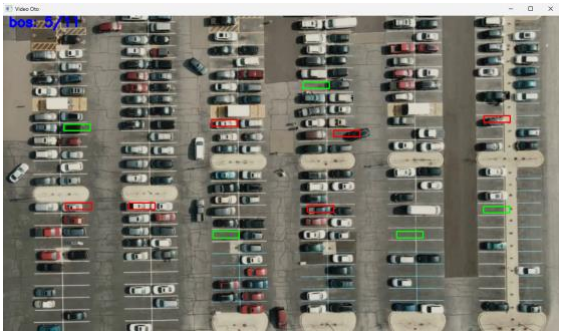


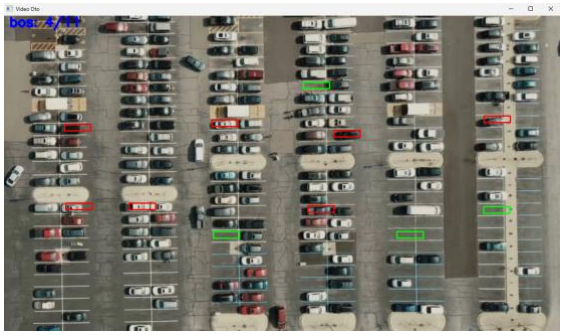












5. Tartışma ve Sonuçlar

Bu çalışmada, belirli bölgelerin video akışında izlenmesi ve analiz edilmesi amacıyla bir sistem geliştirilmiştir. Sistemde, kullanıcı tarafından seçilen noktalar üzerinden video kareleri işlenerek, her bir bölgedeki boşluk veya doluluk durumu belirlenmiştir. Sol tıklama ile nokta eklenip sağ tıklama ile nokta silinmesi sağlanarak, kullanıcı etkileşimiyle sistemin dinamik olarak güncellenebilmesi sağlanmıştır. Çalışma sırasında kullanılan temel teknikler arasında görüntü işleme, adaptif eşikleme, bulanıklaştırma, medyan bulanıklaştırma ve dilatasyon işlemleri yer almıştır.

Elde edilen sonuçlar, seçilen alanlarda boşlukların belirlenmesinde başarılı olmuştur. Boş alanlar yeşil, dolu alanlar ise kırmızı renkle işaretlenerek, her karedeki boş alan sayısı ekranda görsel olarak gösterilmiştir. Sistem, farklı video içeriklerinde de doğru sonuçlar verecek şekilde esnek bir yapıya sahiptir. Ancak, video çözünürlüğü, kare hızı ve ışık koşulları gibi faktörler görüntü işleme sonuçlarını etkileyebilir. Özellikle düşük çözünürlüklü veya düşük kaliteli video verilerinde, işlenen karelerin doğruluğu düşebilir. Bu durum, ileride yapılacak iyileştirmelerde dikkate alınarak, sistemin daha sağlam ve doğru hale getirilmesi mümkün olacaktır.

Proje, video izleme ve analiz süreçlerinde, kullanıcı etkileşimiyle anlık değişiklik yapabilen bir sistem önerisi sunmaktadır. Ayrıca, görüntü işleme teknikleriyle boşluk algılama ve doluluk tespiti üzerine yapılan bu tür çalışmaların, güvenlik izleme, fabrika izleme, trafik analizi gibi çeşitli alanlarda uygulanabilirliği bulunmaktadır. Literatüre katkı sağlamak adına, geliştirilen sistemin daha geniş veri setleriyle test edilmesi ve daha karmaşık algoritmalarla desteklenmesi, verimliliği ve doğruluğu artırabilir.

5.1 Literatüre Katkılar

Projenin literatüre katkıları şu şekildedir:

Görüntü işleme tekniklerinin etkin kullanımı: Çalışmada, adaptif eşikleme, medyan bulanıklaştırma ve dilatasyon gibi teknikler bir arada kullanılarak, video karelerinin daha güvenilir şekilde analiz edilmesi sağlanmıştır.

Kullanıcı etkileşimiyle dinamik güncelleme: Kullanıcıların video üzerinde nokta ekleyip silebilmesi, gerçek zamanlı ve dinamik video analiz sistemlerinin geliştirilmesinde yenilikçi bir yaklaşım sunmaktadır.

Video izleme alanında uygulama potansiyeli: Çalışma, video akışları üzerinde boşluk algılama ve doluluk tespiti ile ilgili potansiyel uygulama alanlarını keşfetmektedir. Özellikle güvenlik, endüstriyel izleme ve trafik analizi gibi alanlarda kullanılabilir.

6. Proje Github Linkleri

211307048 Buse Yener -

https://github.com/buseyener/ParkAlanTespiti

211307089 Nisa Aksoy –

https://github.com/nisaaksoy/ParkAlan-Tespiti

Kaynakça

1. https://docs.python.org/3/
2. Bayram, S., & Gönen, M. (2016). Görüntü İşleme ve Bilgisayarla Görme. Seçkin Yayıncılık.
3. https://opencv.org/
4. https://numpy.org/
5. https://docs.python.org/3/library/pickle.html
6. https://docs.opencv.org/4.x/d2/de8/group__core__array.html
7. https://mesutpiskin.com/blog/wp-content/uploads/2017/01/OpenCV%20Kitap.pdf
8. https://senolomer0.medium.com/opencv-i%CC%87le-g%C3%B6r%C3%BCnt%C3%BC-i%CC%87%C5%9Fleme-1-d7879e2386e1