PLAKA TANIMA SİSTEMİ

YUNUS BERKAY FİDAN / 193405158
YİĞİT CANPOLAT / 213405002

Giriş:

- Araba plakası tanıma, trafik güvenliği, otomatik park sistemleri ve güvenlik kontrolleri gibi alanlarda önemli bir uygulamadır.
- Bu sunumda, verilen bir araba görüntüsünden plaka numarasını tanımak için geliştirilmiş bir programı tanıtacağız.

Proje Amacı:

- Bu projenin amacı, görüntü işleme ve metin tanıma tekniklerini kullanarak bir araba plakasını otomatik olarak tanımlamaktır.
- Kodumuz, OpenCV ve pytesseract gibi araçları kullanarak bu görevi gerçekleştirir.

Kod Açıklaması:

1. Gerekli kütüphaneleri ve araçları projeye dahil ediyoruz.
    1. cv2: Görüntü işleme için OpenCV kütüphanesi.
    2. imutils: OpenCV'ye yardımcı fonksiyonlar sağlayan bir yardımcı kütüphane.
    3. numpy: Dizi işlemleri için kullanılan bir kütüphane.
    4. pytesseract: Metin tanıma için kullanılan bir araç.
2. Görüntüyü okuyor ve boyutunu yeniden ayarlıyoruz.
    1. cv2.imread() işlevi, belirtilen görüntüyü okur.
    2. cv2.resize() işlevi, görüntünün boyutunu belirli bir boyuta dönüştürür.

Kod Açıklaması:

1. Görüntüyü gri tonlamaya dönüştürüyoruz.
    1. cv2.cvtColor() işlevi, BGR görüntüyü gri tonlamalıya dönüştürür.
2. Gürültüyü azaltmak için görüntüye bilateral filtre uyguluyoruz.
    1. cv2.bilateralFilter() işlevi, gürültüyü azaltmak ve kenarları korumak için bir bilateral filtre uygular.
3. Kenarları tespit etmek için Canny kenar tespiti uyguluyoruz.
    1. cv2.Canny() işlevi, kenarları tespit etmek için Canny kenar tespiti algoritmasını uygular.

Kod Açıklaması:

1. Konturları buluyoruz.
    1. cv2.findContours() işlevi, görüntüdeki konturları bulur.
    2. imutils.grab_contours() işlevi, konturları alır ve liste olarak döndürür.
    3. Konturları alanlarına göre sıralayarak en büyük 10 konturu seçiyoruz.
2. Plaka konturunu buluyoruz.
    1. Konturlar üzerinde döngüye girerek plaka konturunu buluyoruz.
    2. cv2.arcLength() işlevi, bir konturun çevresini hesaplar.
    3. cv2.approxPolyDP() işlevi, konturu yaklaşık olarak temsilendir.

Kod Açıklaması:

8. Plaka konturu bulunduysa, çizim işlevlerini kullanarak plaka konturunu görselleştiriyoruz.
9. Plaka alanını maskelemek için kontur kullanarak bir maske oluşturuyoruz.
    8. np.zeros() işlevi, gri tonlamalı görüntünün boyutunda bir sıfırlar dizisi oluşturur.
    9. cv2.drawContours() işlevi, konturu kullanarak maskeyi oluşturur.
    10. cv2.bitwise_and() işlevi, maskeyi görüntüye uygulayarak plaka alanını alır.

Kod Açıklaması:

8. Maske üzerinde belirli bir eşiği geçen pikselleri seçerek plaka bölgesini kırpıyoruz.
9. Kırpılan plaka bölgesini metne dönüştürmek için pytesseract kullanıyoruz.
    8. pytesseract.image_to_string() işlevi, plaka görüntüsündeki metni tanır.
10. Tanınan plaka numarasını ekrana yazdırıyoruz.
11. Sonuçları görselleştiriyoruz.
    8. cv2.imshow() işlevi, orijinal görüntüyü ve kırpılan plaka görüntüsünü görüntüler.
12. Kullanıcının herhangi bir tuşa basmasını bekliyor ve görüntü pencerelerini kapatıyoruz.
    8. cv2.waitKey() işlevi, kullanıcının bir tuşa basmasını bekler.
    9. cv2.destroyAllWindows() işlevi, görüntü pencerelerini kapatır.

Kazanımlar:

- Bu proje sayesinde, görüntü işleme, kenar tespiti, kontur analizi ve metin tanıma gibi konularda deneyim kazandık.
- Geliştirdiğimiz kodun gerçek dünya uygulamalarında nasıl kullanılabileceğini gördük.
- Proje aşamalarında kendi kendimize öğrenme süreci yaşadık ve kendimizi bu alanda geliştirdik.

Sonuç:

- Araba plakası tanıma programımız, verilen bir araba görüntüsünden plaka numarasını başarıyla tanıyabiliyor.
- Proje, otomatik plaka tanıma sistemleri, trafik izleme sistemleri, güvenlik uygulamaları gibi birçok alanda kullanılabilir.
- Geliştirme potansiyeli olan bir proje olup, daha da optimize edilebilir ve farklı görüntü koşullarında daha iyi performans sağlayacak şekilde geliştirilebilir.

Kaynaklar:

- OpenCV Resmi Dokümantasyonu
- pytesseract Resmi Dokümantasyonu
- Programming Fever