



BARTIN ÜNİVERSİTESİ

MÜHENDİSLİK, MİMARLIK ve TASARIM FAKÜLTESİ

BİLGİSAYAR MÜHENDİSLİĞİ

BSM423 – GÖRÜNTÜ İŞLEMENİN TEMELLERİ

PYTHON İLE PLAKA TANIMA PROJESİ

Proje Raporu – Grup 13

04.01.2022

YASİN KARAKUŞ – YUSUF DEMİRÖZ

18010310014 - 18010310026

yasin.karakus@ogrenci.bartin.edu.tr – yusuf.demiroz@ogrenci.bartin.edu.tr

İÇİNDEKİLER

ÖZET

1. GİRİŞ

2. YÖNTEM

2.1. Görüntü Ön İşleme

2.1.1.RGB Görüntünün Alınması

2.1.2.RGB'den Gri Seviyeli Görüntüye Dönüştürme İşlemi

2.1.3.Gürültü Temizleme (Bulanıklaştırma)

2.2. Kenar Bulma (Canny Edge) Algoritması

2.2.1.Görüntünün Bulanıklaştırılması

2.2.2.Görüntünün Yoğunluk Gradyanının Hesaplanması

2.2.3.Maksimum Olmayan Bastırma (Non Maximum Supression - NMS)

2.2.4.Görüntü Çift Eşikleme (Double Thresholding) İşlemi

2.2.5.Görüntü Üzerinde Kenar İzleme (Hysteresis) İşlemi

2.3. Kontur Uygulaması ve Plakanın Ayrıştırılması

2.4. Tesseract ile Plakadan Optik Karakter Tanıma

2.5. Plaka Verilerinin Kaydedilmesi

3. SONUÇ VE DEĞERLENDİRME

4. REFERANSLAR

ÖZET

Bu çalışmada, görüntü işleme algoritmaları kullanılarak araç üzerindeki plakayı bulan bir sistem geliştirilmiştir. Güvenlik sistemi gereken yerler, istatistiksel verilerin gerektiği yerler, ortak kullanım alanları, otopark giriş ve çıkışları, trafik kontrolü, üniversite giriş-çıkışları, site giriş ve çıkışları gibi araç sayısının yoğun olduğu muhtelif yerlerde kullanılmaktadır. Bu sistem sayesinde insan gücü, maliyeti ve güvenlik tehditlerini en aza indirmek amaçlanmaktadır. Geliştirilen sistem Python programlama dili ve açık kaynak olan OpenCV 4.0.1 (Açık Kaynak Bilgisayarlı Görme Kütüphanesi) ve Tesseract-OCR 0.3.8 (Optik Karakter Tanıma Kütüphanesi) kullanılarak gerçekleştirilmiştir. Plaka tanıma sistemi test edilirken farklı pozisyonlardan alınan ve farklı ortamlardan alınan görüntüler üzerinde denenmiştir.

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

1. GİRİŞ

Ülkemizde trafiğe kayıtlı araç sayısı, 2021 yılı şubat ayında 24 milyon 328 bin olarak belirtilmiştir ve bu sayı gün geçtikçe artış sağlamaktadır [1]. Bu araçların sadece insan gücüyle kontrol edilmesi neredeyse imkânsız hale gelmiştir. Bu zorluğun üstesinden gelmek için çeşitli teknolojiler üretilmiştir. Bu teknolojilerinden birisi olan ve kendi içerisinde de çok farklı tekniklerle gelişime uğramış olan plaka tanıma sistemi; günümüzde trafik kontrolünü sağlamak, otoparklarda hizmete dair ücretlendirme yapmak, ülke sınırlarında ülkeye geçiş izni kontrolleri yapmak gibi araç giriş ve çıkışlarının yoğun olduğu muhtelif yerlerde kullanılmaktadır [2]. Plaka tanıma sistemi insan gücünü en aza indirmekte, 24 saat boyunca aralıksız çalışabilmekte, güvenlik düzeyini artırarak hayatımızı kolaylaştırmaktadır.

Araç plakaları ülkeden ülkeye farklılıklar göstermektedir. Şekil 1’de projede kullanılan bazı ülkelere ait plaka örnekleri vardır. Bu plakalar arasındaki benzerlik karakterler arasındaki sabit kontrast olmasıdır. Arka plan boyutları ülkeden ülkeye değişmektedir.



Şekil 1 Ü ülkelere göre araç plakası örnekleri (ABD, Bulgaristan ve Türkiye)

Ülkemize plakalar Karayolları Yönetmeliği tarafından belirlenen özellikler içermektedir. Plaka gövdesi alüminyumdan yapılmaktadır. Boyutları taksi, otobüs, minibüs, kamyon gibi araçlarda 11x52 cm’dir. Araçların hem ön kısmında hem de arka kısmında plaka bulunması zorunludur. Plaka beyaz bir tabaka üzerinde tanımlanmış ve üzerinde siyah karakter üzerine yazılmış bir biçimdedir. Bu projedeki amaç Türk plakalı sivil araçların plakalarını tespit ederek üzerinde bulunan karakterleri veri tabanına kaydetmesidir.

2. YÖNTEM

Plaka tanıma sistemi üç aşamadan oluşmaktadır. Birinci aşama ilk olarak araç üzerinde bulunan plakanın ayrıştırılma işlemidir. İkinci aşama ise ayrıştırılan plaka üzerindeki karakterlerin ayrıştırılması işlemidir. Karakterleri ayrıştırmak için karakterlerin başlangıç ve bitiş noktaları kontrol edilir [3]. Plaka üzerinde gerekli olan morfolojik işlemler uygulanarak bu işlem gerçekleştirilir. Üçüncü aşamada ise tespit edilen plakaya ait karakterler veritabanına kaydedilmektedir.

2.1. Görüntü Ön İşleme

Kullanıcıdan alınan görüntü ham bir şekilde gelmekte ve RGB renk uzayında bulunmaktadır. RGB (Red, Green, Blue) renk uzayında bulunan resim düzgün bir şekilde işlenebilmesi için gerekli olan bazı dönüşümler yapmak gerekir.[4]

2.1.1. RGB Görüntünün Alınması

Plaka tanıma sisteminin uygulanıp plakanın tespit edileceği araç görselinin elde edilmesi gerekmektedir. Kullanıcıdan fotoğraf yüklemesi yapılması istenerek ya da herhangi bir kamera aracılığıyla bu işlem gerçekleştirilebilir.



Şekil 2 RGB Seviyeli İşlenmemiş(Ham) Görüntü

2.1.2. RGB'den Gri Seviyeli Görüntüye Dönüştürme İşlemi

Bir RGB görüntüyü gri seviyeli görüntüye çevirme işlemi görüntü üzerindeki işlemleri daha basit ve daha kolay bir şekilde gerçekleştirmek üzere yapılır. Gri seviyeli bir görüntü elde etmek için ağırlıklı yöntem (weighted method)

kullanılmıştır. Bu yöntemde kırmızı kanalın yaklaşık %30'u, yeşil kanalın yaklaşık %59' u ve mavi kanalın yaklaşık %11'lik kısmı alınmaktadır.



Şekil 3 RGB Seviyeden Gri Seviyeye Dönüştürülmüş Görüntü

2.1.3. Gürültü Temizleme (Bulanıklaştırma)

Plaka tanıma sistemde plaka kenarlarının net bir biçimde ortaya çıkmasını sağlamak ve kaybolan kenarları netleştirmek ortaya çıkarmak için bilateral filtre kullanılmıştır [5].

Bilateral filtre, görüntüler için doğrusal olmayan, kenar koruyucu ve gürültü azaltıcı bir yumuşatma filtresidir. Her pikselin yoğunluğunu, yakındaki piksellerden gelen ağırlıklı ortalama yoğunluk değerleriyle değiştirir. Bu ağırlık bir Gauss dağılımına dayalı olabilir. En önemlisi, ağırlıklar yalnızca piksellerin Öklid mesafesine değil, aynı zamanda radyometrik farklılıklara da bağlıdır (örneğin, renk yoğunluğu, derinlik mesafesi, aralık farklılıkları vb). Bu da keskin kenarları korur.

2.2. Kenar Bulma (Canny Edge) Algoritması

2.2.1. Görüntünün Bulanıklaştırılması

Kenar tespit etmede görüntüyü bulanıklaştırmak için genellikle Gauss Filtresi kullanılır lakin biz Bilateral Filtre kullandık. Bilateral Filtreye ait detaylar bir üst maddede yer almaktadır.

2.2.2. Görüntünün Yoğunluk Gradyanının Hesaplanması

Gradyan algılayıcı, kenar algılama operatörlerini kullanarak kenarın yönünü ve yoğunluğunu algılar. Kodlaması ve anlaşılması çok kolay olduğu için Sobel operatörünü kullandık. Piksel yoğunluğunun değişimini tespit etmek için hem X eksenini hem de Y

ekseninde Sobel filtreleri uyguladık.

2.2.3. Maksimum Olmayan Bastırma (Non Maximum Supression - NMS)

NMS, bir kenar inceltme tekniğidir. Kenar yönü hesaplandıktan sonra bunları görüntüde takip edebiliriz. Bir piksele bakarken, çevreleyen pikselleri ters trigonometrik yönde kontrol ederken yalnızca dört yön vardır:

- 0 derece => yatay yön
- 45 derece => pozitif köşegen
- 90 derece => dikey yön
- 135 derece => negatif köşegen



Şekil 4 NMS Uygulamasını İzah İçin Örnek Görsel

Örneğin görüntüdeki (x, y) noktasındaki pikseli analiz ediyoruz. Hangi komşuları kontrol edeceğimizi belirlemek için $[x, y]$ açısını yaklaşık olarak bulmalıyız. Yukarıdaki şema, yaklaşımları anlamanıza yardımcı olabilir.

- Kırmızı alan $[0, 22.5)$ ve $[157.5, 180]$ aralıklarını kapsıyor => piksel $(x, y-1)$ ve piksele $(x, y+1)$ bakıyoruz
- Mavi alan $[22.5, 67.5)$ => piksele $(x-1, y-1)$ ve piksele $(x+1, y+1)$ bakıyoruz.
- Sarı alan $[67.5, 112.5)$ => piksele $(x-1, y)$ ve piksele $(x+1, y)$ bakıyoruz.
- Yeşil alan $[112.5, 157.5)$ => piksele $(x+1, y-1)$ ve piksele $(x-1, y+1)$ bakıyoruz.

Mevcut piksel komşularının her ikisinden de büyükse, onu tutarız. Sonuç olarak, algoritma daha ince kenarlar alır.

2.2.4. Görüntü Çift Eşikleme (Double Thresholding) İşlemi

NMS uygulandıktan sonra, kalan pikseller gerçek kenarların daha doğru bir temsilidir, ancak yine de parazit ve renk varyasyonları tarafından oluşturulan bazı pikseller vardır. Bunlardan kaçınmak için görüntüyü filtrelemek zorunludur. Bu, tüm pikselleri 2 kategoriye (zayıf ve güçlü) ayırarak gerçekleştirilir. Bir pikselin değeri bizim üst eşiklerimizden büyükse, onu güçlü olarak kabul ederiz. Piksel, düşük eşikten daha parlak

ancak yüksek eşikten daha koyuysa, pikseli zayıf olarak kabul ederiz. Alt eşikten daha küçük olan pikseller yok sayılır.

2.2.5. Görüntü Üzerinde Kenar İzleme (Hysteresis) İşlemi

Görüntü en son haliyle zayıf piksellerden ve güçlü piksellerden oluşmaktadır. Tüm gerçek kenar piksellerinin güçlü yapılması ve kenar olmayan tüm pikselleri sıfıra dönüştürmelisi gerekmektedir. Bunun için de zayıf bir piksel, yalnızca 8 yönden herhangi birinde güçlü bir komşusu varsa, güçlü bir piksele dönüştürülür.



Şekil 5 Canny Edge Algoritması Uygulandıktan Sonra Elde Edilen Kenarlar

2.3. Kontur Uygulaması ve Plakanın Ayrıştırılması

2.3.1. Görüntü Üzerine Kontur Uygulanması

Görüntü üzerinde gereken işlem kontur uygulayarak aranan plaka bölgesini işaretlemektir. [8] Konturlar, aynı renk veya yoğunluğa sahip olan tüm kesintisiz noktaları (sınır boyunca) birleştirir. Konturlar, şekil analizi, nesne algılama ve tanıma için kullanışlıdır. Kontur yapmadan önce kenar bulma ve genişletme işlemi uygulayarak, konturun doğru sonuç vermesini sağlar.



Şekil 6 Kontur Uygulamasından Sonra Birleştirilen Kenarların Orijinal Görüntü Üzerinde Gösterimi

2.3.2. Görüntü Üzerinden Plakanın Ayrıştırılması

Görüntü üzerinden plakayı ayırştırmak için maskeleme işlemi yapılır. Maskeleme yaparak bundan sonraki işlemlerde sadece plaka üzerinde işlemler gerçekleştirilir.



Şekil 7 Görüntü Üzerinden Elde Edilen Plaka

2.4. Tesseract ile Plakadan Optik Karakter Tanıma

Optik karakter tanıma (Optical Character Recognition, OCR) bir fotoğrafta, basılı bir belgede, el yazısıyla yazılmış bir metinde yer alan yazıların bilgisayar ortamına aktarılması için kullanılmaktadır. Bu tip dosyalar, OCR yazılımları tarafından işlenerek metin dosyalarına çevrilir.

OCR yazılımlarından en çok tercih edilen Tesseract, 1985-1994 yılları arasında HP tarafından geliştirilmiştir. 2005 senesinde açık kaynak haline çevrilmiş ve 2006'dan beri ise Google tarafından geliştirilmeye devam edilmektedir.

Tesseract plaka üzerindeki lisans numaralarını yakalamak için kullandığımız algoritmadır. Görüntü üzerinde sayıları ve harfleri vurgulayarak görüntüde bulunan diğer nesnelerden ayırmaya çalışır.[9] Karakter tanıma işlemi ile otomatik plaka tanıma, verileri düzenlenebilir hale getirme, aranabilir ve kolayca saklanabilir bilgilere dönüştürür. Plaka tanıma işleminde en kritik ve sonuç sürecidir.

2.5. Plaka Verilerinin Kaydedilmesi

Çeşitli sayısal hesaplamalar ve optik karakter tanıma algoritmamızın elde ettiği sonucun kaybolmaması, işlenmek üzere saklanması için xlsxwriter kütüphanesi ile sonuçlar excel tablosuna kaydedilmektedir.

3. SONUÇ VE DEĞERLENDİRME

Oluşturmuş olduğumuz sistemi 9 farklı araç görüntüsü içerisinde teste tabi tuttuğumuzda yaklaşık %88 plaka tanıma oranı elde edilmiştir. Plakadan optik karakter okumada ise PyTesseract yetersiz kalmıştır.

4. REFERANSLAR

- 1- Türkiye'deki Kayıtlı Araç Sayısı, <https://www.dunya.com/ekonomi/subatta-trafikteki-arac-sayisi-72-bin-artti-haberi-615403>
- 2- Barroso J., Rafael A., Dagless E. L and BulasCruz J., Number plate reading using computer vision, IEEE - International Symposium on Industrial Electronics ISIE'97, Universidade do Minho, Guimarães, Julho, Portugal, 1997.
- 3- Christopher J. S., Applications Of Computer Vision to Road Traffic Monitoring, PhD Thesis, University of Bristol, UK, 1997.
- 4- B.D. Acosta, Experiments in image segmentation for automatic US license plate recognition, M.Sc. thesis, Department of Computer Science, Faculty of the Virginia Polytechnic Institute and State University, 2004.
- 5- C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271), 1998, pp. 839-846, doi: 10.1109/ICCV.1998.710815.
- 6- Luciano da Fontoura Costa and Roberto Marcondes Cesar Jr., Shape Analysis and Classification, CRC Press, 2001, Chap 3.
- 7- Liang, J. I., Jim Piper, and Jing-Yan Tang. "Erosion and dilation of binary images by arbitrary structuring elements using interval coding." Pattern Recognition Letters 9.3 (1989): 201-209.
- 8- Suzuki, Satoshi. "Topological structural analysis of digitized binary images by border following." Computer vision, graphics, and image processing 30.1 (1985): 32-46.