

iOS Platformunda Mobil Trafik Ceza Bildirim Uygulaması

İbrahim Şanlıalp¹, Akif Kutlu², İbrahim Arda Çankaya², Asım Sinan Yüksel²,
1 Ahi Evran Üniversitesi, Bilgisayar Mühendisliği Bölümü, Kırşehir

2 Süleyman Demirel Üniversitesi, Bilgisayar Mühendisliği Bölümü, Isparta

ibrahim.sanlialp@ahievran.edu.tr, akifkutlu@sdu.edu.tr, ardacankaya@sdu.edu.tr, asimyuksel@sdu.edu.tr

Özet: Mobil cihazların kullanımı her geçen gün artmaktadır. Geliştirilen mobil uygulamalar sayesinde de insanlar her türlü işlerini mobil cihazlarına yükledikleri uygulamalar vasıtasıyla gerçekleştirebilmektedir. Bu nedenle günümüzde mobil uygulama geliştirmek büyüyen bir trend olarak ilerlemektedir. Popülerliğini ve pazar payını gün geçtikçe artıran Apple iOS platformu mobil uygulama geliştirmeyi sağlayan bu platformlardan biridir. iOS platformunda gerçekleştirilecek olan bu çalışmanın amacı, trafik polislerinin ceza işlemlerini kolaylaştıran, Türk sivil araçlardan alınan plaka resimlerini OpenCV görüntü işleme kütüphanesi kullanılarak resimdeki anlamlı bilgileri çıkarıp işleyen ve sonucunda elde edilen karakterleri plaka bilgisine dönüştürdükten sonra, gerekli ceza bilgilerinin düzenlenerek araç sahiplerine web servis aracılığıyla bilgilendirilen bir sistemin tasarımı ve gerçekleştirimi hedeflenmiştir.

Anahtar Kelimeler: Mobil Uygulamalar, Araç Plaka Tanıma Sistemi, Trafik Bilgi Sistemleri, Karakter Tanıma

Mobile Traffic Fine Notification Application in iOS Platform

Abstract: Mobile device usage increases day by day. People are able to perform any work by using mobile applications that they installed on their smart devices. For this reason, today, mobile application development has become a growing trend. Apple's iOS platform that is increasing its popularity and market share is one of these platforms that enables mobile application development. The aim of this study that will be achieved in iOS platform is design and implement a system that facilitates the fine processing operations of traffic police, extracts and processes the meaningful data from the license plate images that were taken from Turkish civil vehicles by using OpenCV image processing library, and then converting the characters to license plate data, finally generating traffic fine information and notifying vehicle owners via web service.

Keywords: Mobile Applications, Vehicle License Plate Recognition System, Traffic Information Systems, Character Recognition

1. Giriş

Mobil uygulamalar tablet bilgisayarlar, akıllı telefonlar, el terminalleri gibi mobil cihazlarda çalışmaları için tasarlanmış yazılımların genel adıdır. Bu uygulamalara dünyada kullanıcıların ihtiyaçları arttıkça birlikte akıllı telefonları giderek önemli bir noktaya getirmiştir. Mobil platformlarda çalışmalar gerçekleştiren programcılar gelişen yazılımlar sayesinde mobil telefonlar aracılığıyla sosyal medya ortamına değişik işlevsel ve görsel uygulamalar geliştirebilmektedirler (1).

Akıllı mobil telefon kullanımının hızla artması ve internet'in hızla gelişmesi sonucunda mobil uygulamalar hayatımızın vazgeçilmez bir parçası haline gelmiştir. Çok pratik olmaları aynı zamanda da çoğunluğu zaman ve maliyet açısından ekonomik oldukları için kullanıcı sayısı da kısa zamanda büyük miktarda artmıştır.

Mobil uygulama geliştirme platformlarından biri olan ilk adıyla iPhone OS olan iOS, başlangıçta Apple firmasının iPhone için geliştirdiği fakat sonraları iPad ve iPod Touch gibi cihazlarda da kullanılmaya başlandığı Mac OS X'den türetilmiş mobil işletim sistemidir. iOS içinde Core Servisleri tabakası, Core OS tabakası, Cocoa Touch tabakası ve Medya tabakası olan 4 katman bulunmaktadır (2). iOS platformunda uygulama geliştirmek için kullandığımız Objective-C dili ise, ANSI C'ye nesneye yönelik programlama özellikleri kazandırılarak geliştirilmiş bir dildir. Bu ilaveler ilk nesneye yönelik programlama dillerinden biri olan Smalltalk dilindeki yapılarla benzer. Objective-C, GNU ve NeXTSTEP C derleyicileri tarafından da desteklenmektedir. Objective-C aslında oldukça eskidir ve 1980'lere kadar uzanır ancak günümüzde Apple firmasının Mac sistemleri ve iPhone uygulamaları alanında kullanılmaya başlanınca çok popüler hale gelmiştir. Objective-C ilk kez Brad Cox tarafından, ken-

di şirketi bünyesinde(Stepstone Corporation) tasarlanmıştır. Objective C,

- Zengin bir nesne (object) kütüphanesi
- Gerekli geliştirme araçları (tools) ve
- Nesneye Yönelik Programlama özellikleri ve destek kütüphanelerine sahiptir (3).

Uygulamada kullanılan OpenCV, Open Source Computer Vision Library' nin kısaltılmış hali olup C++ ve C dilleri ile oluşturulmuş ,Mac OS X, Linux ve Windows gibi değişik platformlarda çalışabilen açık kaynak kodlu görüntü işleme (image processing) ve bilgisayar ile görme (computer vision) kütüphanesidir. OpenCV kütüphanesi içerisinde bulunan 500'den fazla fonksiyonla tıbbi görüntüleme,ürün denetimi, güvenlik, kamera kalibrasyonu kullanıcı ara yüzü gibi alanlarda akademik çalışmalar yapılabilir, uygulamalar geliştirebilir. Görüntü işlemeye hayatımızda endüstriyel ürünlerden güvenlik ürünlerine, mobese kameralarından statlardaki kameralara kadar nerdeyse her alanda ihtiyaç duyulmaya başlanmıştır. OpenCV sayesinde bütün bu bahsedilenleri yapabilmemizi sağlamak üzere tasarlanmış bir kütüphanedir (4).

Gelişen teknolojiler ve kullanıcıların bu teknolojilere hızlı bir şekilde adapte olmaları yeni uygulamalar geliştirmeyi de kaçınılmaz kılmıştır. Bu şartlar ve olanaklar göz önüne alınarak, IOS, Objective-C, XCode platformu, OpenCV kütüphaneleri ve web servis teknolojileri kullanılarak trafik polislerinin trafikte kesilen cezaları anlık olarak kullanıcılara bildirmesini hedefleyen mobil bir uygulama geliştirilmesi hedeflenmektedir. Türk sivil araçlardan alınan plaka resimlerini OpenCV görüntü işleme kütüphanesi kullanarak resimdeki anlamlı bilgiler çıkarıp işleyen ve sonucunda elde edilen karakterleri plaka bilgisine dönüştürdükten sonra, gerekli ceza bilgileri düzenlenerek araç sahiplerine web servis aracılığıyla bilgilendiren özgün bir trafik ceza bildirim uygulaması geliştirilecektir. Daha önceden kullanıcılara anlık ceza bildirimi yapan bir teknik veya tekniklerin birleşmesinden oluşan bir çözüm mevcut değildir. Yeni geliştirilecek bu uygulama ile trafik bilgi sistemi alanında özgün çözüm üretilmesi hedeflenmektedir.

Bölüm 2'de geçmiş çalışmalar, bölüm 3 sistemin işleyişi, bölüm 4'te yapılan projeden, bölüm 5 te sonuçtan , bölüm 6 da ise ileriye yönelik çalışmalar hakkında bilgi verilecektir.

2. Geçmiş Çalışmalar

Çeşitli platformlarda plaka tanıma ile çalışmalar yapılmasına karşın iOS platformunda yeteri

kadar çalışma bulunmamaktadır. Aynı zamanda çalışmada web servis ile entegrasyon edilmesi ile ortaya özgün bir çalışma olması hedeflenmektedir. Bu bilgiler ışığında literatüre özgün bir çalışma kazandırılmak istenmektedir ve pek fazla bu tür uygulama çalışmaları henüz yapılmamıştır.

Ali Gökay Günaydın 'ın çalışmasında, Java platformu üzerinde plaka tanıma sistemi tasarlanmış ve geliştirilmiştir. Plaka yerini belirleme algoritması, kısıtların dikey köşe özellikleri üzerinden metin olmayan alanların filtrelenmesi üzerine kurulmuştur. Karakter ayrıştırma işlemi için yatay ve dikey izdüşümleri, Optik Karakter Tanıma modülü (OCR) için ise Çok Katmanlı Algılayıcılar (MLP) kullanılmıştır (5).

Parisi ve arkadaşları İtalyan tarzı araç plakaları tanıması için çalışma yapmışlardır. Plaka görüntüleri genellikle turnike kapılarındaki kameralardan alınmıştır. Plaka karakterlerini tanımak için güçlü ve hızlı 1-D DFT şemaları kullanılmıştır. Karakterlerin sınıflandırılması son zamanlarda gelişen BRLS öğrenme algoritması tarafından eğitilen çok katmanlı sinir ağı aracılığıyla oluşturulmuştur (6).

Ali Bakkaloğlu'nun çalışmasında Plaka yerinin tespiti için resim üzerindeki piksel değerlerinin değişim oranları incelenmiştir. Bu inceleme sonucunda plaka bölgesi tespit edilmiştir. Karakter ayrıştırma işlemi için resim üzerindeki sütun toplam vektör değerleri kullanılmıştır. Ayrıştırılan karakterlerin tanınması için şablon eşleştirme yöntemi kullanılmıştır. Bu 3 ana işlem sırasında, resimler işlenirken bilinen resim filtreleme fonksiyonları kullanılmamıştır. Bilinen resim filtreleri yerine doğrudan işlemin amacına yönelik algoritmalar yazılmıştır. Bu yöntemle, işlemin en hızlı şekilde tamamlanması sağlanmıştır (7).

Mohd ve arkadaşları çalışmalarında, Parçacık Sürü Optimizasyonu ile RBF Sinir Ağı parametrelerinin kombinasyonunu kullanarak geliştirilebilir bir araç plaka tanıma sistemi geliştirmişlerdir. RBF Sinir Ağı parametrelerini optimize etmek için PSO kullanmışlardır. Bu iki metodun kombinasyonu sayesinde plaka tanıma süreleri azaltarak daha hızlı tanıma gerçekleştirmişler (8).

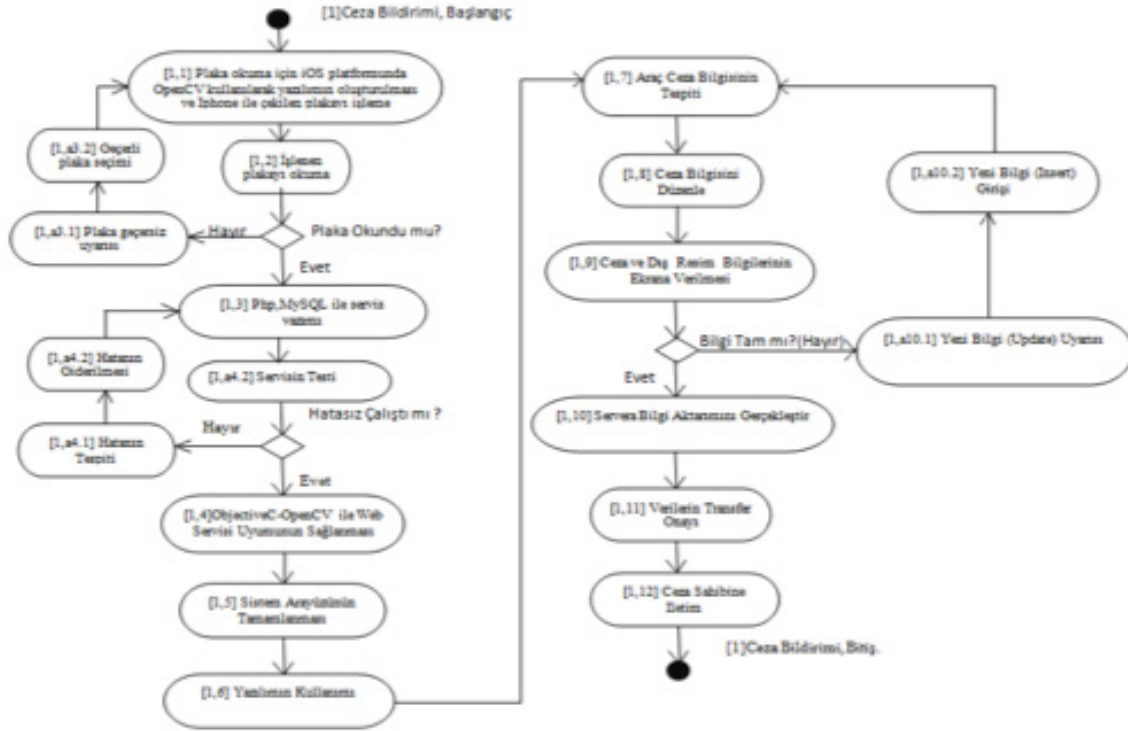
Atınç Yılmaz'ın çalışmasında görüntü işlemeyi dijital olarak alınan görüntülerin işlenerek yapılarının ve özelliklerinin değiştirilmesini, geliştirilmesini aynı zamanda bu görüntüler vasıtasıyla analizlerin yapılmasını elverişli hale getiren teknoloji olduğunu ifade etmektedir. Görüntü işleme ile bir

görüntülere Açık Kaynak Kodlu Görme Kütüphanesi (OpenCV) kullanarak Gri ölçekli adaptasyonu, Histogram filtre, RGB denkleştirme ve Sınır Algılama görüntü işleme algoritmalarını uygulamışlardır (10).

Deepthi ve Sankaraiah ise çalışmalarında gelişen teknoloji ile mobil cihazlarda görüntü işlemenin önemini vurgulamışlar ve geliştirdikleri entegre çalışma ile mobil cihazlarda elde edilen görüntülerin bilgi kaybı olmadan depolanmasına katkı sağlamıştır. Çalışmalarında kullandıkları

3. Trafik Ceza Uygulamasının Yöntemi

Uygulama IOS platformunda Objective - C programlama dili ve OpenCV (11) kütüphanesi kullanılarak hazırlanmıştır. Şekil 1 de sistemin akış diyagramı gösterilmiştir.



Şekil 1. Sistemin Akış Diyagramı

3.1 Plaka Resminin Çekilmesi

Uygulama içerisinde Plaka Tanı kısmı açılarak plakası çekilmek istenen aracın plakası kamera-ya yaklaştırılarak çekilmesi işleminden oluşmaktadır. Plakadaki karakterlerin tanımlanması için plakanın kamera görüntüsünün kenarlarında yaklaşık olarak %10 - %15 oranında boşluk bırakılarak çekilmesi gerekmektedir. Şekil 2 çekilmiş plaka resmini göstermektedir.



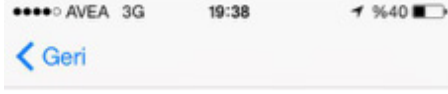
Şekil 2. Plaka Resmi

3.2 Resmin İşlenmesi

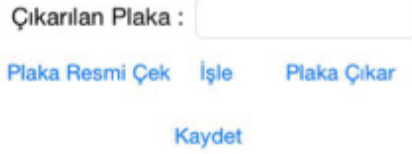
Resmin işleme kısmında, ilk olarak önceden hazırlanmış olan test verileri uygulamaya tanıtılır. Karakterleri tanımak için açık kaynak kodlu Tesseract motoru (12) kullanılmaktadır.

Çekilen RGB formatındaki resim 0-255 arasında renkler alacak şekilde gri formata dönüştürülür. daha sonra kenarların bulunması gerçekleştirilir. Kenar bulma işlemi için Sobel algoritması kullanılmıştır. Sobel Algoritması kenar bulma algoritmaları arasında en çok kullanılan ve bilinen algoritmadır. Kenarları bulma işlemi sayesinde resim içerisindeki gerekli işlemler rahatlıkla yapılabilmektedir. Bu algorithmada konvulsiyon çekirdekleri kullanılmaktadır. bunlardan birisi yatay diğeri ise dikey kenarları bulmaya yaramaktadır. Kenarları bulunan resimde eşik değeri uygulanarak değerler tam olarak beyaza ve si-

yaha çekilirler. Daha sonra resme morfolojik görüntü işleme yöntemleri uygulanarak plakada az görünen ya da değerlerde silinmiş olan veriler tamamlanır ve resimde var olan gürültüler yok edilmiş olur. Böylelikle karakter tanımlama işlemi kolaylaştırılmış olur. Çekilen resimde plakanın orta kısma göre ayarlanması karakterlerin eksiksiz okunmasında büyük önem taşımaktadır. Şekil 3 yukarda bahsedilen işlemlerden sonra elde edilen görüntüyü göstermektedir.



20 R 8880



Şekil 3. İşlenmiş Resim

Plakanın orta kısımda olmasının yanında, plakanın çekilen resimde yaklaşık olarak ekranın kenarlarından %10 luk boşluk olacak şekilde ayarlanması gerekmektedir, aksi takdirde okuma işlemi tam olarak gerçekleşemeyecektir.

3.3 Diğer İşlemler

Plaka resmi işlendikten sonra Şekil 2’de gösterilen “Plakayı çıkart” butonuna bastığımızda çıkarılan plaka “Çıkarılan Plaka” kısmına gelmiş olur. Eğer plaka okumada bir terslik ya da yanlışlık var ise düzeltmek istersek bir daha plaka resmi çekebilir ya da hatalı karakter klavye yardımıyla düzeltebilir. Plaka tanımlama işleminden sonra bir diğer menüden aracın istenilen kısımlardan resimleri çekilir.

Resimler çekildikten sonra araç ile ilgili işlemlerin ayrıntılarının girilmesi için ceza türü, ceza miktarı

ve ayrıntılı polis memuru notları kısımları doldurulur.



Şekil 4. Ayrıntılı Ceza Bilgileri

Bütün bilgiler doldurduktan sonra veriler sisteme gönderilmeden önce önizleme ekranı açılarak, uygulamayı kullanan kişinin girmiş olduğu bütün veriler aynı ekranda görünür ve onay istenir, kullanıcı onay verdiği takdirde, yazılan web servisi aracılığı ile bütün veriler internet ortamındaki veritabanına kaydedilir.



Şekil 5. Ceza Gönderimi Onay Ekranı

5. Sonuçlar

25 adet plakadan okunan değerler tablodan alınan değerler Tablo 1 'de yer almaktadır.

Plaka Doğrulukları	Doğru	Yanlış	Başarı Oranı
Gece	11	9	%55
Gündüz	16	4	%80

Tablo 1. Başarı Oranları

Tablodaki veriler incelendiğinde mobil telefonlardaki kameraların gece çekimlerinin düşük kalitede gerçekleşmesi sonuçlara doğrudan etki göstermiştir.

6. Sonraki Çalışmalar

Yapılan uygulama kullanıcılar için plakayı kamerada istenilen oranda tutulmasının ne kadar zor olduğunu göstermiştir. Ayrıca işlemlerin hızlı ve daha rahat yapılabilmesi için fotoğraf çekmekten daha çok video çekiminden plakanın tanımlanabilmesinin daha etkili olacağını göstermiştir. Uygulama Ceza bilgilerini sisteme göndermekte araç sahibiyle herhangi bir şekilde iletişim kurmamaktadır.

Bir sonraki çalışmalarda hem plakanın kamerada oranlanması işlemi esnekleştirilecek, hem de video üzerinden plaka tanımlama işlemleri yapılmaya çalışılacaktır. Ayrıca Araç sahiplerini hızlı bir şekilde bilgilendirebilmek için operatör sistemleri kullanılarak araç sahibine kısa mesaj gönderiminin yapılması planlanmaktadır.

7. Kaynaklar

[1] Köroğlu, O. En yaygın iletişim ortamı: Mobil iletişim ortamında içerik ve yayıncılık. Civilacademy, 55(2013).

[2] Haslam, K., "Macworld Expo Optimised OS X sits on versatile flash",. <http://tr.wikipedia.org/wiki/%C4%B0OS> (2007), Erişim Tarihi: 10.12.2014

[3] Mithat, U., "Objective C ve iPhone'da Uygulama Geliştirme", Nirvana yayınevi, 443(2012).

[4] Tınaz, S., Karayakalı, S., "Trafik İşaret ve Levhalarını Tanıma", Tez Ara Rapor(2011).

[5] Günaydın, A., "Kısıt Tabanlı Gerçek Zamanlı Plaka Tanıma Sistemi", Orta Doğu Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans

Tezi, 48syf., Ankara(2007).

[6] Parisi, R., Di Claudio, E.D., Lucarelli, G., Orlandi, G., "Car plate recognition by neural networks and image processing," Circuits and Systems, 1998. ISCAS '98. Proceedings of the 1998 IEEE International Symposium on , pp. 195-198, (1998).

[7] Bakkaloğlu, A., "Araç Plaka Tanıma Sistemi", Selçuk Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, 87syf., Konya(2011)

[8] Maruzuki, M.I.F., Ishak, S.N., Setumin, S. Control System, Computing and Engineering (ICCSCE), 2012 IEEE International Conference on , pp. 511-514(2012)

[9] Yılmaz, A., 2007. Kamera Kullanılarak Görüntü İşleme Yoluyla Gerçek Zamanlı Güvenlik Uygulaması, Haliç Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, 103p, İstanbul(2007)

[10] Deepthi, R.S., Sankaraiah, S., "Implementation of mobile platform using Qt and OpenCV for image processing applications", Open Systems (ICOS), 2011 IEEE Conference on, pp. 284-289(2011)

[11] iOS OpenCV Kütüphanesi, http://docs.opencv.org/doc/tutorials/ios/table_of_content_ios/table_of_content_ios.html , Erişim Tarihi: 15.12.2014

[12] iOS Tesseract Karakter Tanıma Kütüphanesi, <https://github.com/gali8/Tesseract-OCR-iOS>, Erişim Tarihi: 15.12.2014