

PlatePass

Caner Ay Celep

201307036

cnr24clp@gmail.com

Abstract– Plate recognition (LPNR) for desktops app. Project was developed with asp.net core. Users can use camera or stock photo image (.jpg or .png) for plate recognition. Sobel and Gray filter apply for cropped images and dedection plate number with OCR.

Keywords – LPNR, asp.net , sobel, gray, OCR.

Özet– Masaüstü uygulamaları için plaka tanıma (LPNR). Proje asp.net core ile geliştirildi. Kullanıcılar plaka tanıma için kamera veya hazır fotoğraf görüntüsünü (.jpg veya .png) kullanabilir. OCR ile kırılmış görüntüler ve algılama plakası numarası için Sobel ve Gray filtresi uygulanır.

Anahtar Kelimeler – LPNR, asp.net , sobel, gray, OCR.

Problem Tanımı

Proje kapsamında avm veya site gibi yerlerde plaka tanıma sistemlerinin geçişi için bir uygulama geliştirilmek istenmiştir. Bu doğrultuda araç resminden veya kameradan alınan bir görüntünün plaka kısımlarının kesilmesi , sobel ve gray filtrelerinin uygulanarak ilgili plaka bilgisinin veritabanı servisinde kayıtlı olup olmadığının alınması buna göre geri bir bildirim olarak plakanın tanımlı olup olmadığını sonucunda kullanıcıyı bilgilendiren bir otomasyon yazılımı yapılmıştır.

I. Yapılan Araştırmalar

Görüntü işleme için kullanılacak olan kütüphanelerin belirlenmesi , resimler üzerinde yapılacak olan filtreler ile alakalı kapsamlı bir araştırmaların yapılması. Veritabanı kayıt işlemleri için EF-Core Code-First yaklaşımının belirlenmesi. Kameradan alınacak görüntünün işlenip veritabanı sistemlerinde kaydı olup olmadığının bakılması.

II. Kullanılan Teknoloji

- Asp.Net Core (Back-End)
- Ef Core
- OpenCv
- EmguCv
- TesseractOcr

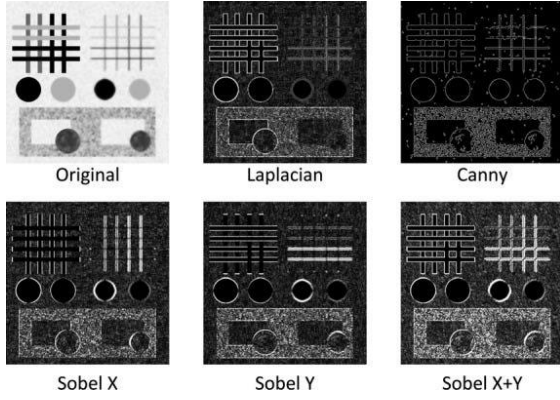
III. Araç Kaydetme

Araç kayıt işlemleri için yeni bir form panel eklendi. Form panel ile ilgili kullanıcı menu-strip aracından araç kayıt seçeneğine girdiği zaman araç kaydetmesi için gereken bilgilerin olduğu sayfaya ulaşmaktadır. İlgili resim dosyasını seçtikten sonra dosyanın önizlemeli bir seçeneğini ve kullanıcının vermiş olduğu resim dosyasından kırılan plaka bilgilerinin yer aldığı önizlemeli resim görüntülenmektedir. Kullanıcı bu sayede bu sayfadan seçmiş olduğu aracın plaka bilgilerinin direk textbox nesnesinde görebiliyor. Kullanıcılara ait ad, soyad gibi bilgilerin de doldurulduktan sonra veritabanına plaka kayıt işlemleri yapılıyor. İlgili kayıt sayfası “Şekil-1” ‘de gösterilmiştir.



Şekil-1

Ressim üzerinde bir görüntü alındığı zaman ilk olarak plaka kısmının fotoğraftan kırılması ve alınması gerekiyor. Bu doğrultuda ilk olarak Mat sınıfını kullanarak resmin bir görüntüsünü almış oldum. Renkli çıktı üzerinde işlem yapma maliyeti siyah beyaz çıktı üzerinde işlem yapmaktan daha maliyetli olduğu için ilgili resme sobel , gray , gaussian blur efektleri uyguladım. Sobel efekti sayesinde plaka çerçevesi ve içindeki yazılar daha detaylı ve okunabilir gözükmiş olacak bu sayede OCR aracımız ile ilgili yazı bilgisini okuyabilecek. İlgili Sobel efekti ile alakalı görüntü “Şekil-2” ‘de verilmektedir



Şekil-2

Sobel filtresi uygulandıktan sonra ise resimdeki pürüzlükleri ortadan kaldırmak ve plaka okuma işlemleri için OCR aracının resim üzerindeki yazıları daha kolay okuyabilmesi için Gauss filtresi uygulandı. Gauss filtresi sayesinde resim üzerindeki pürüzlükler ortadan kalkmış oldu.İlgili şekil “Şekil-3” ‘de verilmiştir.



Şekil-3

Sobel filterisini gerçekleştirdikten sonra ise resim üzerindeki plaka adreslerinin okunması için yapılacak işlem olan Tesseract işlemlerinin yapılması için ilgili olan Tesseract kütüphanesini projenin içine dahil etmiş oldum. Tesseract birden çok dil desteği olduğu için dil desteğini Türkçe olarak ayarlayıp Türkiye’de geçerli olan plaka biçimini Tesseract OCR’a bir kural olarak eklemiş oldum. İlgili kod parçası “Şekil-4” ‘de verilmiştir.

```

Tesseract
void GetTextFromImg()
{
    using (MemoryStream ms = new MemoryStream())
    {
        photoPlate.Image.Save(ms, System.Drawing.Imaging.ImageFormat.Jpeg);

        using (var engine = new TesseractEngine(System.Windows.Forms.Application.StartupPath + @"(tessdata-main", "tur", EngineMode.Default))
        {
            engine.SetVariable("tesseract_char_whitelist", "0BCDEFHJKLMNPRSTUVWYZ-1234567890");
            using (var img = Pix.LoadFromMemory(ms.ToArray()))
            {
                using (var page = engine.Process(img))
                {
                    string text = page.GetText();
                    thisPlaka.Text = text;
                    if (thisPlaka.Text == null)
                    {
                        thisPlaka.Text = "Resin çok bulanık !";
                    }
                }
            }
        }
    }
}

```

Şekil-4

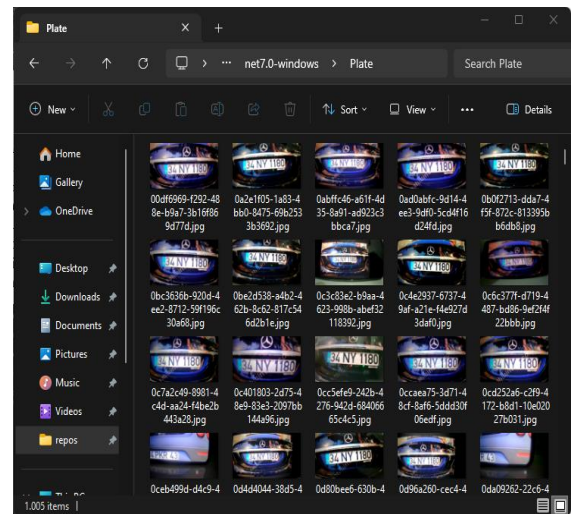
IV. Araç Sorgulama

Araç sorgulama ekranında ise veritabanına kaydı yapılan araçların sorgulama işlemleri ve buna göre aracın kayıtlı bir araç olup olmadığı bilgisini veren sayfa tasarımı yapılmış olup. Kullanıcının istemesi dahilinde kamera ile araç sorgusunu yapabileceği bir sistem geliştirilmiştir. Kullanıcı isterse dosya üzerinden bir sorgu gerçekleştirebilir veya kamera üzerinden bir sorgu gerçekleştirebilmektedir. İlgili sayfa “Şekil-5” ‘de gösterilmiştir.



Şekil-5

Kullanıcı bu sayfadan araçların sistemde kayıtlı olup olmadığı durumunu sorgulayarak görebilmekte. Sistemde yapılan en son 10 geçiş kaydını da bu sayfa üzerinde görüntüleyebilmekte. Bu sayede kullanıcı ilgili kullanıcıların geçişleri hakkında bilgi olabilmekte , tanımsız veya tanımlı araç durumuna göre göz önünde bulundurabilmekte. Kamera açarak okutulan araçların görüntüleri ise klasör içerisinde tutulmakta bu sayede sorgulanan araçların kaydı yapılabilir. İlgili kayıt “Şekil-6” ‘da verilmiştir.



Şekil-6

V. Araç Düzenleme

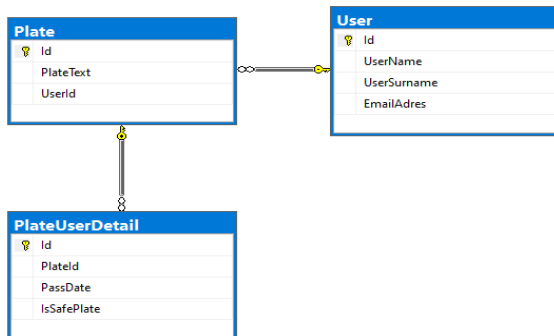
Araç düzenleme sayfasında ise kullanıcılar yanlış bir kayıt işlemi gerçekleştirmesi durumunda gerekli düzenleme işlemlerinin yapılması veya silinmesi istenilen bir plaka kaydının silinmesi durumunu gerçekleştirebilirler. Bu sayede geçersiz kayıtların önüne geçilmiş olunmakla beraber güncellenilmek istenilen plaka adreslerinin düzenlenmesi gerçekleşmiş olmaktadır. İlgili sayfa “Şekil-7” ‘de gösterilmektedir.

The screenshot shows a web application titled "Plate Register". On the left, there is a sidebar with a logo and user information: Plaka: 34N01180, Ad: Cemer Ay, Soyad: Celal, Email: cm24dp@gmail.com. Below this are buttons for "Ara", "Görüle", and "Kaydı Sil". The main area displays a table with columns: ID, Pass Date, Plate ID, and Plate. The table contains 12 rows of data. At the top right of the table, there is a button labeled "Tüm Geçişler".

Şekil-7

İlgili veritabanı servislerindeki ilişkiler aracılığı ile ilgili plaka adresleri silindiği zaman veya kullanıcı bilgilerinde bir değişiklik yapıldığı zaman ilişkisel olarak verilerin durumları da buna göre değişmekte. Kullanıcı ve plaka arasında bire çok ilişki kullanılmıştır. Eğer kullanıcı silinirse kullanıcıya ait olan bütün plaka bilgileri de silinir. Bir kullanıcı birden çok plaka bilgisine sahip olabilmekte. Yani bir kullanıcı üzerinden birçok plaka kaydı yapılabilmektedir. İlgili ilişkisel diyagrama ait görüntüler “Şekil-8” ‘de verilmektedir.

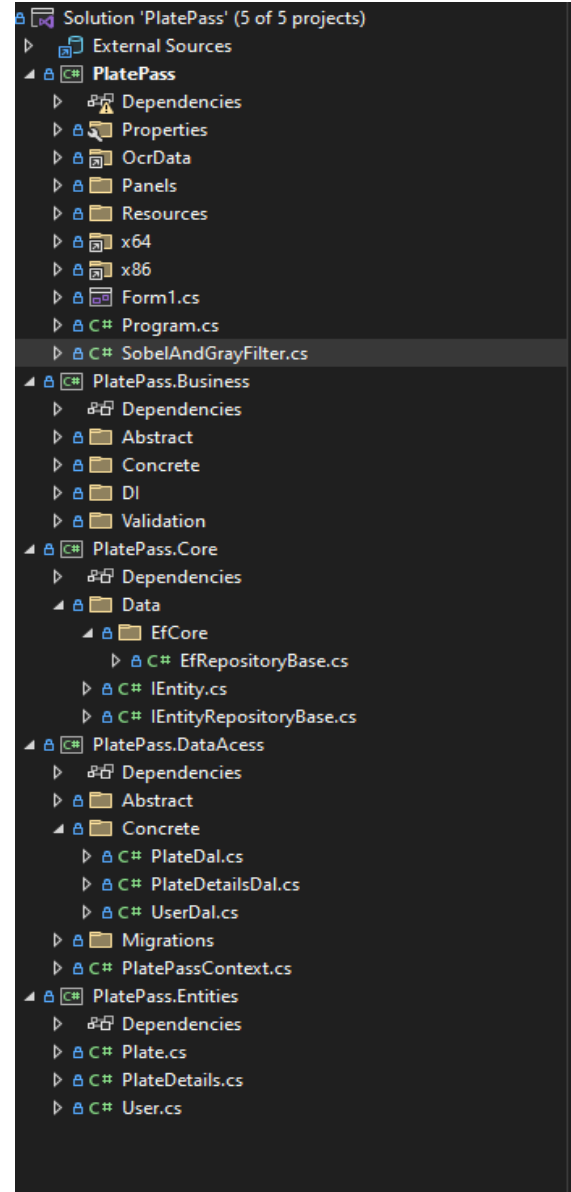
EFMigrationsHistory	
MigrationId	
ProductVersion	



Şekil-8

VI. Proje Katmanları

Proje içerisinde bulunan DataAccess,Core,Entity katmanlarının yapısı “Şekil-9” ‘da verilmiştir.



Şekil-9

VII. Kaynakça

- https://www.emgu.com/wiki/index.php/License_Plate_Recognition_in_CSharp
- <https://github.com/tesseract-ocr/tesseract>
- <https://ironsoftware.com/csharp/ocr/blog/using-ironocr/license-plate-ocr-csharp-tutorial/>