# Optik Form Okuyucu

### Github Adresi

<https://github.com/Python-Egitici-Egitimi-OOP-Grup-8/Optik-Form-Okuyucu> ( OMR Kısmı )

<https://github.com/Python-Egitici-Egitimi-OOP-Grup-8/web_app_project> ( Web Değerlendirme Kısmı )

### Grubu oluşturan öğretmenlerin listesi

Celal AKSU, Nihat Bal, Mert DOĞAN, İbrahim KILIÇ, Yakup KUCUKKARACA, Şahin MANSUROĞLU, Tunahan ONCEL, Mustafa ÖZER

# Optik Form Okuyucu Analiz Raporu

### Kısa Özet

Optik Form Okuyucu projesine, sıradan OMR yazılımlarından farklı olarak, cep telefonundan dahi standardın dışında yapılmış olan taramaları bile okuyabilen, her türlü optik form okuyabilme alt yapısında geliştirilmiş ve web ortamındaki değerlendirme bölümü ile de okul öğretmenlerimizin kuracakları basit bir web sunucuda bile en üst düzeyde stabilite ve hızlı işleme standartlarında geliştirilmiş bir Optik Form Okuyucu yazılımın oluşturulması planlaması ile başlanmış ve Python kütüphanelerinin geniş imkanları ve mevcut geliştirilmişlik başarımına ulaşmış projeler baz alınıp üzerlerine kullanıma sunulmuştur.

### Problem Tanımı

Öğrencilerin kendi başlarına uyguladıkları ve sonunda optik formlarını, en basit düzeyli cep telefonlarından fotoğrafını çektikleri sınav uygulamalarına ait optik formlarının değerlendirmelerini yapabiliyor muyuz?

Optik formların günümüz en sık tercih edilen okuyucu yazılımlarda okunabilmesi için bu yazılımların yanında çok net tarama işlemleri de yapabilen ve pahalı tarayıcı donanımları da satın alınması gerekmektedir. Oysa ki, neredeyse tüm öğrencilerde bulunan cep telefonlarını bir optik form tarayıcı olarak kullanabilirsek, hem bu donanım maliyetinden kurtulmuş, hem de iş yükünü dağıtarak iş gücünden, zamandan ve donanımın kaplayacağı mekandan da tasarruf ederiz.

### Analiz Süreci

Yazılımın üretilmesi sürecinde, Python’un güçlü mimaride geliştirilmiş olan kütüphanelerinin görsel ve matematiksel işlem yeteneklerinden faydalanılmış, Github platformunun birlikte çalışma imkanı ve Cisco’nun Netacad eğitimlerinin sağladığı bilgi kaynakları kullanılmıştır. Geliştirme ortamı olarak Microsoft’un açık kaynaklı geliştirme platformu olan Visual Studio Code’dan yararlanılmıştır.

#### İhtiyaç Analizi

Optik Form Okuyucu pek çok ticari yazılım bulunmaktadır. Bu yazılımlar pahalı tarayıcı cihazlara ve optik okuma süreçlerinin tamamlanması için de işgücüne ihtiyaç duymaktadır. Ayrıca bu yazılımlar sadece belirli form yapılarını okuyabilmekte, standardın dışına çıkmış optik taramalarda başarımları ciddi oranda düşmektedir. Ayrıca yeni optik form tanımlamalarında yine maliyet doğurmaktadır.

Optik Form Okuyucu projemiz sayesinde, öğretmenlerimiz okullarında veya ihtiyaç duydukları diğer her türlü ortamda, neredeyse her türlü formu hiçbir ticari desteğe ihtiyaç duymadan, sadece birkaç tarama ve parametre girişi ile optik form ile uyguladıkları sınavları okuma imkanı bulacaklardır. Ayrıca programın güçlü resim işleme mimarisi sayesinde, öğrencilerce işaretlenen optik formların taraması için güçlü tarayıcı donanımlara ihtiyaç duyulmayacak ve öğretmenin en ekonomik cep telefonundan bile çektiği fotoğraflardan bile değerlendirmesi sağlanacaktır.

Proje sadece öğretmenlerin kendi uyguladıkları sınavların değerlendirmesi ile sınırlı değildir. Web arayüzü sayesinde öğrencilerin kendi kendilerine uygulayacakları sınavları da yine kendi cep telefonlarından çekip gönderecekleri web ortamında veya geliştirilebilecek olan mobil yazılımlarda hemen değerlendirmeye gönderilebilecektir.

#### İçerik Analizi

Projenin geliştirilmesinde öncelikle OMR konusunda python programlama dili ile geliştirilmiş mevcut açık kaynak kodlu projelerin tümünün Github üzerinden incelenmesi ve ulaştığı düzey proje hedefimize en uygun olan yazılımın mevcut durumu baz alınarak nihai hedefimize yol almaya başlama kararı verilmiştir.

Yazılımda python programlama platformu ve numpy, pandas, matplotlib, imutils kütüphanelerinden faydalanılmıştır. Proje geliştirilirken, isteyen kullanıcılarının sadece masaüstü cihazlarının bile yeterli olabileceği bir mimari düşünülmüş ancak daha üst düzeyli değerlendirme ve sınav sonuç işlemleri için ise web arayüzü geliştirilmiştir. OMR ana projesinin web arayüzü ile haberleşmesi için ise masaüstü GUI ekranı oluşturulmuş, web arayüzüne de gelen json verilerini değerlendirme yeteneği kazandırılmıştır.

#### Durum Ortam Analizi

Yazılımın geliştirilmesi sürecine başlanırken kullanılan python kütüphaneleri olan numpy, pandas, matplotlib, imutils kütüphanelerinin birlikteliği sağlanmıştır. Test süreçlerinde kullanılmak üzere sıfırdan bir deneme optik formu dizayn edilmiş ve yine tarafımızdan çoğaltılıp doldurulan optik formlarla denemeleri yapılmıştır.

#### Kullanıcı Analizi

Yazılım tamamen GNU lisansı ile dağıtılmıştır. Böylece kullanıcılar kodlarında kendi ihtiyaçlarına göre her türlü özelleştirmeleri dilediklerince yapabilecekler ve gelişimine katkı sağlayabileceklerdir.

Yazılımın kullanılabilmesi için sunulmuş olan Masaüstü GUI ve web arayüzü sayesinde, özünde herhangi bir gelişmiş teknik bilgiye sahip olmaya gerek yoktur. Ancak projenin kaynak kodundan faydalanıp kullanabilmek için temel python bilgisine sahip olmak, python kütüphanelerini tanıyor olmak, web arayüzü üzerinde proje geliştirmesinde yer almak için de temel sqlite veritabanı, DJango web framework, JSON işleme ve temel HTML bilgisine sahip olmalıdır. Ayrıca web ortamında uygulama geliştirileceği ve veritabanı bilgi alışverişi olacağı, bu alışveriş de istemcilerin bilgi girişleriyle gerçekleşeceği için SQL Injection, XSS gibi önemli güvenlik tehditleri ve önleme yollarına da hâkim olması, public erişime açık projeler için hayati öneme sahiptir olacaktır.

# Optik Form Okuyucu Tasarım Raporu

### Kısa Özet

Proje tasarımı sürecinde öncelikle işin OMR tarafına odaklanılmış ve esnek yapıya sahip çekirdek geliştirilmesi sağlanmıştır. Ardından masaüstü GUI uygulaması geliştirilmiştir. Diğer bir koldan da sınav değerlendirmelerini yapacak olan ve OMR yazılımının masaüstü arayüzünden gönderilecek olan json verisinin işlenmesini ve değerlendirmesini sağlayacak web uygulaması geliştirilmiştir.

Projenin Masaüstü Uygulaması:

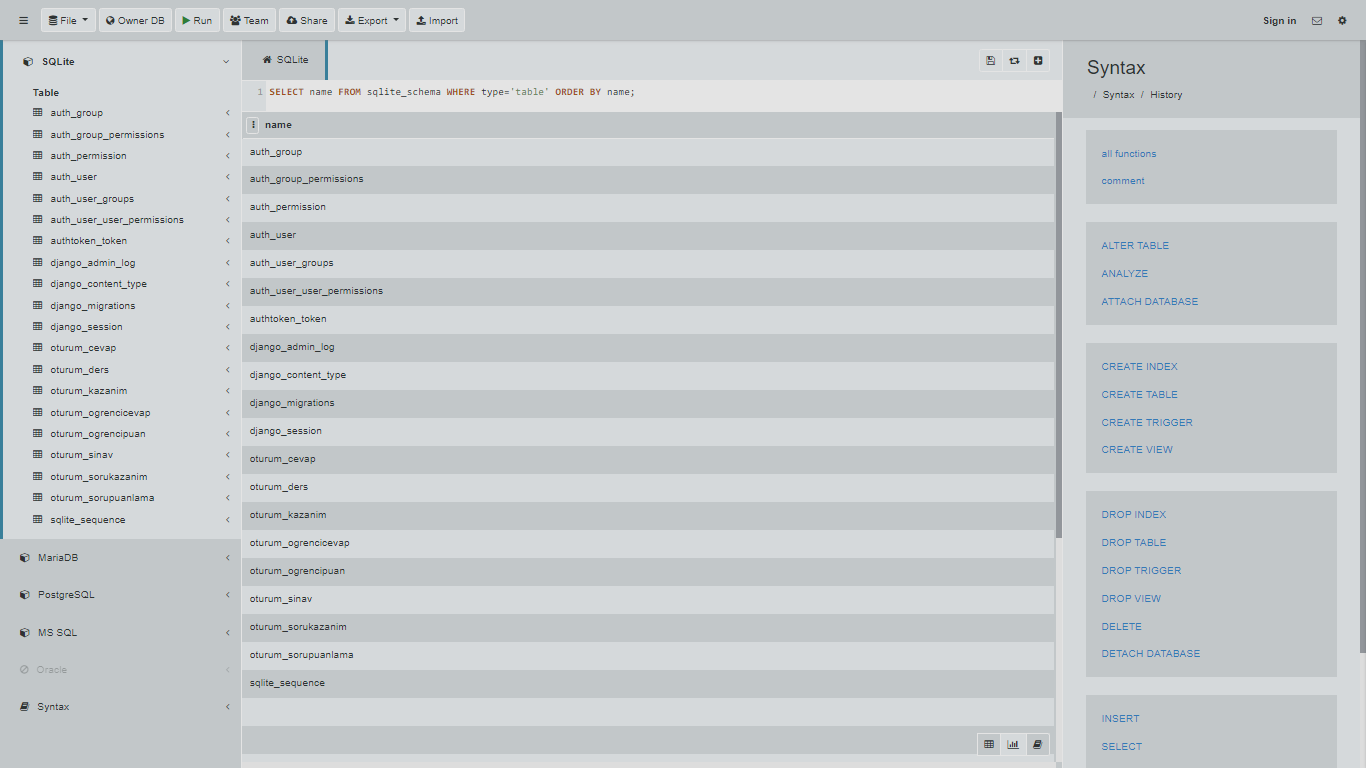
Projenin masaüstü uygulaması iki formdan oluşmaktadır.

Birinci ve program başlatıldığında ilk açılan form kullanıcının giriş yapması için kullanılan formdur. Burada kullanıcı bilgileri REST API’ye gönderilerek uygulama başlatılmaktadır. Sucudan token bilgisi alınmakta ve bu token bilgisi sonrasında sunucuya veri göndermek için kullanılmaktadır.

İkinci form kullanıcının optik formaları okuması ve sunucuya gönderilmesi için kullanılan formdur. Burada optik formların bulunduğu klasör seçilerek işlem başlamaktadır. Arka planda optik formlar okunarak sonuçlar csv dosyasına kayıt edilmektedir. Sonraki adımda csv dosyasındaki veriler, kullanıcının aldığı token bilgisi ile json formatında sunucuya gönderilmektedir.

## Veri Tasarımı

Web değerlendirme uygulamasında alınan, değerlendirilen ve üretilen veriler sqlite veritabanında saklanmıştır. Veritabanı tablo yapısı aşağıdaki gibidir:



Masaüstü Bölümünde kullanılan csv veri yapısı örneği aşağıdaki gibidir.

"file\_id","input\_path","output\_path","score","SINAVKODU","TCKIMLIKNO","q1","q2","q3","q4","q5","q6","q7","q8","q9","q10","q11","q12","q13","q14","q15","q16","q17","q18","q19","q20","q21","q22","q23","q24","q25","q26","q27","q28","q29","q30","q31","q32","q33","q34","q35","q36","q37","q38","q39","q40","q41","q42","q43","q44","q45","q46","q47","q48","q49","q50"

"20200814\_133604.jpg","inputs/Sinif4/20200814\_133604.jpg","outputs\/CheckedOMRs/Sinif4\_20200814\_133604.jpg","0","268407","26419134857","","","","","","","","","","","","","","","","","","","","","","","","D","A","","","","","","","","","","","","","","","","","","","","","","","","B",""

"20200814\_133617.jpg","inputs/Sinif4/20200814\_133617.jpg","outputs\/CheckedOMRs/Sinif4\_20200814\_133617.jpg","0","268407","37194867825","A","B","C","D","E","E","D","C","B","A","A","A","A","A","A","B","B","B","B","B","C","C","C","C","C","D","D","D","D","D","E","E","E","E","E","A","E","B","D","C","C","B","D","A","E","A","C","E","C","A"

Sunucuya gönderilen json verisi aşağıdaki gibidir.

{'ogrenci': 15, 'sinav': 46, 'C1': 1, 'C2': 2, 'C3': 3, 'C4': 4, 'C5': 5, 'C6': 5, 'C7': 4, 'C8': 3, 'C9': 2, 'C10': 1, 'C11': 1, 'C12': 1, 'C13': 1, 'C14': 1, 'C15': 1, 'C16': 2, 'C17': 2, 'C18': 2, 'C19': 2, 'C20': 2, 'C21': 3, 'C22': 3, 'C23': 3, 'C24': 3, 'C25': 3, 'C26': 4, 'C27': 4, 'C28': 4, 'C29': 4, 'C30': 4, 'C31': 5, 'C32': 5, 'C33': 5, 'C34': 5, 'C35': 5, 'C36': 1, 'C37': 5, 'C38': 2, 'C39': 4, 'C40': 3, 'C41': 3, 'C42': 2, 'C43': 4, 'C44': 1, 'C45': 5, 'C46': 1, 'C47': 3, 'C48': 5, 'C49': 3, 'C50': 1}

## Ara yüz Tasarımı

Kullanıcıların optik form değerlendirmelerini zorlanmadan halledebilmeleri için masaüstü GUI arayüzü hazırlanmıştır. Optik form okunmuş datanın işlenmesi için de web arayüzü oluşturulmuştur.

Masaüstü arayüz tasarımı

1. Kullanıcı giriş arayüzü

ekran görüntüsü içeren bir resim

Açıklama otomatik olarak oluşturuldu

1. Form okuma ve sunucuya gönderme işlemleri arayüzü

ekran görüntüsü içeren bir resim

Açıklama otomatik olarak oluşturuldu

## Kod Tasarımı

Kodlama yapısı oluşturulurken kullanıcıların olabildiğince az python programlama bilgisi ile modifikasyonları hedeflenmiştir. OMR çekirdeği oluşturulurken kullanıcıların kendi ihtiyaç duyacakları her türlü optik form tasarımına göre özelleştirme yapabilecekleri biçimde inşa edilmiştir. Web arayüzü tasarlanırken ise django kütüphaneleri üzerine yapılan kod inşası sürecinde olabildiğince OMR kod yapısından yalıtılmış ve farklı OMR okuma yazılımlarından da gelebilecek olan datanın da işlenebilirliği göz önüne alınmıştır.

Masaüstü kod yapısı

main.py : Fork edilen repositorynin çalıştırılabilir dosyasıdır.

sunucudosya/veriVeGirisIslemleri.py : Giriş ve sunucuya veri gönderme işlemlerinin yapıldığı dosyadır.

ekran görüntüsü, oturma, cep telefonu, ekran içeren bir resim

Açıklama otomatik olarak oluşturuldu

AnaEkranIslemleri.py : Programın GUI ile etkileşimini düzenler

ekran görüntüsü, ekran, oturma, televizyon içeren bir resim

Açıklama otomatik olarak oluşturuldu

iç mekan, ekran görüntüsü, oturma, ekran içeren bir resim

Açıklama otomatik olarak oluşturuldu

inputs/ klasörü : Okunacak optik formların kopyalandığı ana klasördür. Ayrıca template.json ve omr\_marker.jpg dosyalarını içerir. Bu dosyalar formun okunması için kullanılan varsayılan dosyalardır. Kullanıcılar kendi oluşturdukları formlara göre template.json dosyasını yeniden oluşturabilir.

outputs/ klasörü : Optik formlar okunduktan sonra sonuç ile ilgili dosyaların kayıt edileceği yerdir.

outputs/Results/ klasörü : Okunan formların sonuçlar bu klasörde Results\_XXXX.csv formatında saklanır.

template.py :

utils.py :

## Zaman Çizelgesi

Proje geliştirme sürecine öncelikle proje ekibinin ortak paydada ihtiyaç duyduğu ve büyük toplulukların da karşılanabilir bir ihtiyacına odaklanması düşünülerek başlanmıştır. Optik Form Okuyucu projesine karar verildikten sonra öncelikle mevcut açık kaynaklı ve python kodlama ile oluşturulmuş benzeri uygulamaların hangi düzeye kadar ulaştıkları ve hedef projeye yakın olan projelerin durumları incelenmiştir. Karar verilen başlangıç yapısı öncelikle projenin ihtiyaçlarına göre geliştirilerek OMR çekirdek okuma yazılımı istendik düzeye eriştirilmiştir. Ardından bu çekirdek ile çalışabilecek masaüstü GUI tasarımı oluşturulmuş ve aynı süreçte okuma verilerinin değerlendirmesini yapacak olan web arayüzünün geliştirmesi tamamlanmıştır. Son olarak da masaüstü GUI yazılımının web arayüzü ile haberleşmesi hedeflenmiştir.

# Optik Form Okuyucu Gerçekleştirme Raporu

## Karşılaşılan Sorunlar ve Uygulanan Çözümler

Projenin gerçekleşmesi sürecinde karşılaşılan en büyük sorun, üretilmesi istenen projenin sıfırdan kodlanmasının süre olarak imkansızlığı olmuştur. Bir diğer sorun da, projeyi üretecek olan grubun üyelerinin birbirini ve yeteneklerini tanımıyor, projenin geliştirmesinde kimin ne türde yeteneğinin olduğu ve bu yeteneğini projeye yansıtabilecek müsaitliğinin olup olmadığı ve varsa bile buna fırsatının ne kadar süreyle olabileceğinin belirsizliğidir. Ekip liderinin olmayışı da, sürecin yönetilmesinde ve projenin tamamlanmasında frenleyici bir unsurdur.

Bu sorunların farkındalığı ile başlayan proje geliştirme süreci, pozitif yaklaşımlar ve planlamalarla olabilecek en optimum düzeyde aşılmaya çalışılmıştır.

Süre sorununu aşmak için “tekerleği yeniden icat etmeme” ilkesinden ola çıkılarak, hedeflenen projeye en yakına getirilmiş açık kaynaklı uygulamaya ulaşmak için yapılan araştırma neticesinde oldukça zaman tasarrufu sağlanmıştır. Grup üyelerine ait tecrübe belirsizliğini gidermek için whatsapp uygulaması üzerinde sanal toplantılar düzenlenerek projeye kimin nasıl bir katkı sağlayabileceği belirlenmiştir. Ekip liderinin olmaması ve grup üyelerine görev dağılımının sağlıklı gerçekleştirilememesi sorunu ise öncelikle ekibin kendi içinde web, çekirdek ve masaüstü olarak ayrılması ve her bir alt ekibin de iç görev dağılımında birbirinin kaldığı yerden devam etme ve eksik uyguladığını tamamlama şeklinde ilerlemiştir. Bunun yanında bir ekibin işini bitirdiğinde diğer alt ekibe yardım etmesi ile de sürecin başarımının arttırılması sağlanmıştır.

## Proje Bileşenleri ve Görevleri

# Optik Form Okuyucu

Optik formları tarayıcınızı 🖨 veya cep telefonunuzu 🤳 kullanarak tarayıp okutun.

## Telif Hakkı

Bu proje, GNU Genel Erişim Lisansı ile dağıtılmaktadır.

### Desteklenen İşletim Sistemi

Windows işletim destekleniyor olsa da, hatadan arınmış bir deneyim için **Linux** tavsiye edilir.

### Bağımlılıkların Kurulumu

python -m pip install --user --upgrade pip

python -m pip install --user opencv-python

python -m pip install --user opencv-contrib-python

**Bilgi:** Linux işletim sisteminde çalışırken bazı kütüphanelerin yüklenmesi gerekebilir.

#### Derleme İçin Gerekli Kütüphaneler ve Kurulumu

cd Optik-Form-Okuyucu/

python -m pip install --user -r requirements.txt

**Bilgi:** Eğer bazı kütüphanelerin zaten kurulu olduğu uyarısı alırsanız, --ignore-installed parametresini yukarıdaki komuta ekleyip tekrar çalıştırmayı deneyin.

### Programı Çalıştırma

1. Optik formlarınızın ve şablon json dosyanızın olduğu klasörü inputs dizinine taşıyın (Örnek kullanım için aşağıdaki yöntemi kullanabilirsiniz):
2. # Bilgi: inputs dizininde önceden kalma dosyaları öncelikle silmelisiniz.

cp -r ./ornekler/taslak1 inputs/

1. Optik okuma scriptini çalıştırın:
2. # Bilgi: main.py scripti, içinde açıklamaları da yer alan --setLayout , --autoAlign , --noCropping , --inputDir , --outputDir , --template parametreleriyle de çalıştırılabilir. AnaEkranIslemleri scripti bu parametreleri arayüz ile uygulamak içindir.

python main.py

## Github Yükleme Süreci

Projenin Github’a yüklenmesi için, Visual Studio Code ve Github Desktop yazılımlarının birlikteliğinden yararlanılmış ve başarılı araçları sayesinde otomatize edilmiştir. Projeyi kendi repository’sine dahil etmek isteyenlerin basitçe bir fork oluşturup aynı yazılımlarla içeriye alması yeterli olacaktır.

# Optik Form Okuyucu Test Raporu

## Karşılaşılan Sorunlar ve Uygulanan Çözümler

Yazılımın sağlıklı çalıştırılması için farklı optik formlar denenmelidir. Ayrıca okulların açık olduğu dönemlerde farklı şehirlerdeki ve branşlardaki öğretmenlerle proje paylaşılmalı ve kendilerinden optik form tanımı yaparak programın başarımını test edip geri dönüş sağlamaları istenmelidir.

## Test Sürecinde Kullanılan Modüller

Programı test sürecinde ücretsiz bir web sunucu hizmeti kiralanmıştır. Böylece çekirdeğin üretip masaüstü GUI tarafından gönderilen optik form okuma datasının mekan bağımsız olarak işlenebilmesi hedeflenmiştir.

## Değerlendirme Kriterleri

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **KRİTERLER** | Ekip Üyeleri | Eğitmen 1 | Eğitmen 2 | Eğitmen 3 | Ortalama |
| 1. Analiz Raporunun Tamamlanması |  |  |  |  |  |
| 1. Tasarım Raporunun Tamamlanması |  |  |  |  |  |
| 1. Gerçekleştirim Raporunun Tamamlanması |  |  |  |  |  |
| 1. Gantt Diagramı |  |  |  |  |  |
| 1. Arayüz tasarımı |  |  |  |  |  |
| 1. Veri Tasarımı-Sınıf Tasarımı |  |  |  |  |  |
| 1. Kullanıcı Yardım Dökümanı |  |  |  |  |  |
| 1. Programın Çalıştırılması |  |  |  |  |  |
| 1. Yazılım Test Çalışması |  |  |  |  |  |