

Optical Mark Reader (OCR) Modülü: Teknik Sunum Notu

Genel Bakış

OCR modülü, belgelerden otomatik olarak metin ve veri çıkarımını sağlayarak veri analizine hızlı bir altyapı sunar. Aşağıdaki bölümlerde, bu modülün işleyişi, teknik altyapısı ve geliştirme süreci detaylandırılmıştır.

1. Giriş

Amaç: Belgelerden otomatik metin ve veri çıkarımı sağlamak. **Kullanılan Teknolojiler:** Python, OpenCV, OCR.space API, Django, YAML.

2. Modülün İşleyiş Akışı

a. Görüntü Yükleme ve Ön İşleme

- Yükleme:** Belge ve şablon görüntüleri cv2.imread ile yüklenir.
- Ön İşleme Adımları:**
 - Kırpma:** Kenar boşlukları crop_borders fonksiyonuyla temizlenir.
 - Yeniden Boyutlandırma:** resize_image fonksiyonuyla uygun boyutlara ayarlanır.
 - Eğiklik Düzeltme (Deskew):** deskew_image ile Hough dönüşümü kullanılarak eğiklik giderilir.
 - Kontrast ve Gürültü Azaltma:** CLAHE (Contrast Limited Adaptive Histogram Equalization) tekniği, parlaklık kontrastını yerel olarak iyileştirir ve ayrıntıları daha belirgin hale getirir. Morfolojik işlemler ise görüntüdeki gürültüyü azaltmak ve istenmeyen ufak detayları temizlemek için kullanılır. Bu yöntemler görüntünün OCR işlemi için daha uygun hale gelmesini sağlar.
 - Eşikleme:** Adaptif eşikleme, bir görüntüdeki her bölge için dinamik bir eşik değeri hesaplayarak, metin ve işaretlemelerin daha net bir şekilde ayrılmasını sağlar. Bu teknik, düzensiz aydınlatma koşullarında bile başarılı sonuçlar elde edilmesini sağlar.

b. Görüntü Hizalama

- Feature Matching:** ORB (Oriented FAST and Rotated BRIEF) algoritması, görüntüler arasındaki önemli noktaları (keypoints) algılar ve tanımlar. BFMatcher (Brute Force Matcher) ise bu noktalar arasındaki benzerlikleri eşleştirerek iki görüntüyü hizalamaya olanak tanır. Bu teknik, belgelerin doğru konumlandırılmasında önemlidir.
- Alternatif Yöntem:** Hizalama başarısız olursa yedek bir yöntem devreye girer.

c. OCR İşlemi

- OCR.space API Kullanımı:** Bu API, metinleri görüntülerden çekmek için bulut tabanlı bir OCR teknolojisi sunar. Özellikle çoklu dil desteği ve karmaşık belgelerdeki metinlerin hassas çıkarılmasında etkilidir.
- Metin Normalizasyonu:** Çıkarılan metinler büyük harfe dönüştürülerek ve özel karakterler temizlenerek standart bir formatta sunulur. Bu, sonraki veri işleme adımlarının daha tutarlı ve hatasız olmasını sağlar.

d. Veri Çıkarma ve Analiz

- Başlık Koordinatları Bulma:** find_heading_coordinates ile belirli başlıkların konumları tespit edilir.
- ROI (Çıkarılacak Alan) Belirleme:** extract_roi ile belirlenen alanlardan veri çıkarılır.
- Cevaplar ve Diğer Bilgiler:**

- Cevap Seçenekleri:** Doluluk oranlarına göre seçenekler `extract_answers` ile bulunur.
- İsim ve Öğrenci Numarası:** Harfler ve rakamlar sırasıyla `extract_name` ve `extract_student_number` ile alınır.
- Test Grubu:** `extract_test_group` ile seçili test grubu tespit edilir.

e. Sonuçların Kaydedilmesi ve Görselleştirme

- Kaydetme:** Elde edilen veriler `save_results` ve `save_results_to_db` ile JSON veya veritabanına kaydedilir.
- Görselleştirme:** ROI alanları görüntü üzerine çizilerek kaydedilir.

3. Teknik Detaylar

Kütüphaneler

- OpenCV:** Görüntü işleme ve analiz.
- RapidFuzz:** Metin benzerlik hesaplamaları.
- YAML:** Konfigürasyon yönetimi.
- Django:** Veritabanı işlemleri ve ayar yönetimi.

Konfigürasyon Yönetimi

- config.yaml:** Tüm parametreler bu dosyada tanımlanarak esneklik sağlanır.
- Parametreler:** Görüntü boyutlandırma, deskew ayarı, OCR dil seçimi, ROI koordinatları vb.

Hata Yönetimi ve Logging

- Logger:** Tüm işlem adımları ve hatalar detaylı şekilde kaydedilir.
- Hata Kontrolleri:** Her adımda potansiyel hatalar kontrol edilerek uygun mesajlar loglanır.