

El Yazısı Reçete Tanıma (OCR) Proje Planı

Proje Adı

PaddleRx: Yapay Zeka Destekli El Yazısı Reçete Çözümleyicisi

Proje Hedefi

PaddleOCR'ın HTR (Handwritten Text Recognition - El Yazısı Metin Tanıma) yeteneklerinden yararlanarak, doktorların el yazısıyla yazdığı reçetelerdeki ilaç isimleri, dozajlar ve kullanım talimatları gibi temel bilgileri yüksek doğrulukla tespit edip yapılandırılmış bir JSON çıktısına dönüştüren bir prototip geliştirmek.

Temel Teknolojiler

Kategori	Teknoloji	Amaç
Model	PaddleOCR (Önceden Eğitilmiş HTR Modeli)	Metin tespiti (detection) ve tanıma (recognition).
Geliştirme Dili	Python	Çoğu AI/ML kütüphanesi Python ile uyumludur.
Veri Yapılandırma	Pandas, JSON	Veri işleme ve yapılandırılmış çıktı oluşturma.
Arayüz (Opsiyonel)	Streamlit / Flask	Modelin kullanılacağı basit bir web arayüzü.
Bağlam Anlama	NLP (Doğal Dil İşleme) Teknikleri	Tanınan metinlerin tıbbi terim olup olmadığını anlama.

Aşama 1: Ortam Kurulumu ve Veri Hazırlığı

1.1. Geliştirme Ortamının Kurulumu

- Gereksinimler: Python (3.8+), pip.
- PaddlePaddle (PaddleOCR'ın ana kütüphanesi) ve bağımlılıklarının kurulumu.
- Gerekli diğer kütüphanelerin (OpenCV, NumPy vb.) kurulması.

1.2. Reçete Veri Seti Toplama

- **Kritik Adım:** Modelin "doktor el yazısını" öğrenmesi için yüksek kaliteli ve çeşitli reçete görüntülerinden oluşan bir veri seti toplanır.
- Veri Seti İçeriği: Farklı doktorlara, farklı reçete formatlarına ve zorluk seviyelerine ait görseller.
 - *Not: Gerçek hasta verisi kullanılacaksa, mutlaka anonimleştirilmelidir.*
- Veri Setinin Bölünmesi: Eğitim (%80), Doğrulama (%10) ve Test (%10) setlerine ayrılması.

Aşama 2: Veri Etiketleme ve Ön İşleme

2.1. Veri Etiketleme (Labeling)

Elle yazılmış reçete OCR'ı için iki temel etiketleme türü gereklidir:

1. **Tespit (Detection) Etiketlemesi:** Reçete görseli üzerinde ilaç adı, dozaj, talimat gibi ana bileşenlerin bulunduğu bölgelerin (sınırlayıcı kutu/bounding box) işaretlenmesi.
 - *Örn. {"transcription": "Aspirin", "points": [[x1, y1], [x2, y2]]}*
2. **Tanıma (Recognition) Etiketlemesi:** Her bir sınırlayıcı kutudaki el yazısının dijital metin karşılığının yazılması. Bu, modelin "Acetaminofen" yazan bir el yazısını doğru şekilde tanıyabilmesi için temeldir.

2.2. Görüntü Ön İşleme

- Görüntü kalitesini artırmak, gürültüyü azaltmak ve kontrastı optimize etmek için algoritmalar (Gaussian Bulanıklığı, Eşikleme/Thresholding vb.) uygulanır.
- Görüntüler, modelin kabul ettiği standart boyuta (örneğin 640x640) getirilir.

Aşama 3: Model İnce Ayarı (Fine-Tuning) ve Eğitim

3.1. Önceden Eğitilmiş Modeli Seçme

- PaddleOCR'ın genel HTR (Handwritten Text Recognition) veya yüksek doğruluklu metin tanıma için kullanılan bir önceden eğitilmiş modeli (örneğin PP-OCRv3 HTR) temel model olarak seçilir.

3.2. Modelin Tıbbi Verilerle Eğitilmesi

- **Fine-Tuning:** Aşama 2'de hazırlanan etiketli veri seti kullanılarak temel modelin son katmanları (veya tüm katmanları) yeniden eğitilir. Bu, modelin genel el yazısından çıkıp **tıbbi el yazısı** uzmanı olmasını sağlar.
- Eğitim Parametreleri: Epoch sayısı, öğrenme oranı, optimizasyon algoritması (Adam vb.) belirlenir ve eğitim başlatılır.

3.3. Değerlendirme ve Optimizasyon

- Eğitim sırasında doğrulama seti (Validation Set) kullanılarak model performansı izlenir.
- Model, test seti üzerinde çalıştırılarak doğruluk (Accuracy) ve F-skoru gibi metriklerle

değerlendirilir.

- Düşük doğruluk varsa: Daha fazla veri toplanır veya eğitim parametreleri ayarlanır.

Aşama 4: Uygulama Geliştirme ve Anlam Çıkarımı

4.1. OCR Motorunun Entegrasyonu

- Eğitilmiş model bir API (REST veya Python sınıfı) içine sarılır.
- Kullanıcıdan alınan reçete görseli bu API'ye beslenir ve metin çıktısı alınır.

4.2. Yapılandırılmış Çıktı ve Anlam Çıkarımı

OCR ile elde edilen ham metin, reçetedeki anlam karmaşasını çözmek için işlenmelidir:

- Named Entity Recognition (NER) / İsimlendirilmiş Varlık Tanıma:** Metin içindeki kelimelerin ne anlama geldiği (ilaç adı mı, dozaj mı, kullanım süresi mi) tespit edilir.
- Tıbbi Terim Kontrolü:** Tanınan kelimeler, bir farmasötik veya tıbbi terim veritabanı ile karşılaştırılarak hatalı tanımlar (örneğin "Parol" yerine "Parol1" yazılması) düzeltilir.
- JSON Çıktısı:** Tüm bu işlemler sonucunda, reçete bilgileri standart ve makine tarafından okunabilir bir JSON formatına dönüştürülür:

```
{
  "hasta_id": "...",
  "ilaclar": [
    {
      "ad": "...",
      "dozaj": "...",
      "siklik": "..."
    }
  ]
}
```

4.3. Kullanıcı Arayüzü (UI) Prototipi

- Kullanıcıların reçete görseli yükleyebileceği ve anlık olarak dijital çıktıyı görebileceği basit bir arayüz (Streamlit veya Flask ile) oluşturulur.
- Okunamayan veya düşük güvenilirlikle tanınan bölgeler arayüzde işaretlenerek (örneğin sarı renkle) eczacıya manuel kontrol uyarısı verilir.

Önemli Zorluklar ve Azaltma Yöntemleri

Zorluk	Açıklama	Azaltma Yöntemi
Yazı Tutarsızlığı	Doktorların el yazılarının çok değişken olması.	Aşama 1'de toplanan veri setinde mümkün olduğunca

		fazla farklı el yazısı örneği kullanmak.
Tıbbi Terminoloji	Modelin, genel metin yerine karmaşık ilaç isimlerini doğru tanıması.	Aşama 3'te büyük bir tıbbi terim sözlüğünü model eğitime dahil etmek ve Aşama 4'te NER/sözlük kontrolü yapmak.
Düşük Güvenilirlik	Reçetede kritik bir bilginin yanlış okunması (örn. dozaj hatası).	Aşama 4'te düşük güvenilirlik skoruna sahip metinler için kullanıcı arayüzünde görsel uyarılar sağlamak.