

# Akciğer Görüntü Verileri İle Hastalık Tahmin Modeli

1<sup>st</sup> Süleyman Talha Sarı  
Bilişim Sistemleri Mühendisliği 3.sınıf

Kocaeli Üniversitesi

2<sup>nd</sup> Melih İyigören

Bilişim Sistemleri Mühendisliği

Kocaeli Üniversitesi

**Özet—** Bu çalışma, internetten toplanan akciğer görüntülerini işleyerek hastalık tahmini yapabilen bir yapay zeka modeli geliştirmeyi amaçlamaktadır. Görüntüler yeniden boyutlandırma, parlaklık ve kontrast ayarı ile veri artırma işlemlerinden geçirilmiştir. Python[2], Selenium[3], BeautifulSoup[4] ve OpenCV[5] kullanılarak veriler işlenmiş, model tahmin gücünü artırmak için ön işleme adımları uygulanmıştır. Bu proje, medikal tanı süreçlerinde otomasyonu desteklemeyi hedeflemektedir.

**Anahtar Kelimeler—**görüntü işleme, yapay zeka modeli, python, veri işleme.

## GİRİŞ

Bu projede, akciğer hastalıklarının tahminine yönelik bir yapay zeka modeli geliştirmek amacıyla akciğer görüntüleri toplandı ve işlendi. Görüntüler, veri toplama kısıtlamalarına rağmen internetten elde edildi ve kullanılabilir olanlar seçildi.

Verilerin standart hale getirilmesi için, Python ve OpenCV kütüphanesi kullanılarak boyut, parlaklık ve kontrast ayarlamaları gerçekleştirildi. Bu ön işleme adımları, modelin daha doğru tahmin yapabilmesi için görüntü verisinin kalitesini ve tutarlılığını artırmak amacıyla uygulanmıştır. Proje boyunca veri artırma teknikleri kullanılarak veri seti zenginleştirilmiş, böylece oluşturulacak olan modelin genelleme yeteneği geliştirilmiştir.

## I. PROJE AŞAMALARI

### A. Veri Kaynağı Olacak Website Seçimi(Heading 2)

Bu projede akciğer hastalıklarının tespiti amacıyla kullanılacak görsellerin toplandığı web sitesi **Radiopaedia** (<https://radiopaedia.org>)[1] olarak seçilmiştir. Radiopaedia, tıp alanında geniş bir görsel veri tabanına sahip, özellikle tıbbi görüntüleme konusunda zengin kaynaklar sunan bir platformdur. Akciğer hastalıklarıyla ilgili çok sayıda yüksek kaliteli tıbbi görsel barındıran bu site, projede kullanılacak veriler için ideal bir kaynaktır. Web sitesi, tıbbi görüntülerin doğru etiketlenmiş ve güvenilir kaynaklardan alınmasını sağlayarak, modelin doğruluğunu artırmaya yardımcı olacaktır.

### B. Veri Çekme Teknolojileri ve Dil Seçimi

Veri çekme işlemi için **Python** programlama dili tercih edilmiştir. Python, veri işleme ve web scraping işlemleri için yaygın olarak kullanılan güçlü bir dil olup, geniş kütüphane desteği sunmaktadır. Veri çekme sürecinde, **Selenium** ve

**BeautifulSoup** gibi popüler Python paketleri kullanılmıştır. Selenium, dinamik web sayfalarından veri çekebilmek için, web tarayıcılarını otomatikleştirme yeteneği sağlar ve JavaScript tarafından oluşturulan içeriği düzgün bir şekilde alabilmemize olanak tanır. BeautifulSoup ise, statik HTML içeriklerini parse ederek, verileri hızlı ve verimli bir şekilde çıkarmamıza yardımcı olmuştur. Bu araçlar, Radiopaedia gibi etkileşimli ve dinamik içeriklere sahip web sitelerinden görsel ve metin verilerini toplamak için ideal seçimlerdir. Python'un esnek yapısı ve bu araçların sağladığı güçlü özellikler, veri çekme işleminin güvenli, verimli ve doğru bir şekilde gerçekleştirilmesini sağlamıştır.

## II. WEB KAZIMA İLE İNTERNET ÜZERİNDEN VERİ ÇEKİMİ

Yazılan Python kodu, Radiopaedia sitesinden "lung" (akciğer) ile ilgili görselleri sitemap URL'leri üzerinden otomatik olarak indirir. Kod, Selenium kullanarak her bir sitemap sayfasını açar, XML yapısını işler, ve ilgili görsellerin URL'lerini alır. Belirli kriterlere uyan görseller requests ile indirilir ve bir klasörde saklanır. İndirme işlemi sırasında bağlantı veya yükleme hataları oluşursa, WebDriver yeniden başlatılarak işlem devam eder.

### A. Sitemap İşleme Fonksiyonu

Python dilinde yazılmış olan bu fonksiyon bir sitemap URL'si alır, içindeki görselleri işler ve "lung" kelimesini içeren görselleri indirir.

- driver.get(sitemap\_url): Sayfa tarayıcıda açılır.
- root = ET.fromstring(sitemap\_content): XML verisini alır ve ElementTree kökünü oluşturur.
- findall("./{\*}image"): Tüm image elemanlarını arar.
- caption ve title: Görsellerin açıklama ve başlıkları kontrol edilir.
- download\_image(): image\_url verilen dosya adına kaydedilir.

### B. Sitemap URL Listesi

```
# Sitemap URL'leri
sitemap_urls = [
    "https://radiopaedia.org/sitemap-articles_1.xml",
    "https://radiopaedia.org/sitemap-articles_2.xml",
    "https://radiopaedia.org/sitemap-articles_3.xml",
    "https://radiopaedia.org/sitemap-articles_4.xml",
    "https://radiopaedia.org/sitemap-articles_5.xml",

```

- İşlenecek tüm sitemap URL'leri bir liste olarak tanımlanmıştır.

### C. Sitemap URL'lerin İşlenmesi

```
# URL'leri işleme alma
for sitemap_url in sitemap_urls:
    print(f"İşlem başlatılıyor: {sitemap_url}")
    process_sitemap(sitemap_url)
```

- process\_sitemap() fonksiyonu her bir URL için çağrılır.

### III. GÖRÜNTÜ VERİLERİNİ İŞLEME

Görüntü işleme için **OpenCV** (Open Source Computer Vision Library) kullanılmıştır. OpenCV, görüntü işleme ve bilgisayarla görme alanında yaygın olarak kullanılan açık kaynaklı bir kütüphanedir. Görüntüleri yeniden boyutlandırma, parlaklık ve kontrast ayarlamaları yapma gibi temel işlemler için güçlü fonksiyonlar sunar. Ayrıca, bu kütüphane, yüksek verimli ve hızlı işlem yapma kapasitesiyle büyük veri setleri üzerinde çalışmayı mümkün kılar. OpenCV, görüntüleri doğru formatta işleyerek, derin öğrenme modellerine uygun hale getirmek için ideal bir araçtır.

#### A. Python Kodlama Kısım

- İlk olarak görseller üzerinde işlem yapabilmek için gerekli kütüphaneler aktarılmaktadır.
- Verilerin işlenmesi için kullanılacak olan parametreler tanımlanmıştır.

```
if not os.path.exists(output_folder):
    os.makedirs(output_folder)
```

- Bu blok, çıktı görsellerinin kaydedileceği klasörün var olup olmadığını kontrol eder. Eğer klasör mevcut değilse, os.makedirs() fonksiyonu ile belirtilen output\_folder adıyla yeni bir klasör oluşturulur.

```
def normalize_brightness_contrast(image, target_brightness, target_contrast):
    # Görüntüyü normalize etmek için ortalama ve standart sapmasını hesapla
    mean, stddev = cv2.meanStdDev(image)

    # Parlaklık ayarı
    brightness_adjusted = image - mean[0][0] + target_brightness

    # Kontrast ayarı
    contrast_factor = target_contrast / (stddev[0][0] + 1e-5) # Sıfıra bölünmeyi önlemek için küçük bir sayı ekledik
    normalized_image = cv2.convertScaleAbs(brightness_adjusted, alpha=contrast_factor, beta=0)

    return normalized_image
```

- Bu fonksiyon, verilen görüntünün parlaklık ve kontrastını normalize etmek için kullanılır:

```
# Görüntüleri işleme
for filename in os.listdir(input_folder):
    img_path = os.path.join(input_folder, filename)
    img = cv2.imread(img_path, cv2.IMREAD_GRAYSCALE) # Görselleri gri tonlamada açıyoruz

    if img is not None:
        # Yeniden boyutlandır
        resized_img = cv2.resize(img, target_size)

        # Parlaklık ve kontrastı normalize et
        normalized_img = normalize_brightness_contrast(resized_img, standard_brightness, standard_contrast)

        # Çıktı dosyasını kaydet
        output_path = os.path.join(output_folder, filename)
        cv2.imwrite(output_path, normalized_img)
        print(f"{filename} başarıyla işlendi ve kaydedildi.")
    else:
        print(f"{filename} yüklenemedi.")
```

- Görüntülerin işlenmesini ve işlenen görüntülerin kaydedilmesini sağlayan kod parçacığı.

### IV. SONUÇ

Bu çalışmada, Radiopaedia sitesinden akciğerle ilgili görsellerin otomatik olarak toplanması, ön işlenmesi ve veri setinin oluşturulması başarıyla gerçekleştirilmiştir. Kullanılan Python kodu, Selenium ve requests kütüphaneleri aracılığıyla belirlenen sitemap bağlantılarını tarayarak uygun görselleri indirip parlaklık, kontrast ayarı ve yeniden boyutlandırma gibi işlemlerle standardize etmiştir. Bu görseller, makine öğrenmesi modeline uygun hale getirilerek akciğer hastalıklarının tespit edilmesi üzerine eğitilecek bir veri kümesi oluşturulmuştur. Çalışmanın sonucunda elde edilen veri seti, modelin performansını artıracak nitelikte temiz ve homojen hale getirilmiştir. Gelecekte, daha geniş kapsamlı ve çeşitli görsellerle veri seti geliştirilerek modelin doğruluğunu ve genellenebilirliğini artırmak mümkün olacaktır.

#### - Kaynakça

- [1] <https://radiopaedia.org/1>.
- [2] Python.org
- [3] Selenium.dev
- [4] <https://medium.com/@fatihazir/beautifulsoup-k%C3%BCt%C3%BCphanesi%CC%87-i%C3%87le-python-kullanarak-web-scraping-uygulamasi-65755f1da27e>
- [5] Opencv.org