Tümör Tiplerinin Bilgisayarlı Tomografi Görüntüleri ile Sınıflandırılmasına İlişkin Veri Setinin Toplanması

Umut Gülfidan  
*Bilişim Sistemleri Mühendisliği*  
Kocaeli ÜniversitesiKocaeli, Türkiye  
[umutgulfidan41@gmail.com](mailto:umutgulfidan41@gmail.com)

Ahmet Efe Tosun  
*Bilişim Sistemleri Mühendisliği*  
Kocaeli Üniversitesi *Kocaeli*, Türkiye  
[ahefto@gmail.com](mailto:ahefto@gmail.com)

*Özet*— Bu çalışma, beyin tümörü tiplerinin bilgisayarlı tomografi (BT) görüntüleri ile sınıflandırılması için kapsamlı bir veri setinin toplanması ve ön işleme adımlarının uygulanmasını kapsamaktadır. Menenjiyom, hipofiz tümörü, gliom ve tümörsüz beyin olmak üzere dört sınıfa ait veriler, web kazıma yöntemi ile elde edilmiştir. İlgili kodlar “Java” programlama diliyle yazılmıştır. Toplanan görüntüler veri artırma teknikleri ile çoğaltılarak veri seti kapsamlı hale getirilmiştir. Bu veri setinin, beyin tümörlerinin otomatik sınıflandırılmasında ve tanı süreçlerinde yapay zeka modellerini destekleyerek tanı doğruluğunu artırması amaçlanmaktadır.

Anahtar kelimeler—Beyin tümörü, bilgisayarlı tomografi, veri seti, sınıflandırma, web kazıma, meningioma, pituitary, glioma, java

# Giriş

Beyin tümörleri, ciddi sağlık sorunlarına yol açabilen karmaşık patolojik yapılardır ve doğru bir sınıflandırma yapılması tanı ve tedavi sürecini önemli ölçüde iyileştirebilir. Bilgisayarlı tomografi (BT) görüntüleri, tıbbi görüntülemede yaygın olarak kullanılarak, tümörlerin tiplerinin belirlenmesine olanak tanımaktadır. Son yıllarda, makine öğrenimi ve derin öğrenme tabanlı modellerin gelişimi, BT görüntülerinin otomatik olarak analiz edilmesini sağlayarak bu alanda önemli ilerlemelere yol açmıştır. Ancak, bu tür çalışmalar için yeterli ve çeşitli veri setlerinin bulunması gereklidir. Bu çalışma, menenjiyom, hipofiz tümörü, gliom ve tümörsüz beyin sınıflarını içeren kapsamlı bir veri setinin toplanması ve bu veri setinin sınıflandırma modelleri için hazırlanmasını amaçlamaktadır.

# Proje Kurulumu ve Gereksinimleri

Proje, JDK 23 ile geliştirilmiştir. Projeyi kurmadan önce gerekli yazılımların ve kütüphanelerin yüklenmiş olduğundan emin olun.

## Gereksinimler

### Java Development Kit (JDK) 23: Projeyi geliştirmek ve çalıştırmak için gerekli Java derleyici.

### Selenium: Web tarayıcılarını otomatikleştirmek için kullanılan kütüphane. Projede 4.25.0 sürümü kullanılılmıştır. Bu sürümü ve bu sürümden sonraki sürümleri kullanabilirsiniz.

### OpenCV: Görüntü işleme uygulamaları için kullanılan kütüphane. Projede 4.10.0 sürümü kullanılmıştır.Bu sürümü ve bu sürümden sonraki sürümleri kullanabilirsiniz.

### JSON.org (JSONObject): JSON formatındaki verileri işlemek için gerekli kütüphane.

Gerekli tüm kütüphanelere ilişkin bağlantılar proje GitHub sayfasında ve raporun ekler bölümünde verilmiştir.

## Kurulum Adımları

### İndirilen JDK dosyasını çalıştırın ve ekrandaki talimatları izleyerek kurulumu tamamlayın.

### Tercih ettiğiniz bir IDE kullanarak projeyi açın. JDK sürümünün doğru ayarlandığından emin olun.

### Projedeki kütüphane dizinleri ile ilgili hata alabilirsiniz. Kütüphane yollarını silerek, kendi indirdiğiniz kütüphaneleri doğru şekilde belirtin.

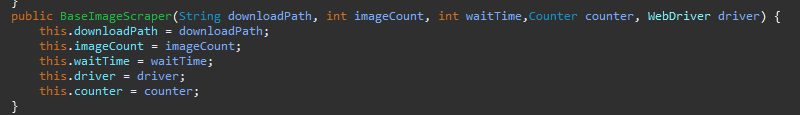
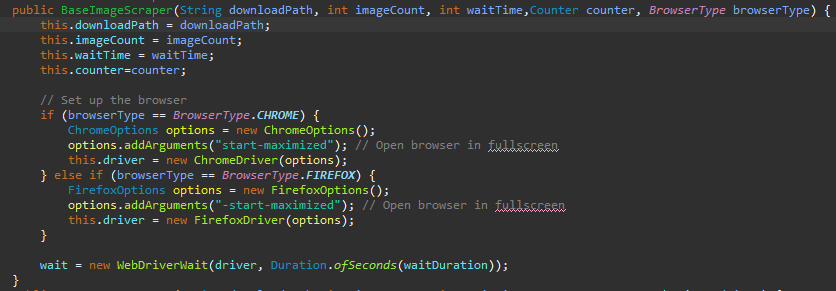
### Ana sınıfı çalıştırmadan önce örnek metotlara göz atın. Metotları ihtiyacınıza uygun yapılandırarak projeyi çalıştırabilirsiniz.

# Veri Toplama ve İlgili Kodların Açıklaması

## ImageScraper arayüzü ve “BaseImageScraper” ana sınıfı

Bütün web kazıma sınıflarının ortak bir şablona sahip olabilmesi için bir “ImageScraper” arayüzü (interface) oluşturulmuştur. Bu arayüzü uygulayan “BaseImageScraper” sınıfı, tüm web kazıma sınıflarında kullanılacak metotları ve özellikleri tanımlamaktadır.

“BaseImageScraper” sınıfında, dosyanın indirileceği dizini belirten downloadPath, indirilecek resim sayısını ifade eden imageCount, resimler arasında bekleme süresini belirten waitTime,waitDuration gibi alanlar tanımlanmış ve bu alanlara değer atamaları, yapıcı metot (constructor) ile sağlanmıştır. Ayrıca, hata sayısı, yapılan istek sayısı, indirme sayısı ve ID sayıları gibi verileri tutan counter alanı da bulunmaktadır. Bu alan, Counter sınıfının bir örneğidir ve ilgili verilerin arttırılması ve atanması için gerekli metotları içermektedir.

1. WebDriver parametreli yapıcı metot.
2. Enum Parametreli yapıcı metot.

“BaseImageScraper” sınıfındaki driver alanı için, Selenium kütüphanesinden bir WebDriver sınıfı örneği oluşturabilir veya oluşturduğumuz BrowserType enum'ı ile istediğiniz tarayıcıyı seçebilirsiniz. Ayrıca, sınıf içerisinde resim indirme, tarayıcı kapatma ve resim yolunu atama gibi işlemleri gerçekleştiren metotlar da mevcuttur.

Resimin kaydedileceği ad ve uzantılar için aşağıdaki bölümde değişiklikler yapabilirsiniz:

1. Kaydedilecek resmin ve uzantının belirtildiği alanlar.

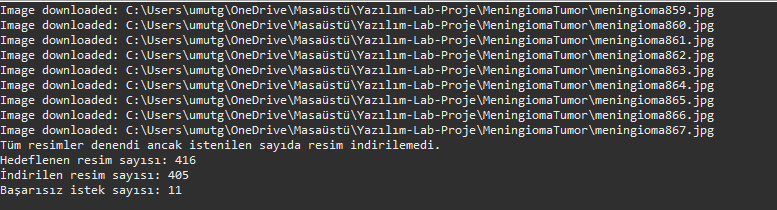
## Web Kazıma Sınıfları

İlgili veriler MedPix ve OpenI gibi, Ulusal Sağlık Enstitüleri (NIH) ve Ulusal Tıp Kütüphanesi (NLM) tarafından geliştirilen tıbbi görüntüleme web sitelerinden ve Google Görseller’den toplanmıştır.. Tüm bu kazıma işlemleri için “BaseImageScraper” sınıfından yeni sınıflar türetilmiştir. Yapıcı metotları eklenmiş ve web kazıma yapmak için “downloadImages” metotları geçersiz kılınmıştır (override). Dolayısıyla herhangi bir sınıfla web kazıma yapmak için ilgili sınıftan bir örnek türetip bu metod kullanılmalıdır. Aşağıda Google Görseller ile kazıma yapan bir örnek verilmiştir.

1. Web kazımaya ilişkin kod parçası

Burada Google Görseller yerine ilgili siteye ilişkin scraper sınıfı da kullanarak bu kod çalıştırılabilir. Örneğin “GoogleImageScraper” yerine “MedpixImageScraper” yazılarak Medpix’den veri kazınabilir. Yukarıdaki metodun parametreleri konsoldan kullanıcı girişi olarak alınarak dinamik bir yapı sağlanabilir.

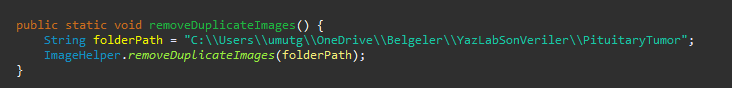
Yukarıdaki kod ana sınıfa eklenip çalıştırıldığında konsolda şuna benzer bir çıktı alınacak ve programın sonunda kaç resim indirildiği, HTTP isteklerinin sayısı ve oluşan hata sayısı gibi bilgileri içeren bir çıktı alınacaktır.

1. Web kazıma işleminde örnek çıktı.

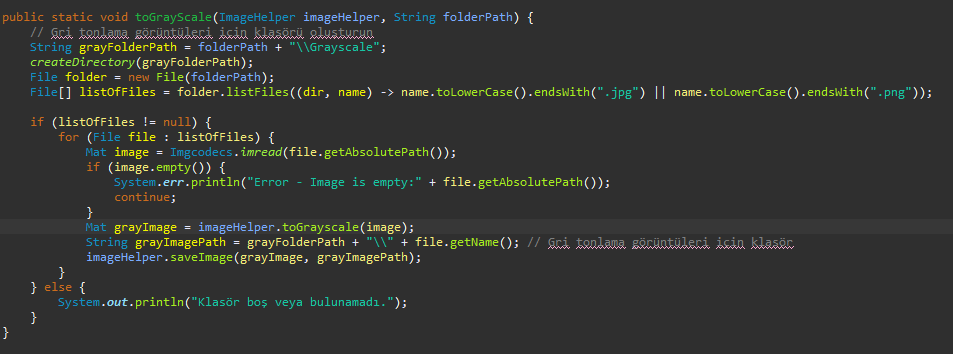
# Veri Ön İşleme

Bu projede veri ön işleme aşamasında, özellikle tekrar eden görüntülerin kaldırılması, görüntülerin gri tonlamaya çevrilmesi ve çeşitli veri artırma (augmentation) işlemleri uygulanmıştır. Bu işlemler için geliştirilen ve bir dizi görüntü işleme fonksiyonları içeren ImageHelper sınıfı kullanılmıştır.

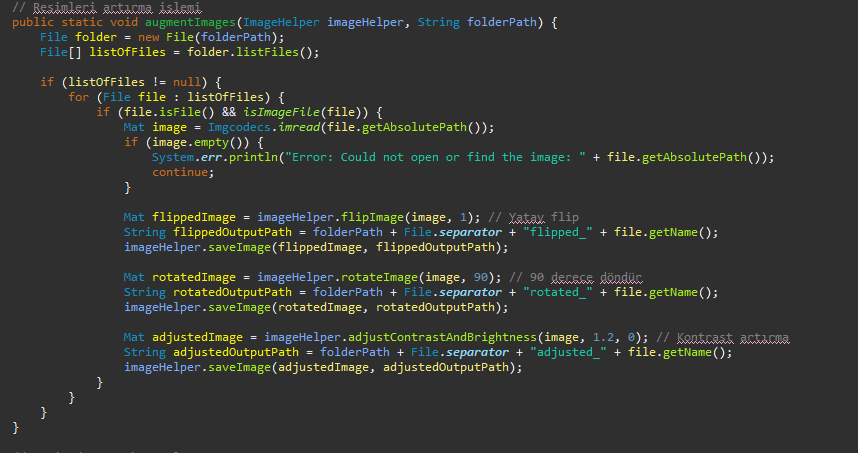
Öncelikle, tekrar eden resimleri tespit etmek ve kaldırmak amacıyla removeDuplicateImages metodu yazılmış ve kullanılmıştır. Bu metod, her bir dosyanın içerik hash’ini hesaplayarak aynı içeriğe sahip dosyaları tespit eder. Aynı hash değerine sahip resimlerden biri korunurken, diğer kopyalar silinmektedir. Bu sayede, veri kümesinde gereksiz yer kaplayan kopya verilerden arındırılmış, daha tutarlı bir veri seti elde edilmiştir. Ana sınıfta tekrar eden verlileri kaldırmak için aşağıdaki örneği inceleyebilirsiniz.

1. Tekrar eden verilerin kaldırılmasına ilişkin kod parçası

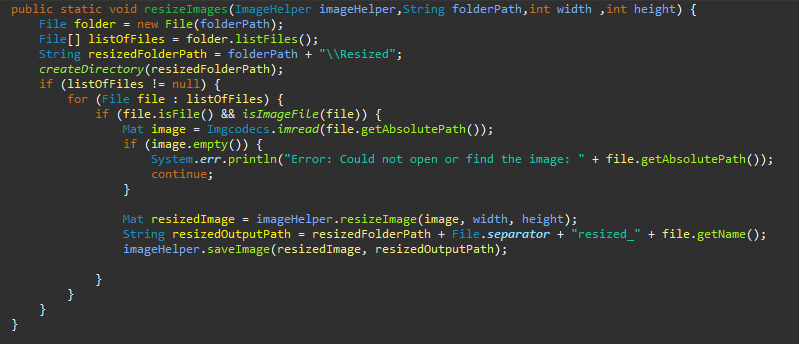
Tekrar eden veriler temizlendikten sonra, veriler gri tonlamaya çevrilmiştir. toGrayscale metodu, tüm resimleri gri tonlama (grayscale) formatına dönüştürerek veriyi sadeleştirir ve işlem yükünü azaltır. Gri tonlamaya dönüştürme, özellikle belirli analiz ve sınıflandırma işlemlerinde gereksiz renk bilgilerini ortadan kaldırarak, modelin daha hızlı ve doğru çalışmasına katkı sağlar. Aşağıda bir klasördeki tüm resimleri gri tonlamaya çeviren bir metot örnek verilmiştir. Bu kod çalıştırıldığın bulunulan dizin içerisinde “Grayscale” adında bir klasör açılacak ve gri tonlamaya çevirilmiş resimler buraya kaydedilecektir. ImageHelper sınıfında bunu gri tonlamaya çeviren kod OpenCv kütüphanesini kullanmaktadır.

1. Gri tonlamaya çeviren örnek kod parçası.

Ayrıca, veri setini zenginleştirmek ve modelin daha iyi genelleme yapabilmesi için döndürme, kontrast ayarlama, yansıtma (flip) gibi veri artırma işlemleri de gerçekleştirilmiştir. RotateImage, adjustContrastAndBrightness ve flipImage metodları kullanılarak verilerde çeşitli değişiklikler yapılarak daha geniş bir eğitim veri seti oluşturulmuştur. Bu işlemler, modelin farklı koşullarda daha iyi sonuçlar vermesini desteklemektedir. Aşağıda ImageHelper sınıfı kullanılarak veri arttırmaya ilişkin bir kod parçası yazılmıştır. Bu örnekte klasördeki her resim için yansıtma, kontrast ayarlama ve döndürme metotları kullanılarak verinin arttırılması sağlanmıştır.

1. Veri Arttırma işlemine ilişkin kod parçası

En son işlem olarak görseller tekrardan boyutlandırıldı. Görüntülerin boyutlandırılması işleminde, her bir resim belirlenen 512x512 ve 256x256 piksel boyutlarına yeniden ölçeklendirildi. Bu işlem, resimlerin analize ve eğitim aşamalarına uygun hale getirilmesi için gereklidir. Aşağıdaki kod, “resizeImages” fonksiyonu ile belirtilen genişlik ve yükseklik değerlerine göre görüntülerin yeniden boyutlandırılmasını gerçekleştirmektedir. Kodda, ImageHelper sınıfı ile resimlerin boyutlandırılması ve belirtilen klasöre kaydedilmesi sağlanmıştır.

1. Görsel yeniden boyutlandırma işlemine ilişkin ana sınıftaki örnek bir metot.

Örnek olarak, resizeImages metodu, verilen klasördeki tüm resimleri hedef boyutlara (örneğin, 256x256 piksel) yeniden ölçeklendirir ve yeni boyutlandırılmış dosyaları "Resized" adlı bir klasöre kaydeder:

# Sonuçlar ve Değerlendirme

Bu projede, menenjiyom, hipofiz tümörü, gliom ve tümörsüz beyin sınıflarına ait BT görüntülerinden oluşan bir veri seti toplandı ve çeşitli ön işleme adımları uygulandı. Java, Selenium ve OpenCV gibi araçlar kullanılarak veriler toplandı, çoğaltıldı ve işleme tabi tutuldu.

Veri seti, beyin tümörlerinin otomatik sınıflandırılmasında kullanılabilecek bir kaynak sunmaktadır. Gelecekte, bu veri setinin daha da genişletilmesi ve farklı yapay zeka modelleri ile test edilmesi planlanmaktadır.

##### Kaynaklar

1. *Selenium Documentation, "Selenium WebDriver Documentation,",* [*https://www.selenium.dev/documentation/*](https://www.selenium.dev/documentation/)
2. *D. Octaviani, "Automation of Google Image Scraping Using Selenium,",* [*https://medium.com/@dian.octaviani/method-1-4-automation-of-google-image-scraping-using-selenium-3972ea3aa248*](https://medium.com/@dian.octaviani/method-1-4-automation-of-google-image-scraping-using-selenium-3972ea3aa248)
3. GeeksforGeeks, "Rotating Images Using OpenCV in Java,", <https://www.geeksforgeeks.org/rotating-images-using-opencv-in-java/>
4. W3docs. "Getting a File's MD5 Checksum in Java." W3docs Snippets, <https://www.w3docs.com/snippets/java/getting-a-files-md5-checksum-in-java.html>

##### Ekler

JDK indirme linki: [JDK](https://www.oracle.com/tr/java/technologies/downloads/)

OpenCV indirme linki: [OpenCV](https://opencv.org/releases/)

Selenium indirme linki: [Selenium](https://www.selenium.dev/downloads/)

Proje Kaynak Kodları : [Proje Github Linki](https://github.com/umutgulfidan/WebImageScraper)

Elde Edilen Verilere İlişkin Tablo : [Excel](https://docs.google.com/spreadsheets/d/113_H-TnYGTuDJVdzqkttt-iReyHAHpU2B6Hdr_E7OhI/edit?gid=0#gid=0)

Veri Seti : [OneDrive](https://drive.google.com/drive/folders/1VNVsV2ZaD1ba_qC-zAIhhVVzZhHOTJXG?usp=sharing)