Kocaeli Üniversitesi, Teknoloji Fakültesi Bilişim Sistemleri Mühendisliği Bölümü: Yazılım Geliştirme Laboratuvarı I 2024-2025 Güz Proje I Proje Raporu

Selin AKPOLAT
Kocaeli Üniversitesi
B*ilişim Sistemleri Mühendisliği:*
Yazılım Geliştirme Laboratuvarı I
İzmit/Kocaeli
221307045@kocaeli.edu.tr

Derya GELMEZ
Kocaeli Üniversitesi
B*ilişim Sistemleri Mühendisliği:*
Yazılım Geliştirme Laboratuvarı I
İzmit/Kocaeli
221307055@kocaeli.edu.tr

Bu çalışma, Python programlama dili kullanılarak çeşitli internet sitelerinden kitap kapak görsellerinin toplanması, toplanan görsellerin elenmesi ve bu görsellerin sınıflandırma amacıyla çoğaltılmasını ele almaktadır. Web kazıma işlemleri, görsel veri kümesi oluşturma amacıyla yapılmıştır. Veri çoğaltma aşamasında, parlaklık-kontrast ayarı ve döndürme işlemleri gerçekleştirilmiştir.

Anahtar Kelimeler — Web Kazıma, Python, Görsel Veri Çoğaltma, Kitap Kapakları, Görsel Sınıflandırma

I. GİRİS

Görsel sınıflandırma problemlerinde modelin doğruluğunu artırmak için kaliteli ve çeşitlendirilmiş bir veri seti oldukça önemlidir. Bu çalışmanın amacı, kitap kapaklarından kategori tahmini yapmak için üç temel veri hazırlık aşaması üzerinde yoğunlaşmaktır: veri toplama, veri eleme ve veri çoklama. Bu aşamalar, kitap görsellerinden kategori tahminine yönelik bir veri seti oluşturmak ve model performansını optimize etmek için yapılmıştır.

II. YÖNTEMLER

A. ***Veri Toplama:*** İlk aşama olan veri toplamada, öncelikle yeterli data çekebilecek siteler araştırıldı. Siteler belirlendikten sonra sayfanın statik yapıda mı dinamik yapıda mı olduğu belirlendi. Buna göre kullanılacak kütüphanelerin araştırması yapıldı. Python'un Selenium ve Requests kütüphaneleri kullanılarak internet üzerinden kitap görselleri otomatik olarak çekildi. Bu işlemde:

- ***Web sürücüsü başlatma:*** Belirli bir web sitesine erişip kitap kapak görsellerini toplamak için Python programlama dili ve Selenium kütüphanesi kullanıldı. Bu amaçla, Selenium WebDriver aracı, Chrome tarayıcısını başlatmak üzere yapılandırıldı. İlk olarak, ChromeOptions nesnesi ile tarayıcı ayarları yapılandırılıp, çeşitli komutlarla tarayıcının tam ekran başlatılması sağlandı ve tarayıcının testin sonunda kapanması engellenerek açık kalması sağlandı; böylece kod çalıştırıldıktan sonra tarayıcı penceresi kapanmadı ve çıktılar gözlemlenebilir oldu. Yapılandırmalarla Chrome tarayıcısını başlatır ve bu sürücü ile istenilen web sitesine erişim sağlanır.

- ***Görselleri çekme ve kaydetme:*** Bu aşamada save_url fonksiyonu, Selenium WebDriver kullanarak kategori sayfalarını ziyaret edip ve her kitap için görsel URL'lerini toplandı. Bu URL'ler, ilgili kategori klasörlerinde .txt dosyalarına kaydedildi. download_images fonksiyonu ile kaydedilen URL'lerden HTTP GET isteğiyle görselleri indirildi; görseller başarılı bir şekilde indirildiğinde, Pillow (PIL) kütüphanesi ile 260x400 piksel boyutunda yeniden boyutlandırılıp JPEG formatında kaydedildi.

Bu işlemler sonucunda her kategoriden (kişisel gelişim, çocuk, tarih, çizgi roman, polisiye) 6000 civarında kitap görseli elde edildi.

B. ***Veri Eleme****:* Veri eleme aşamasında, çekilen görsellerde tekrarlayan görüntülerin tespiti, görüntü kalitesi bozuk olan görsellerin ayrıştırılması ve çıkarılması amaçlandı. Bu adımda Perceptual Hashing (phash) tekniği kullanılarak aynı veya benzer görseller belirlendi ve aynı görselden tek görsel kalacak şekilde silindi. Sonrasında manuel eleme ile görüntü kalitesi bozuk olan görseller ve kategoriyle uyumsuz görseller elenip silindi.

- ***Görsellerin Hash Değerlerini Hesaplama:*** Imagehash kütüphanesi kullanılarak her bir görselin hash değeri hesaplandı. İlk olarak, verilerin indirildiği klasördeki her dosya .jpg veya .png uzantısına göre filtrelendi. Görsel dosyası açıldıktan sonra, phash (algısal hash) yöntemiyle görselin hash değeri elde edildi.[3]

- ***Tekrarlayan Görselleri Ayırma:*** Görsellerin hash değerleri, daha önce hesaplanan hash değerleriyle karşılaştırılarak tekrarlayan görseller belirlendi. Eğer aynı hash değerine sahip başka bir görsel daha önce işlenmişse, bu görsel tekrarlayan olarak kabul

edilip dosya farklı bir belirtilen klasöre taşındı. Eğer hash değeri ilk kez hesaplanıyorsa, hash değeri ve dosya yolu hashes isimli bir sözlükte tutuldu. Böylece, aynı hash değerine sahip başka bir görselle karşılaşıldığında, bu görsel otomatik olarak belirtilen klasöre taşınarak ayıklandı. Kod tamamlandığında, tüm tekrarlayan görseller belirtilen klasöre taşınmış oldu.

Bu aşama sonucunda veri setindeki tekrarlayan görseller temizlenmiş ve veri çeşitliliği korunarak veri seti optimize edilmiş oldu.

C. ***Veri Çoklama****:* Veri çoklama işlemi, veri setini çeşitlendirmek ve sınıflandırma modelinin daha sağlam bir şekilde eğitilmesini sağlamak için uygulandı. Görseller üzerinde yapılan bu işlemler şunlardır:

- ***Parlaklık ve Kontrast Ayarları:*** Görsellerin parlaklık ve kontrast özelliklerini rastgele değiştirmek için adjust_brightness_contrast fonksiyonu kullanıldı. Bu fonksiyon, her görsele 0.5 ile 1.5 arasında rastgele bir parlaklık değeri ve aynı aralıkta bir kontrast değeri uyguladı.
- ***Döndürme İşlemi:*** Görseller sabit açılar olan 90, 180 ve 270 derece ile döndürülerek farklı açılardan analiz edilebilir hale getirildi. rotate_image fonksiyonu içinde yapılan bu döndürme işlemi sayesinde, aynı görsel farklı perspektiflerden incelenebilir duruma geldi.
- ***Rastgele Döndürme İşlemi:*** Her görsel, ek çeşitlilik sağlamak için 1 ile 360 derece arasında rastgele bir açıyla döndürüldü.

Bu işlemler sonucunda, sınıflandırma modelinin eğitimi için görsel veri seti farklı veri çoklama işlemleri ile çeşitlendirildi.

III. Sorun ve Çözümler

A. ***Veri Toplama Aşamasında Yaşanan Sorun ve Çözümler:***

- İlk yapılan web kazımada direkt olarak sayfa URL' lerinden görseller çekildi ancak bu görseller tek boyuta indirgendiğinde görsellerde ciddi piksel bozulmaları meydana geldi. Bu, projenin devamında sorun teşkil edeceğinden farklı bir web kazıma yoluna gidildi: Site URL'lerinde açılan görsellerin içeriğine tek tek yeni sekmede girilerek görseller indirildi.
- Siteye aynı istek çok fazla kez gönderildiği için site daha fazla veri çekmeye izin vermedi. Buna çözüm olarak verilerin URL'si text dosyasına satır satır kaydedilip ordan indirme işlemi yapıldı.
- İnternet bağlantısında kopmalar sebebiyle çok kez zaman aşımı hatalarıyla karşılaşıldı ve bu hatalar alındığında kodun tekrar tekrar en baştan çalıştırılmaması için text dosyasına kaydedilen URL'lerden hatanın alındığı sayfa tespiti yapıldı. Kodun çalışması hatanın alındığı sayfadan devam ettirildi.
- Veri çekilen sitelerden biri olan DR çok kez bakıma girdi. Bu durum kodun test sürecini yavaşlattı.
- Görselleri tek boyutta indirirken piksel bozulmaları yaşandı bu sorunu aşmak için çeşitli araştırmalar yapılarak LANCZOS interpolasyon yöntemi koda eklendi. Görseller 260x400 piksele yeniden boyutlandırılırken LANCZOS filtresi kullanıldı. Bu filtre, piksel detaylarını koruyup bozulmayı en aza indirdi.
- Bazı görseller PIL (Pillow) tarafından yeniden boyutlandırmaya uygun formatta değildi. Bu yüzden görseller RGB formatına dönüştürülerek boyutlandırmaya uygun hale getirildi.

B. ***Veri Eleme Aşamasında Yaşanan Sorunlar ve Çözümler:***

- Dataları manuel elerken çok fazla datanın tekrar ettiği görüldü. Bu yüzden imagehash kütüphanesi eklenerek veri tekrarı önlendi. Imagehash kütüphanesi, görsellerin içeriklerini sayısal bir biçimde temsil eden hash değerlerini hesaplamak için kullanılan bir klasördür. Bu hash değerleri, görselin temel özelliklerine dayanır ve görsellerin benzerliğini değerlendirmek için etkilidir. Eğer aynı hash değerine sahip bir görsel bulunursa, bu görsellerin içeriğinin benzer olduğunu ve dolayısıyla tekrarlanan görseller oldukları tespit edilir. Bu sayede, tekrarlanan görseller manuel olarak ayıklanmak yerine, otomatik olarak tanımlanıp ilgili klasöre taşındı.

C. ***Veri Çoklama Aşamasında Yaşanan Sorunlar ve Çözümler:***

- Rastgele derecelerde döndürme işlemlerinde karşılaşılan piksel bozulmalarını önlemek için Image.BICUBIC örnekleme yöntemi kullanıldı. Bu yöntem ile, görselde yapılan rastgele derecelerdeki döndürmelerde ortaya çıkabilecek pikselleşme veya bulanıklaşma gibi kalite kayıpları azaltılmış oldu.
- Görsellerin sabit boyutlarda indirilmesinden kaynaklı olarak resimlerde yapılan döndürmelerde boyutun aynı kalması durumunda döndürülmüş hallerini otomatik olarak boyuta göre kırpıyordu. (Örneğin normal şartlarda görsel 90 derece döndürüldüğünde boyutu 260*400 iken 400*260 olması gerekir ama bu işlem yapılmıyordu.) Bunun için kodda döndürme işleminin yapıldığı metoda expand=True parametresi eklendi. Bu parametre ile döndürme sonucu yapılan görseldeki kırpma işlemi önlendi.

IV. Sonuç

Bu çalışmada, kitap görsellerinden kategori tahmini yapmaya yönelik olarak veri toplama, veri eleme ve veri çoklama aşamaları gerçekleştirildi. Veri toplama işlemi ile kitap görselleri otomatik olarak toplandı; veri eleme işlemi ile tekrarlayan görseller çıkarıldı; veri çoklama işlemi ile veri seti çeşitlendirildi. Bu hazırlık adımları, sınıflandırma modelinin doğruluğunu artırmak için önemlidir ve gelecekteki çalışmalar için gereklidir.

V. KAYNAKÇA

Proje sürecinde yararlanılan kaynaklar ve web siteleri:

[1]https://www.udemy.com/course/python-ile-web-kazma-web-scraping-egitimi/

[2]https://stackoverflow.com/

[3]https://pypi.org/project/ImageHash/

[4]https://www.pandora.com.tr/

[5]https://www.dr.com.tr/

[6]https://www.datacamp.com/tutorial/complete-guide-data-augmentation