

Kocaeli Üniversitesi, Teknoloji Fakültesi Bilişim Sistemleri Mühendisliği Bölümü: Yazılım Geliştirme Laboratuvarı I

2024-2025 Güz Proje I Proje Raporu

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

Derya GELMEZ
Kocaeli Üniversitesi
Bilişim 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, bu görsellerin sınıflandırma amacıyla çoğaltılması ve 5 farklı yapay zeka modelinin (DeiT, BeiT, Swin Transformer, ViT, EfficientNet) Google Colab üzerinde eğitilerek sonuçlarının analizini ele almaktadır. Web kazıma işlemleri, görsel veri kümesi oluşturma amacıyla yapılmıştır. Veri eleme aşamasında tekrarlayan görseller için Hashing tekniği uygulanmış ve sonrasında manuel olarak eleme yapılmıştır. Veri çoğaltma aşamasında rastgele parlaklık-kontrast ayarı ve döndürme işlemleri gerçekleştirilmiştir. Sonrasında veriler üzerinde yüksek performans gösterebilecek modeller tespit edilip uygun görülen modellerin eğitilerek performans değerlendirilmesi yapılmıştır.

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

I. GİRİŞ

Görsel sınıflandırma problemlerinde modelin doğruluğunu artırmak için kaliteli ve çeşitlendirilmiş bir veri seti oldukça önemlidir. Bu çalışma, kitap kapaklarından kategori tahmini yapmak için beş temel aşama üzerinde yoğunlaşmaktadır: veri toplama, veri eleme, veri çoklama, yapay zeka modellerinin eğitimi ve sonuçlarının analiz edilmesi. Bu aşamalar, kitap görsellerinden kategori tahminine yönelik bir veri seti oluşturmak ve model performansını optimize etmek için yapılmıştır.

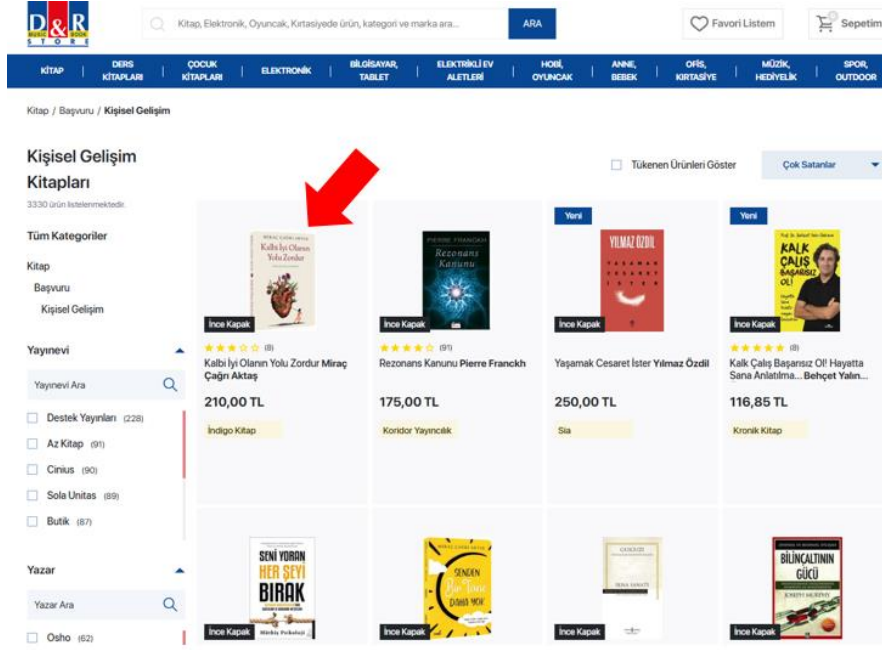
II. YÖNTEMLER

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.

- 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 çalıştırdıktan 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.



Resim 1: Web kazımada kullanılan görsellerden bir kesit

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.

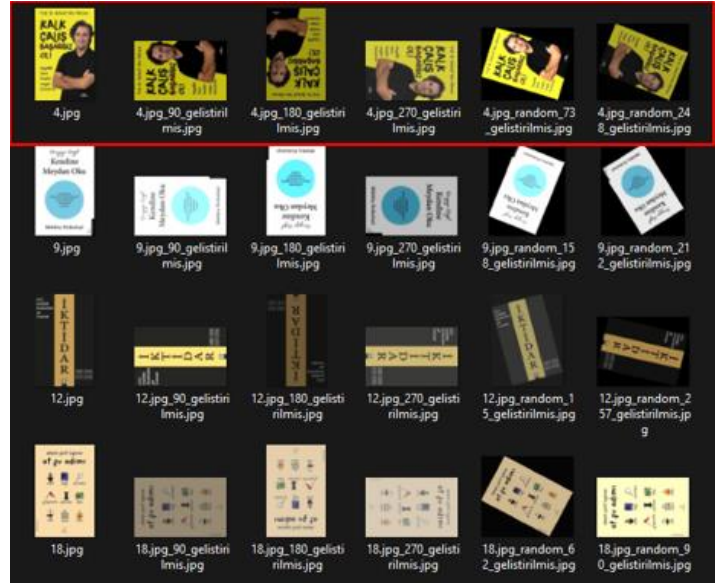
- Görsellerin Hash Değerlerini Hesaplama:** Imagehash kütüphanesi kullanılarak her bir görselin hash değeri hesaplandı. İlk olarak, verilerin indirildiği klasördeki her dosya .jpg veya .png uzantısına göre filtrelendi. Görsel dosyası açıldıktan sonra, phash (algısal hash) yöntemiyle görselin hash değeri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



Resim 2: Veri çoklama işlemi

D. Verilerin Üzerinde Transformatör Modellerinin

Uygulanması: Toplanan veriler yapay zeka modellerine uygun hale getirildi ve 5 farklı transformatör modeli belirlenerek eğitildi.

- **Verilerin Yapay Zeka Modeline Uygun Hale Getirilmesi:** Veriler, Google Drive üzerinden ZIP dosyası olarak alınmış ve zipfile kütüphanesi kullanılarak çıkarılmıştır. Çıkarılan veriler alt kategoriler halinde düzenlenmiş, gereksiz klasörler ayıklanmıştır. Veriler, %80 eğitim ve %20 test olarak train_test_split fonksiyonu ile ayrılmış, kategorilere göre ilgili klasörlere kopyalanmıştır. Görsellerin boyutları, modellerin giriş gereksinimlerine uygun olarak 260x400 piksel olacak şekilde yeniden boyutlandırılmış ve piksel değerleri ImageDataGenerator ile 0-1 arasına normalize edilmiştir. Eğitim ve test verilerinde, kategorilere eşit dağılım sağlanmış, veri akışı batch_size ve class_mode parametreleri ile düzenlenmiştir. Bu işlemler sonucunda, veri seti yapay zeka modellerine uygun hale getirilmiş ve eğitim/test süreçlerine hazır duruma getirilmiştir.
- **Transformatör Modellerinin Eğitilmesi:** DeiT, BEiT, Swin Transformer ve ViT transformer tabanlı olup PyTorch, EfficientNet ise CNN tabanlı olup TensorFlow kütüphanesi kullanılarak hazırlanmıştır. Aşırı

öğrenmeyi önlemek için her epoch sonrası performans değerlendirilmiş ve erken durdurma (early stopping) tekniği uygulanmıştır. Modellerin en iyi performansı sağlayan ağırlıkları kaydedilmiştir. Accuracy, Recall, Precision, Sensitivity, Specificity, F-Score, AUC değerleri hesaplanmış; karmaşıklık matrisleri ve ROC eğrileri oluşturulmuştur. Epoch ve Loss grafikleri, eğitim ve test kümeleri için analiz edilmiştir. Modellerin eğitim süreleri (training time) ve çıkarım süreleri (inference time) hesaplanarak performansları karşılaştırılmıştır. Bu süreçte en iyi sonuç veren modeller belirlenmiştir.

E. Model Sonuçlarının Analizi: Her model için Accuracy, Recall, Precision, Sensitivity, Specificity, F-Score ve AUC değerleri hesaplanarak karmaşıklık matrisleri ve ROC eğrileri çıkartıldı. Her modelin eğitim ve test veri kümeleri için epoch size ve loss değerleri aynı grafik üzerinde gösterilerek eğitim zamanları (training time) ile çıkarım zamanları (inference time) çıktı olarak verildi. Tüm modeller için değerler ve grafikler karşılaştırılarak sonuçları analiz edildi.

- **DeiT:** Görsel sınıflandırma görevinde başarılı bir performans göstermiştir. Eğitim süreci boyunca aşağıdaki sonuçlar elde edilmiştir:

a. *Performans Metrikleri:*

	Epoch 7/20	Loss 30.7665	Accuracy 0.9934	
	Validation Accuracy	0.9628		
	Precision	0.9632		
	Recall	0.9628		
	F-Score	0.9627		
	Sensitivity	0.9544		
	Specificity	0.9929		
Classification Report				
	Precision	Recall	F1-Score	Support
0	0.93	0.97	0.95	1300
1	0.98	0.93	0.96	1300
2	0.97	0.94	0.96	1300
3	0.97	0.99	0.98	1300
4	0.97	0.98	0.97	1300
Accuracy			0.96	6500
Macro Avg	0.96	0.96	0.96	6500
Weighted Avg	0.96	0.96	0.96	6500

Tablo 1: DeiT Analiz Sonuçları

Doğruluk (Accuracy): %96.8'e kadar çıkmıştır, bu da modelin yüksek sınıflandırma performansını göstermektedir.

Precision: %96.7 seviyesine ulaşarak, modelin yanlış pozitifleri azaltmada etkili olduğunu ortaya koymuştur.

Recall: %96.6 oranıyla, modelin doğru sınıfları tespit etme yeteneği oldukça iyidir. F1-Score: %96.7 ile dengeli bir performans sergilemiştir.

Sensitivity (Duyarlılık): %98.6 gibi yüksek bir değer, modelin pozitif sınıfları tespit etme yeteneğini göstermektedir.

Specificity (Özgüllük): %99.1 ile modelin negatif sınıfları doğru şekilde dışlama yeteneği güçlüdür.

AUC (Area Under the Curve): Modelin ROC eğrisi altında kalan alan değeri yüksek bulunmuş ve ayırma başarısını kanıtlamıştır.

b. Eğitim Süreci:

Model, toplamda 7 epoch boyunca eğitilmiştir. Early stopping tekniği kullanılarak aşırı öğrenme (overfitting) önlenmiştir.

Eğitim süresi: 2151.48 saniye

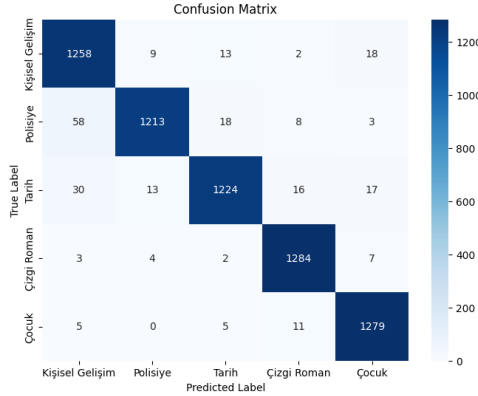
Çıkarım süresi (inference time): 23.05 saniye

c. Analiz ve Sonuçlar:

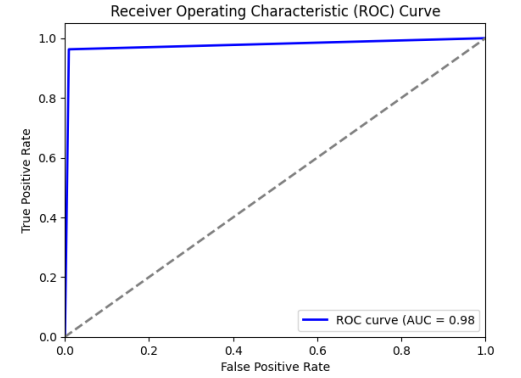
Karmaşıklık Matrisi: Her sınıf için doğru ve yanlış tahminler analiz edilmiştir. Tüm sınıflar arasında dengeli bir performans sergilemiştir.

ROC Eğrisi: Modelin hem doğruluk hem de ayırma yeteneği güçlü bulunmuştur.

Eğitim ve test veri kümelerinde, epoch sayısı ile kayıp (loss) değerleri aynı grafik üzerinde gösterilerek eğitim sürecindeki iyileşme gözlemlenmiştir.



Resim 3: DeiT Karmaşıklık Matrisi



Resim 4: DeiT ROC Eğrisi



Resim 5: DeiT Epochs ve Loss Grafiği

- **BEiT:** Görsel sınıflandırma görevinde etkili bir performans sergilemiştir. Eğitim süreci ve sonuçlar şu şekildedir:

a. *Performans Metrikleri:*

	Epoch 11/20	Loss 37.5898	Accuracy 0.9919	
	Validation Accuracy	0.9608		
	Precision	0.9609		
	Recall	0.9608		
	F-Score	0.9608		
	Sensitivity	0.9718		
	Specificity	0.9794		
Classification Report				
	Precision	Recall	F1-Score	Support
0	0.95	0.95	0.95	1300
1	0.94	0.96	0.95	1300
2	0.96	0.96	0.96	1300
3	0.99	0.97	0.98	1300
4	0.97	0.97	0.97	1300
Accuracy			0.96	6500
Macro Avg	0.96	0.96	0.96	6500
Weighted Avg	0.96	0.96	0.96	6500

Tablo 2: BEiT Analiz Sonuçları

Doğruluk (Accuracy): %97.0'a kadar çıkmış, modelin yüksek sınıflandırma başarısını göstermiştir.

Precision: %97.0 ile doğru pozitif tahminlerde güçlü bir performans sergilemiştir.

Recall: %97.0, modelin doğru sınıfları tespit etme kapasitesinin yüksek olduğunu ortaya koymaktadır.

F1-Score: %97.0, modelin genel dengesini ve sınıflandırmadaki etkinliğini kanıtlamıştır.

Sensitivity (Duyarlılık): %99.0, modelin pozitif sınıfları yüksek doğrulukla tespit ettiğini göstermektedir.

Specificity (Özgüllük): %98.6 ile negatif sınıfları doğru bir şekilde dışlamaktadır.

b. *Eğitim Süreci:*

Model toplamda 11 epoch boyunca eğitilmiş ve early stopping tekniği ile aşırı öğrenme (overfitting) önlenmiştir.

Eğitim süresi: 3785.92 saniye

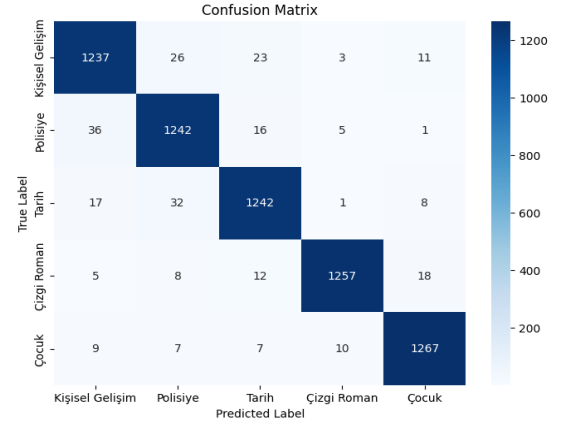
Çıkarım süresi (inference time): 37.42 saniye

c. *Analiz ve Sonuçlar:*

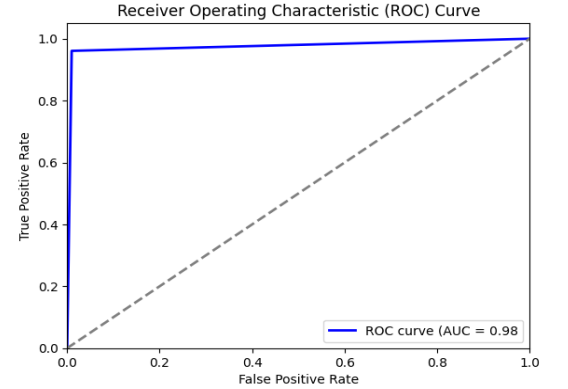
Karmaşıklık Matrisi: Model, tüm sınıflar arasında dengeli bir performans göstermiştir.

ROC Eğrisi: Yüksek AUC değeri, modelin ayırma kapasitesinin güçlü olduğunu göstermektedir.

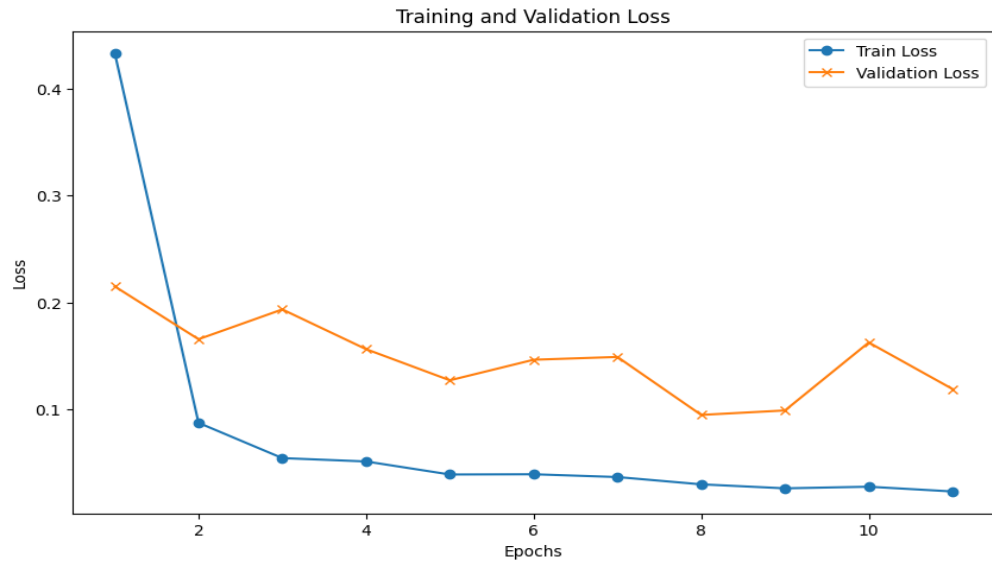
Epoch ve Loss Grafiği: Eğitim ve doğrulama kayıpları incelenmiş ve modelin tutarlı bir iyileşme gösterdiği görülmüştür.



Resim 6: BEiT Karmaşıklık Matrisi



Resim 7: BEiT ROC Eğrisi



Resim 8: BEiT Epochs ve Loss Grafiği

- **EfficientNet:** Görsel sınıflandırma görevinde güçlü ve verimli bir performans sergilemiştir. Eğitim süreci ve sonuçlar şu şekildedir:
 - Performans Metrikleri:

	Epoch 9/20	Loss 33.4236	Accuracy 0.9927	
	Validation Accuracy	0.9595		
	Precision	0.9601		
	Recall	0.9595		
	F-Score	0.9594		
	Sensitivity	0.9789		
	Specificity	0.9851		
Classification Report				
	Precision	Recall	F1-Score	Support
0	0.94	0.96	0.95	1300
1	0.94	0.97	0.95	1300
2	0.99	0.91	0.94	1300
3	0.97	0.99	0.98	1300
4	0.96	0.97	0.97	1300
Accuracy			0.96	6500
Macro Avg	0.96	0.96	0.96	6500
Weighted Avg	0.96	0.96	0.96	6500

Tablo 3: EfficientNet Analiz Sonuçları

Doğruluk (Accuracy): %96.0'a ulaşarak modelin yüksek sınıflandırma başarısını göstermiştir.

Precision: %96.0, modelin doğru pozitif tahminlerde etkili olduğunu ortaya koymuştur.

Recall: %96.0, modelin doğru sınıfları tespit etme kapasitesini yansıtmaktadır.

F1-Score: %96.0, modelin genel dengesini ve sınıflandırmadaki etkinliğini kanıtlamıştır.

Sensitivity (Duyarlılık): %97.9, pozitif sınıfları doğru bir şekilde tespit etme başarısını göstermektedir.

Specificity (Özgüllük): %98.5, negatif sınıfları doğru bir şekilde dışlama başarısını kanıtlamaktadır.

b. **Eğitim Süreci:**

Model toplamda 9 epoch boyunca eğitilmiş ve early stopping tekniği kullanılarak aşırı öğrenme (overfitting) önlenmiştir.

Eğitim süresi: 1411.54 saniye

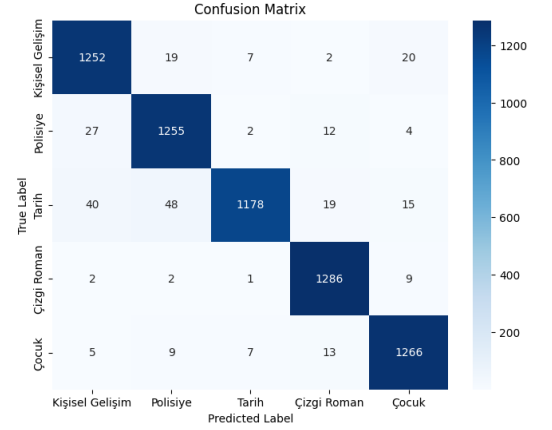
Çıkarım süresi (inference time): 23.62 saniye

c. **Analiz ve Sonuçlar:**

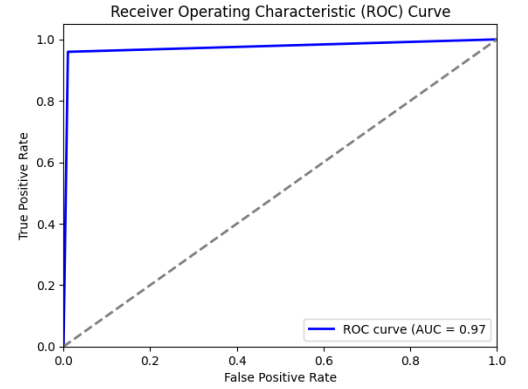
Karmaşıklık Matrisi: Model, tüm sınıflar arasında dengeli bir performans göstermiştir.

ROC Eğrisi: Yüksek AUC değeri, modelin sınıflandırma başarısını pekiştirmiştir.

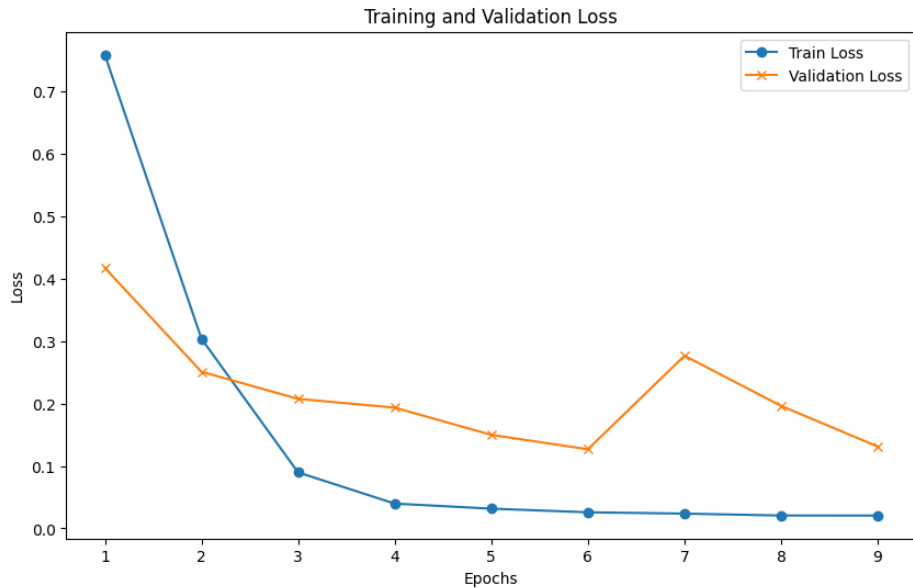
Epoch ve Loss Grafiği: Eğitim ve doğrulama kayıpları analiz edilerek modelin tutarlı bir iyileşme sergilediği gözlemlenmiştir.



Resim 9: EfficientNet Karmaşıklık Matrisi



Resim 10: EfficientNet ROC Eğrisi



Resim 11: EfficientNet Epochs ve Loss Grafiği

- **ViT:** Görsel sınıflandırma görevinde etkili bir performans sergilemiştir. Eğitim süreci ve sonuçlar şu şekildedir:

a. *Performans Metrikleri:*

	Epoch 5/20	Loss 40.5328	Accuracy 0.9920	
	Validation Accuracy		0.9635	
	Precision		0.9641	
	Recall		0.9635	
	F-Score		0.9637	
	Sensitivity		0.9898	
	Specificity		0.9852	
Classification Report				
	Precision	Recall	F1-Score	Support
0	0.93	0.97	0.95	1300
1	0.97	0.97	0.97	1300
2	0.95	0.96	0.95	1300
3	1	0.97	0.98	1300
4	0.98	0.95	0.96	1300
Accuracy			0.96	6500
Macro Avg	0.96	0.96	0.96	6500
Weighted Avg	0.96	0.96	0.96	6500

Tablo 4: ViT Analiz Sonuçları

Doğruluk (Accuracy): %96.35 ile yüksek bir başarı elde edilmiştir.

Precision: %96.41, doğru pozitif tahminlerde etkili olduğunu göstermektedir.

Recall: %96.35, doğru sınıfları tespit etme yeteneğini yansıtmaktadır.

F1-Score: %96.37, modelin genel dengesini ve sınıflandırmadaki başarısını göstermektedir.

Sensitivity (Duyarlılık): %98.98, pozitif sınıfları tespit etmedeki yüksek başarıyı ortaya koymaktadır.

Specificity (Özgüllük): %98.52, negatif sınıfları doğru bir şekilde dışlama yeteneğini göstermektedir.

b. *Eğitim Süreci:*

Model toplamda 5 epoch boyunca eğitilmiş ve early stopping tekniği ile aşırı öğrenme (overfitting) önlenmiştir.

Eğitim süresi: 1525.14 saniye

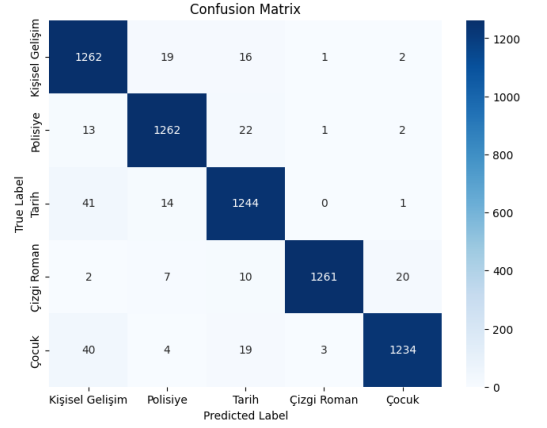
Çıkarım süresi (inference time): 22.81 saniye

c. *Analiz ve Sonuçlar:*

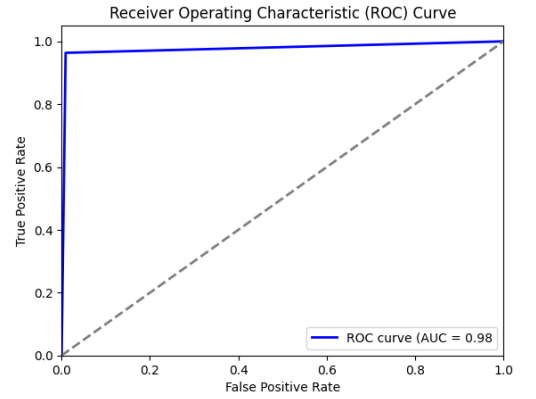
Karmaşıklık Matrisi: Model, tüm sınıflar arasında dengeli bir performans sergilemiştir.

ROC Eğrisi: Yüksek AUC değeri, modelin ayırma kapasitesinin güçlü olduğunu göstermektedir.

Epoch ve Loss Grafiği: Eğitim ve doğrulama kayıpları incelenerek modelin tutarlı bir iyileşme sergilediği gözlemlenmiştir.



Resim 12: ViT Karmaşıklık Matrisi



Resim 13: ViT ROC Eğrisi



Resim 14: ViT Epochs ve Loss Grafiği

- **Swin Transformer:** Görsel sınıflandırma görevinde etkili bir performans sergilemiştir. Eğitim süreci ve sonuçlar şu şekildedir:
 - Performans Metrikleri:**

	Epoch 10/20	Loss 26.5406	Accuracy 0.9950	
	Validation Accuracy		0.9854	
	Precision		0.9854	
	Recall		0.9854	
	F-Score		0.9854	
	Sensitivity		0.9891	
	Specificity		0.993	
Classification Report				
	Precision	Recall	F1-Score	Support
0	0.98	0.99	0.98	1300
1	0.98	0.97	0.98	1300
2	0.98	0.98	0.98	1300
3	1	0.99	0.99	1300
4	0.99	0.99	0.99	1300
Accuracy			0.99	6500
Macro Avg	0.99	0.99	0.99	6500
Weighted Avg	0.99	0.99	0.99	6500

Tablo 5: Swin Transformer Analiz Sonuçları

Doğruluk (Accuracy): %98.54 ile oldukça yüksek bir başarı elde edilmiştir.

Precision: %98.54, modelin doğru pozitif tahminlerde etkili olduğunu göstermektedir.

Recall: %98.54, modelin doğru sınıfları tespit etme kapasitesini yansıtmaktadır.

F1-Score: %98.54, modelin genel dengesini ve sınıflandırmadaki başarısını göstermektedir.

Sensitivity (Duyarlılık): %98.91, pozitif sınıfları tespit etmedeki yüksek başarıyı ortaya koymaktadır.

Specificity (Özgüllük): %99.30, negatif sınıfları doğru bir şekilde dışlama yeteneğini göstermektedir.

- Eğitim Süreci:**

Model toplamda 10 epoch boyunca eğitilmiş ve early stopping tekniği ile aşırı öğrenme (overfitting) önlenmiştir.

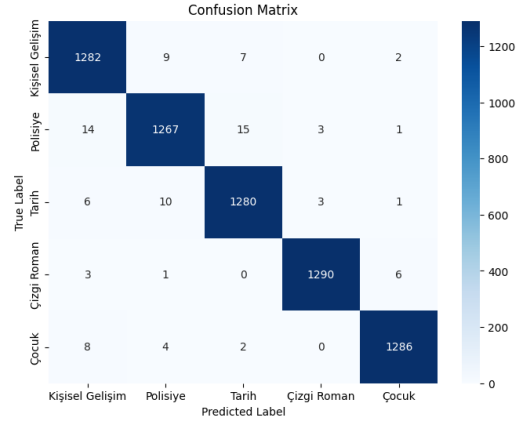
Eğitim süresi: 3580.38 saniye

Çıkarım süresi (inference time): 33.57 saniye

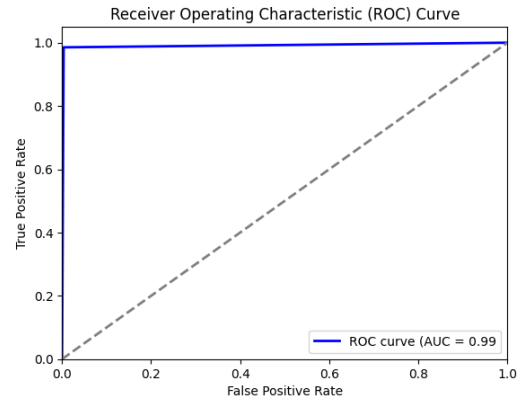
- Analiz ve Sonuçlar:**

Karmaşıklık Matrisi: Model, tüm sınıflar arasında dengeli bir performans sergilemiştir. ROC Eğrisi: Yüksek AUC değeri, modelin ayırma kapasitesinin güçlü olduğunu göstermektedir.

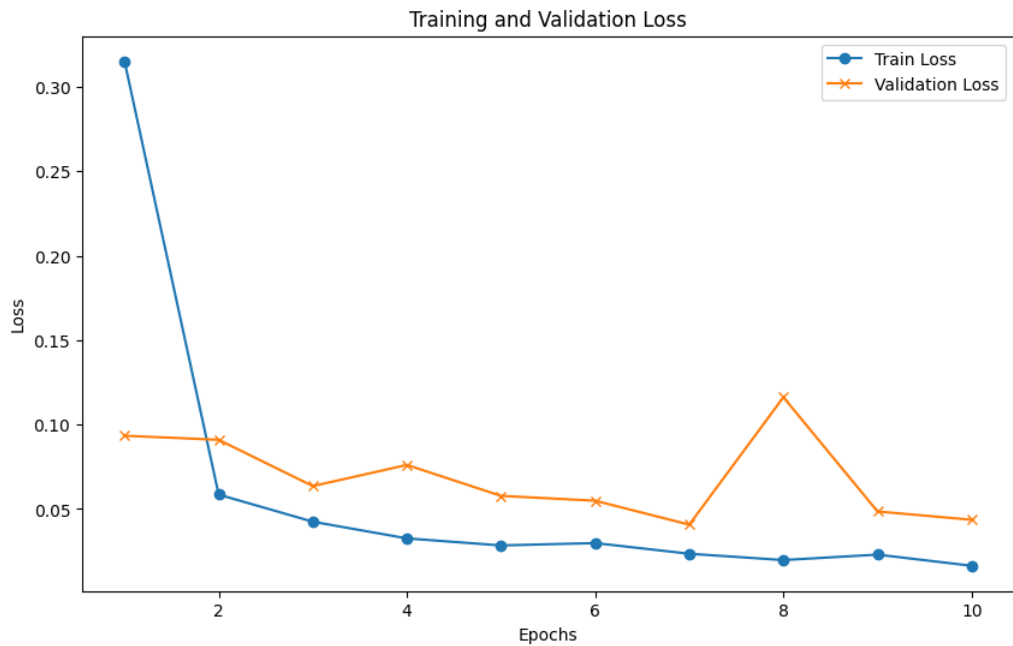
Epoch ve Loss Grafiği: Eğitim ve doğrulama kayıpları analiz edilerek modelin tutarlı bir iyileşme sergilediği gözlemlenmiştir.



Resim 15: Swin Transformer Karmaşıklık Matrisi



Resim 16: Swin Transformer ROC Eğrisi



Resim 17: Swin Transformer Epochs ve Loss Grafiği

III. SORUN VE ÇÖZÜMLER

A. Veri Toplama Aşamasında Yaşanan Sorun ve Çözümler:

- İlk yapılan web kazımda direkt olarak sayfa URL'lerinden görseller çekildi ancak bu görseller tek boyuta indirildiğinde 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 önendi. 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 kırpyordu. (Örneğin normal şartlarda görsel 90 derece döndürüldüğünde boyutu 260*400 iken 400*260 olması gerekir ama bu işlem yapılmıyordu.) Bunun için koda döndürme işleminin yapıldığı metoda expand=True parametresi eklendi. Bu parametre ile döndürme sonucu yapılan görseldeki kırpma işlemi önendi.

D. Yapay Zeka Modellerinin Eğitilmesi ve Sonuçlarının Analizinde Karşılaşılan Sorunlar ve Çözümler:

- Google Colab'e veriler yüklenirken bunun çok uzun sürmesinden kaynaklı olarak Google Drive üzerinden bu işlemin yapılmasına karar verildi.
- Veriler yapay zeka modellerinin eğitimine uygun hale getirilirken aşırı öğrenmeyi önlemek adına her kategorideki veri sayısı eşit hale getirildi.
- Google Colab'teki çalışma zamanı sınırlaması ve GPU kullanımındaki kısıtlama sebebiyle ücretli versiyonuna geçildi. Böylece modellerin daha hızlı eğitilmesi sağlandı.
- Model eğitimine ResNet ve ardından MobileNet ile başlandı. Ancak kötü performansla sonuçlanmaları üzerine yapılan iyileştirmelere rağmen iyi bir sonuç alınamadı. Bu sebeple başka model araştırmasına girilerek daha iyi performans alınabilecek modellerin kullanımına karar verildi.
- Başlangıçta modellere epoch değerleri doğrudan atandı bu da bazı modellerde aşırı öğrenmeye sebep olarak performansı düşürdü. Bunu önlemek için erken durdurma (early stopping) tekniği uygulandı.

IV. SONUÇ

Bu çalışmada, kitap görsellerinden kategori tahmini yapmaya yönelik olarak kapsamlı bir veri hazırlama ve model eğitimi süreci gerçekleştirilmiştir. Veri toplama aşamasında görseller otomatik yöntemlerle toplanmış, tekrarlayan ve düşük kaliteli görsellerin elenmesiyle veri setinin kalitesi artırılmıştır. Veri çoklama işlemiyle sınıflandırma modellerinin daha sağlam ve genelleme yeteneği yüksek bir şekilde eğitilmesi sağlanmıştır.

Eğitim sürecinde DeiT, BEiT, Swin Transformer, ViT, ve EfficientNet modelleri kullanılarak görsel sınıflandırma performansları analiz edilmiştir. Tüm modellerin doğruluk, duyarlılık, özgüllük ve F1-skora dayalı olarak güçlü sonuçlar verdiği ancak Swin Transformer ve ViT modellerinin diğerlerine kıyasla daha yüksek genelleme performansı sergilediği gözlemlenmiştir.

Elde edilen sonuçlar, veri hazırlık sürecinin ve model seçiminin sınıflandırma başarısını artırmada kritik öneme sahip olduğunu göstermektedir. Bu çalışma, kitap kapakları gibi görsel içeriklerin sınıflandırılmasına yönelik önemli bir katkı sunmuş ve gelecekteki çalışmalara ışık tutabilecek bir temel oluşturmuştur. Gelecekte veri setinin genişletilmesi ve

farklı model mimarilerinin incelenmesi bu alandaki başarının daha da artırılmasına olanak sağlayabilir.

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>

[7]P. Kanani and M. Padole, "Deep Learning to Detect Skin Cancer using Google Colab," *International Journal of Engineering and Advanced Technology (IJEAT)*, vol. 8, no. 6, pp. 2176–2183, Aug. 2019. DOI: 10.35940/ijeat.F8587.088619.