

YOUTUBE VIDEO KAPAK RESMİ SINIFLANDIRMA

Yazılım Geliştirme Laboratuvarı güz dönemi projesi

Yusuf Can Demirkol
Kocaeli Üniversitesi
Bilişim Sistemleri Mühendisliği
221307061
josephacn86@gmail.com

Muhammed Abdullah Acar
Kocaeli Üniversitesi
Bilişim Sistemleri Mühendisliği
221307038
acar167257@gmail.com

Bu proje youtube video kapak resimlerinin web kazıma yapılarak elde edilen dataset' in beş farklı transformatör modelinin eğitilmesinde kullanılarak modellerin performanslarının karşılaştırılmasından oluşur. Raporu sırasıyla; web kazıma işlemleri, veri işlemleri, transformatör modellerinin özellikleri, modellerin eğitilmesi, modellerin performans analizleri ve karşılaştırmaları yer almaktadır.

I. WEB KAZIMA İŞLEMLERİ

Projede kullanılan veriler python ve selenium kullanarak oluşturulan web kazıma kodu aracılığıyla youtube platformu üzerinden toplanmıştır. Projede toplamda beş sınıf (eğitim, oyun, spor, belgesel, yemek) her sınıftan beş bin veri, toplamda yirmi beş bin veri kullanılmıştır. Web Kazıma kodları raporun sonunda verilen github linkinde mevcuttur.

II. VERİ İŞLEMLERİ

A. Veri Ayıklama

Web kazıma sonucu elde edilen veriler projede kullanıma uygun hale getirilmek için ayıklanmıştır. Bu aşamada temel amaç modellerin verileri iyi analiz edip öğrenebilmesidir.

Veri ayıklamanın kendi içinde iki aşaması vardır. İlk aşama, verilerin kazınırken ayıklanması; İkinci aşama ise kazınan verilerin elle ayıklanıp son halini almasıdır.

B. Veri Arttırımı ve normalizasyon işlemleri

Ayıklanan verilerin sayısı, projede her sınıf başına minimum sınır olan beş bin veri sayısından az olduğu için verilere veri arttırımı uygulandı. Bu işlemin amacı hem proje isterlerini karşılamak hem de modellerin eğitim ve tahmin güçlerini arttırmaktır.

Veriler 224*224 boyutuna getirilerek modellerin veriyi işlemesi kolaylaştırıldı. Böylece model eğitim sürelerinden tasarruf edildi.

III. TRANSFORMATÖR MODELLERİ VE ÖZELLİKLERİ

Projede toplam beş farklı transformatör modeli kullanıldı. Bu modeller önceden eğitilmiş modellerin mimarisini kullanarak dataset üzerinde tam performans göstermesi için oluşturulmuş public birimlerdir.

Modeller; Vit, Beit, Deit, Swin, ConvNext

A. Vit (Vision Transformer)

a) 1. Tanımı

Vision Transformer (ViT), doğal dil işleme için geliştirilen Transformer mimarisini bilgisayarla görme alanına adapte eden bir derin öğrenme modelidir. ViT, görüntüleri analiz etmek için evrimsel sinir ağları (CNN) yerine Transformer yapısını kullanır. Bu yaklaşım, görsel veriler üzerindeki klasik yöntemlerden farklıdır ve birçok görevde performans avantajı sağlayabilir.

b) 2. Çalışma Prensipleri

ViT'nin temel yaklaşımı, görüntüyü küçük sabit boyutlu parçalara (patch) ayırmak ve bu parçaları bir dizi gibi ele almaktır.

Adımlar:

1. Görüntülerin Patch'lere Ayrılması: Girdi görüntüsü, sabit boyutlu küçük parçalara (ör. 16x16 piksel) bölünür.
2. Lineer Gösterimler: Her bir patch, sabit boyutlu bir vektöre dönüştürülür (lineer gömme).
3. Konum Kodlaması (Positional Encoding): Görsel dizinin sırasını ve yapısını korumak için her bir patch'e konum bilgisi eklenir.
4. Transformer Mimarisi: Transformer'ın özelleştirilmiş encoder katmanları kullanılarak bu dizi işlenir.
5. Sınıflandırma Kafası: Görüntü sınıflandırma veya diğer görsel görevler için nihai bir sınıflandırma bloğu eklenir.

c) 5. Uygulama Alanları

ViT, özellikle aşağıdaki bilgisayarla görme görevlerinde başarı göstermektedir:

- Görüntü sınıflandırma
- Nesne algılama
- Segmentasyon

- Anomali tespiti
- Derin sahne analizi

B. BeIT (BERT Pre-trained Vision Transformer)

a) 1. Tanımı

BeIT (BERT Pre-trained Image Transformer), Vision Transformer (ViT) modelinin görüntü işleme görevlerine yönelik özelleştirilmiş bir türevidir. BeIT, dil işleme görevlerinde başarılı olan BERT modelinin maskeli veri modelleme (Masked Data Modeling) yöntemini görsel veriler için uyarlamaktadır. Bu model, özellikle büyük veri setleri üzerinde kendini denetleyen (self-supervised) öğrenme yaklaşımıyla önceden eğitilir.

b) 2. Çalışma Prensipleri

BeIT'nin temel yeniliklerinden biri, görüntüler üzerinde maskeli görsel modelleme (Masked Image Modeling) kullanmasıdır. Bu, dil işleme görevlerindeki BERT modeliyle benzerlik gösterir.

Adımlar:

1. Girdi Görüntüsünün Patch'lere Bölünmesi:
 - Görüntü, sabit boyutlu patch'lere ayrılır (ör. 16x16 piksel).
2. Masked Image Modeling (MIM):
 - Patch'lerin belirli bir kısmı maskelenir (örneğin, rastgele %30'u).
 - Model, maskelenmiş bölgeleri çevresel bilgilere dayanarak tahmin etmeye çalışır.
3. Pre-Training (Ön Eğitim):
 - Büyük ölçekli görsel veri setlerinde kendini denetleyen bir şekilde eğitilir.
4. Fine-Tuning (İnce Ayar):
 - Sınıflandırma, segmentasyon veya nesne algılama gibi özel görevler için model özelleştirilir.

c) 5. Uygulama Alanları

BeIT, aşağıdaki bilgisayarla görme görevlerinde kullanılmaktadır:

- Görüntü Sınıflandırma: Görselleri önceden tanımlı kategorilere ayırma.
- Segmentasyon: Görüntülerdeki nesneleri veya bölgeleri piksellerine ayırma.
- Nesne Algılama: Görsellerdeki nesneleri tanımlama ve konumlarını belirleme.
- Anomali Tespiti: Standart olmayan görsel özelliklerin belirlenmesi.

C. DeiT (Data-efficient Image Transformer).

a) 1. Tanımı

DeiT (Data-efficient Image Transformer), Vision Transformer (ViT) mimarisini daha az veriyle etkili bir şekilde eğitmek için geliştirilmiş bir modeldir. DeiT, veri verimliliği sorununa odaklanır ve özellikle büyük veri kümelerine erişimi olmayan kullanıcılar için tasarlanmıştır. Bu model, Transformer tabanlı görüntü sınıflandırma modellerini veri verimliliğini artıracak şekilde optimize eder.

b) 2. Çalışma Prensipleri

DeiT'nin temel özellikleri, Transformer mimarisini geliştirmek ve eğitim sürecini daha etkili hale getirmek için optimize edilmiş birkaç yöntem içerir:

- Knowledge Distillation (Bilgi Aktarımı):
 - DeiT, bir "öğrenci" modelin bir "öğretmen" modelden öğrenmesini sağlar.
 - Öğretmen model, genellikle bir CNN (Convolutional Neural Network) tabanlıdır ve daha fazla veriyle önceden eğitilmiş bir modeldir.
 - Görüntü sınıflandırmada, hem gerçek etiketler hem de öğretmen modelin sağladığı bilgi kullanılır.
- Eğitim Süreci:
 - DeiT, standart Transformer mimarisiyle çalışır, ancak eğitim sırasında ek veri artırma (data augmentation) ve düzenleme (regularization) yöntemleri kullanılır.
 - Görüntüleri küçük yama (patch) dizilerine bölerek işlem yapar ve sınıflandırma için Transformer encoder katmanlarını kullanır.
- Distillation Token:
 - DeiT, standart Transformer tokenlarının yanı sıra bir distillation token kullanır. Bu token, modelin öğretmen modelden bilgi öğrenmesini sağlar.

c) 5. Uygulama Alanları

DeiT, aşağıdaki bilgisayarla görme görevlerinde etkili bir şekilde kullanılabilir:

- Görüntü Sınıflandırma:
 - Görüntülerin belirli sınıflara atanması.
- Transfer Öğrenimi:
 - Önceden eğitilmiş bir model olarak farklı görsel görevlerde yeniden kullanılması.

- Küçük Veri Kümeleriyle Eğitim:
 - Veri setinin sınırlı olduğu durumlarda güçlü bir seçenek sunar.

D. Swin Transformer Modeli

a) 1. Tanımı

Swin Transformer (Shifted Window Transformer), bilgisayarla görme görevlerinde yüksek performans sağlayan, Vision Transformer (ViT) modeline dayalı bir derin öğrenme mimarisidir. Swin Transformer, görsel verilerde yerel ve küresel bağlam bilgisini etkili bir şekilde modellemek için **kaydırılmış pencere (shifted window)** mekanizmasını ve hiyerarşik bir yapıyı benimser. Bu özellikleriyle Swin Transformer, hem sınıflandırma hem de nesne algılama gibi çeşitli görsel görevlerde üstün performans sunar.

b) 2. Çalışma Prensipleri

Swin Transformer, geleneksel ViT mimarisindeki hesaplama maliyetlerini ve ölçeklenebilirlik sorunlarını çözmek için geliştirilmiştir.

- Kaydırılmış Pencere (Shifted Window) Mekanizması:
 - Görüntüler sabit boyutlu pencerelere bölünür ve her pencere yerel bağlam bilgisi çıkarır.
 - Pencereler arasında bilgi akışını sağlamak için her katmanda pencereler kaydırılır. Bu yaklaşım, küresel bağlamın etkili bir şekilde modellenmesine olanak tanır.
- Hiyerarşik Yapı:
 - Swin Transformer, farklı çözünürlüklerdeki görsel özellikleri öğrenmek için çok ölçekli bir mimariye sahiptir. Daha derin katmanlarda özellik haritalarının boyutu küçültülür ve bilgi yoğunluğu artırılır.
- Karmaşıklığın Azaltılması:
 - Transformer'ın hesaplama maliyetini azaltmak için yerel pencere başına kendine dikkat (self-attention) uygulanır.

c) 5. Uygulama Alanları

Swin Transformer, bilgisayarla görme alanında aşağıdaki görevlerde geniş bir uygulama yelpazesine sahiptir:

1. Görüntü Sınıflandırma:
 - Görselleri belirli kategorilere ayırma.

2. Nesne Algılama:
 - Görsellerdeki nesneleri tanımlama ve konumlandırma.
3. Görüntü Segmentasyonu:
 - Görsellerdeki nesneleri piksel düzeyinde ayırma.
4. Video İşleme:
 - Zaman boyutundaki görsel analiz için uyarlanabilir.
5. Tıbbi Görüntü Analizi:
 - Anomali tespiti ve özellik çıkartımı.

E. ConvNeXt Modeli

a) 1. Tanımı

ConvNeXt, geleneksel **Convolutional Neural Networks (CNN)** ve modern **Vision Transformer (ViT)** modellerinin en iyi özelliklerini birleştiren bir derin öğrenme mimarisidir. ConvNeXt, klasik evrimsel sinir ağlarını modernize ederek ViT modellerinin başardığı yüksek performansı yakalamayı amaçlar. Bu model, evrim tabanlı yöntemlerin hala görsel görevler için geçerli bir çözüm olduğunu göstermek üzere geliştirilmiştir.

b) 2. Çalışma Prensipleri

ConvNeXt, CNN'lerin temel yapısını korurken modern Transformer modellerinden ilham alarak çeşitli optimizasyonlar yapar:

- Modernizasyon:
 - ConvNeXt, ResNet gibi geleneksel CNN modellerini alır ve bunları daha modern bir yapıya dönüştürür.
 - Layer Normalization (LN), GELU aktivasyon fonksiyonu ve daha büyük kernel boyutları gibi özellikler ekler.
- Büyük Kernel Boyutları:
 - Daha büyük evrim çekirdekleri (ör. 7x7) kullanarak, modelin daha geniş bir algılama alanına sahip olmasını sağlar. Bu, Transformer modellerindeki global self-attention mekanizmasına benzer bir etki yaratır.
- Hiyerarşik Yapı:
 - ConvNeXt, ViT modellerine benzer şekilde çok ölçekli özellik haritaları üretir. Böylece model, hem düşük hem de yüksek seviyeli görsel bilgileri öğrenebilir.
- Optimizasyon Teknikleri:
 - Performansı artırmak için derin ağ tasarımı (deep architecture), ağ başına azalan doğrusal karmaşıklık ve daha iyi veri artırma stratejileri benimsenir.

c) 5. Uygulama Alanları

ConvNeXt, aşağıdaki görevlerde kullanılabilir:

1. Görüntü Sınıflandırma:
 - o Görselleri önceden belirlenmiş kategorilere ayırma.
2. Nesne Algılama:
 - o Görsellerdeki nesnelerin konumunu belirleme ve sınıflandırma.
3. Görüntü Segmentasyonu:
 - o Görüntülerin piksel seviyesinde analiz edilmesi.
4. Medikal Görüntü İşleme:
 - o Özellikle medikal veri setlerinde yüksek performans sağlar.
5. Video Analizi:
 - o Zaman boyutundaki görsel değişiklikleri işlemek için uyarlanabilir.

IV. MODELLERİN EĞİTİLMESİ

Modeller google colab ortamında eğitildi ve performans analizleri yapıldı. Eğitilirken gereken her kod parçası farklı kod hücrelerinde çalıştırılarak olası hata durumlarında hatanın çözülmesi kolaylaştırıldı. Ayrıca projenin okunabilirliği artırıldı. Aşağıda modeller eğitilirken

a) VIT Modelinin eğitilmesi:

- 1) Kullanılacak kütüphaneler import edildi.
- 2) PyTorch'un Dataset sınıfını özelleştirerek bir görüntü veri kümesi oluşturuldu.
- 3) Google Colab ortamında Google Drive'da depolanan sıkıştırılmış ZIP dosyalarını çözümlendi (extract etmek).
- 4) PyTorch kullanılarak görüntü veri kümesi eğitim ve test için hazırlandı.
- 5) Vision Transformer (ViT) modeli kullanılarak bir görüntü sınıflandırma modeli hazırlandı.
- 6) Vision Transformer modelini eğitmek için gereken kaybı hesaplama ve optimizasyon işlemleri tanımlandı.
- 7) Vision Transformer (ViT) modelini eğitecek ve eğitirken eğitim kaybı, doğrulama kaybı, eğitim süresi gibi parametreleri hesaplayan kod hazırlandı.
Bu kod ayrıca her Epoch sonunda valid, tarining loss değerlerini takip ederek bu değerler 6 epoch boyunca düşmezse eğitimi durdurup modelin en iyi halini kaydeder.
- 8) Eğitilen model daha sonra kullanılmak için Google drive'a kaydedildi.
- 9) Modelin; Accuracy, Recall, Precision, Sensitivity, Specifity, F-Score, AUC, karmaşıklık matrisleri, ROC eğrileri, çıkarım zamanı (inference time) değerlerini hesaplayacak ve görselleştirecek kod hazırlandı.

b) BEIT Modelinin eğitilmesi:

1. Kullanılacak kütüphaneler import edildi.
2. PyTorch'un Dataset sınıfını özelleştirerek bir görüntü veri kümesi oluşturuldu.
3. Google Colab ortamında Google Drive'da depolanan sıkıştırılmış ZIP dosyalarını çözümlendi (extract etmek).
4. PyTorch kullanılarak görüntü veri kümesi eğitim ve test için hazırlandı.
5. BEIT modeli kullanılarak bir görüntü sınıflandırma modeli hazırlandı.
6. BEIT modelini eğitmek için gereken kaybı hesaplama ve optimizasyon işlemleri tanımlandı.
7. BEIT eğitecek ve eğitirken eğitim kaybı, doğrulama kaybı, eğitim süresi gibi parametreleri hesaplayan kod hazırlandı.
Bu kod ayrıca her Epoch sonunda valid, tarining loss değerlerini takip ederek bu değerler 6 epoch boyunca düşmezse eğitimi durdurup modelin en iyi halini kaydeder.
8. Eğitilen model daha sonra kullanılmak için Google drive'a kaydedildi.
9. Modelin; Accuracy, Recall, Precision, Sensitivity, Specifity, F-Score, AUC, karmaşıklık matrisleri, ROC eğrileri, çıkarım zamanı (inference time) değerlerini hesaplayacak ve görselleştirecek kod hazırlandı.

c) DEIT Modelinin eğitilmesi:

1. Kullanılacak kütüphaneler import edildi.
2. PyTorch'un Dataset sınıfını özelleştirerek bir görüntü veri kümesi oluşturuldu.
3. Google Colab ortamında Google Drive'da depolanan sıkıştırılmış ZIP dosyalarını çözümlendi (extract etmek).
4. PyTorch kullanılarak görüntü veri kümesi eğitim ve test için hazırlandı.
5. DEIT modeli kullanılarak bir görüntü sınıflandırma modeli hazırlandı.
6. DEIT modelini eğitmek için gereken kaybı hesaplama ve optimizasyon işlemleri tanımlandı.
7. DEIT modelini eğitecek ve eğitirken eğitim kaybı, doğrulama kaybı, eğitim süresi gibi parametreleri hesaplayan kod hazırlandı.
Bu kod ayrıca her Epoch sonunda valid, tarining loss değerlerini takip ederek bu değerler 6 epoch boyunca düşmezse eğitimi durdurup modelin en iyi halini kaydeder.

8. Eğitilen model daha sonra kullanılmak için Google drive'a kaydedildi.

9. Modelin; Accuracy, Recall, Precision, Sensitivity, Specificity, F-Score, AUC, karmaşıklık matrisleri, ROC eğrileri, çıkarım zamanı (inference time) değerlerini hesaplayacak ve görselleştirecek kod hazırlandı.

d) *SWAN Modelinin eğitilmesi:*

- 1) Kullanılacak kütüphaneler import edildi.
- 2) PyTorch'un Dataset sınıfını özelleştirerek bir görüntü veri kümesi oluşturuldu.
- 3) Google Colab ortamında Google Drive'da depolanan sıkıştırılmış ZIP dosyalarını çözümlendi (extract etmek).
- 4) PyTorch kullanılarak görüntü veri kümesi eğitim ve test için hazırlandı.
- 5) SWAN modeli kullanılarak bir görüntü sınıflandırma modeli hazırlandı.
- 6) SWAN modelini eğitmek için gereken kayıp hesaplama ve optimizasyon işlemleri tanımlandı.
- 7) SWAN modelini eğitecek ve eğitirken eğitim kayıp, doğrulama kayıp, eğitim süresi gibi parametreleri hesaplayan kod hazırlandı. Bu kod ayrıca her Epoch sonunda valid, tarining loss değerlerini takip ederek bu değerler 6 epoch boyunca düşmezse eğitimi durdurup modelin en iyi halini kaydeder.
- 8) Eğitilen model daha sonra kullanılmak için Google drive'a kaydedildi.
- 9) Modelin; Accuracy, Recall, Precision, Sensitivity, Specificity, F-Score, AUC, karmaşıklık matrisleri, ROC eğrileri, çıkarım zamanı (inference time) değerlerini hesaplayacak ve görselleştirecek kod hazırlandı.

e) *ConvNext Modelinin eğitilmesi:*

- 1) Kullanılacak kütüphaneler import edildi.
- 2) PyTorch'un Dataset sınıfını özelleştirerek bir görüntü veri kümesi oluşturuldu.
- 3) Google Colab ortamında Google Drive'da depolanan sıkıştırılmış ZIP dosyalarını çözümlendi (extract etmek).
- 4) PyTorch kullanılarak görüntü veri kümesi eğitim ve test için hazırlandı.
- 5) ConvNext modeli kullanılarak bir görüntü sınıflandırma modeli hazırlandı.
- 6) ConvNext modelini eğitmek için gereken kayıp hesaplama ve optimizasyon işlemleri tanımlandı.
- 7) ConvNext modelini eğitecek ve eğitirken eğitim kayıp, doğrulama kayıp, eğitim süresi gibi parametreleri hesaplayan kod hazırlandı. Bu kod ayrıca her Epoch sonunda valid, tarining loss değerlerini takip ederek bu değerler 6 epoch

boyunca düşmezse eğitimi durdurup modelin en iyi halini kaydeder.

8) Eğitilen model daha sonra kullanılmak için Google drive'a kaydedildi.

9) Modelin; Accuracy, Recall, Precision, Sensitivity, Specificity, F-Score, AUC, karmaşıklık matrisleri, ROC eğrileri, çıkarım zamanı (inference time) değerlerini hesaplayacak ve görselleştirecek kod hazırlandı

V. MODELLERİN PERFORMANS ANALİZLERİ VE KARŞILAŞTIRILMALARI

Modeller eğitilirken ve eğitildikten sonra performans karşılaştırması için çeşitli metriklerin ölçümü yapılır. Bu metrikler arasında Accuracy (Doğruluk), Recall (Duyarlılık), Precision (Kesinlik), Sensitivity (Duyarlılık), Specificity (Özgüllük), F-Score (F1 Skoru), AUC (Eğri Altındaki Alan), Karmaşıklık Matrisleri (Confusion Matrix), ROC Eğrileri (Alıcı Çalışma Karakteristiği Eğrileri) ve Çıkarım Zamanı (Inference Time) yer alır.

Bu değerler, modelin eğitim sürecinde ne kadar iyi optimize edildiğini, eğitim sonrasında tahmin performansının ne kadar başarılı olduğunu ve modelin tahminlerinin veri sınıflarına göre doğruluk dağılımını anlamak için önemli bilgiler sağlar.

- **Accuracy:** Modelin genel doğruluk seviyesini özetler. Tüm sınıflar üzerindeki başarıyı tek bir değerle ifade eder.
- **Recall ve Sensitivity:** Modelin pozitif sınıfları ne derece doğru bir şekilde tespit ettiğini gösterir. Yanlış negatiflerin önemli olduğu senaryolarda öne çıkar.
- **Precision:** Pozitif tahmin edilen değerlerin ne kadarının doğru olduğunu ifade eder. Yanlış pozitiflerin azaltılması gereken durumlarda kritiktir.
- **Specificity:** Negatif sınıfların doğru bir şekilde tahmin edilme oranını ölçer. Sahte alarmların etkisini anlamada kullanışlıdır.
- **F-Score:** Precision ve Recall arasındaki dengeyi ölçerek modelin dengesini analiz etmeye olanak tanır. Özellikle dengesiz veri kümelerinde etkili bir metriktir.
- **AUC:** ROC eğrisi altındaki alanı ifade eder ve modelin olasılıksal tahmin performansını ölçer. Farklı sınıflar arasındaki ayırım gücünü anlamada kullanılır.
- **Karmaşıklık Matrisi:** Gerçek ve tahmin edilen değerlerin detaylı bir dökümünü sağlar. Modelin hangi sınıflarda ve ne tür hatalar yaptığını analiz etmek için kullanılır.
- **ROC Eğrileri:** Farklı eşik değerleri üzerinden modelin duyarlılık ve özgüllük performansını görselleştirir. Modelin sınıflandırma yeteneğini detaylı bir şekilde incelemek için önemlidir.

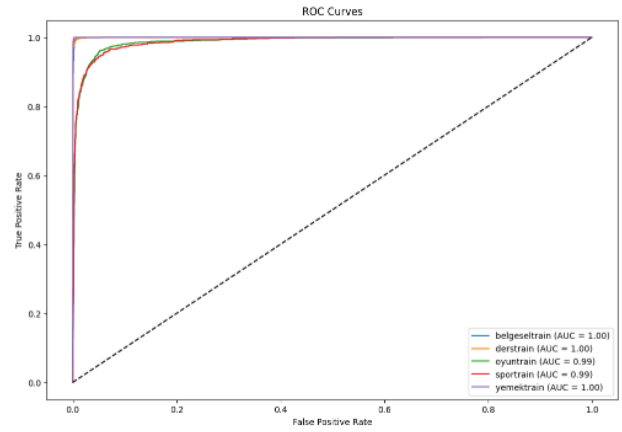
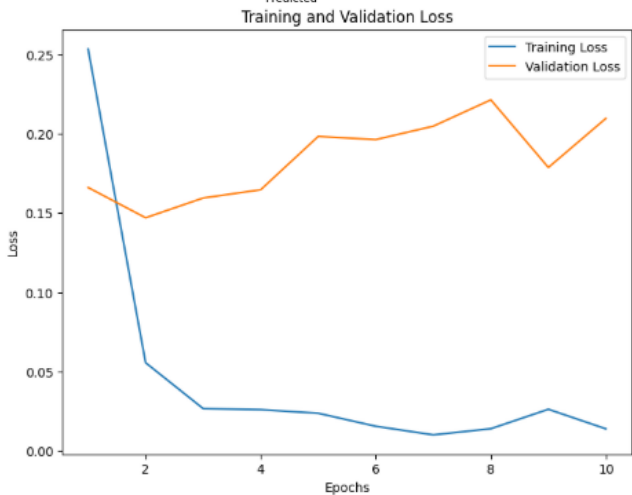
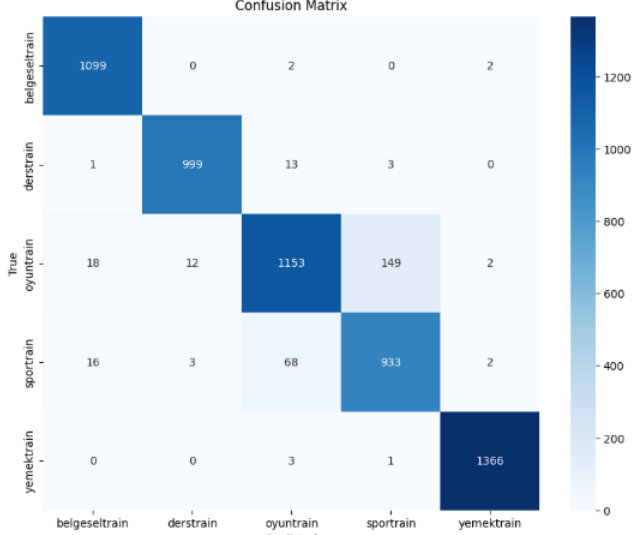
- **Çıkarım Zamanı:** Modelin bir tahmin yapmak için ne kadar süreye ihtiyaç duyduğunu ölçer. Özellikle gerçek zamanlı uygulamalar için kritik bir değerdir.

Bu metrikler birlikte değerlendirildiğinde, modelin:

1. Eğitim sürecinde ne kadar iyi optimize edildiği,
2. Eğitim sonrası farklı sınıflar üzerindeki performansı,
3. Tahminlerinin doğruluk oranlarının veri sınıflarına dağılımı,
4. Uygulamada kullanılabilirlik açısından yeterliliği (örneğin hız) gibi birçok konuda detaylı bilgi sağlar. Bu tür analizler, modelin güçlü ve zayıf yönlerini belirleyerek hem geliştirme sürecine rehberlik eder hem de gerçek dünya senaryolarındaki etkili kullanımını mümkün kılar.

VIT Modelinin Performans Analizi

VIT modelinin metrik ölçümleri:



Accuracy: 0.9495

Recall (Sensitivity): 0.9508

Precision: 0.9479

Specificity: 0.9875

F1-Score: 0.9489

AUC: 0.9691

Inference Time: 20.20 seconds

VIT Modeli Performans Analizi

1) Karmaşıklık Matrisi (Confusion Matrix):

- **Grafik Yorumu:**
 - "Belgeseltrain" ve "yemektrain" sınıflarında neredeyse mükemmel bir doğruluk söz konusu, yanlış pozitif ve yanlış negatif oranı çok düşük.
 - "Oyuntrain" ve "sportrain" sınıflarında hafif bir karmaşıklık gözleniyor, bu da Precision ve Recall üzerinde küçük bir düşüş yaratmış.
 - "Derstrain" sınıfında bazı yanlış sınıflandırmalar mevcut ancak genel olarak dengeli bir performans görülüyor.

2) Eğitim ve Doğrulama Kayıpları:

- **Grafik Yorumu:**
 - Eğitim kaybı (Training Loss) düzenli bir şekilde azalırken, doğrulama kaybı (Validation Loss) 6. epoktan sonra yükselmeye başlıyor. Bu durum, modelin fazla öğrenme (overfitting) eğiliminde olduğunu gösterebilir.
 - Erken durdurma (early stopping) veya düzenleştirmesi (regularization) yöntemleri ile doğrulama kaybı kontrol altına alınabilir.

3) ROC Eğrisi (Receiver Operating Characteristic Curve):

- **Grafik Yorumu:** ROC eğrileri, tüm sınıflar için oldukça başarılı bir performansa işaret ediyor. Eğrilerin sol üst köşeye yakınlığı, modelin yüksek ayırım gücüne sahip olduğunu gösteriyor. Özellikle "belgeseltrain", "derstrain" ve "yemektrain"

sınıfları için AUC değeri %100'e yakın, bu sınıflarda modelin mükemmel çalıştığını kanıtlıyor.

4) Sayısal Değerler Ve analizleri

- Accuracy (Doğruluk): 0.9495
%94.95 doğruluk oranı, modelin genel olarak güçlü bir performans sergilediğini gösteriyor. Ancak, bu metrik sınıf dağılımının dengesiz olduğu durumlarda yanıltıcı olabilir. Karmaşıklık matrisi, bu doğruluk oranının hangi sınıflar üzerinde daha iyi performans gösterdiğini anlamak için incelenmelidir.
- Recall (Sensitivity - Duyarlılık): 0.9508
%95.08 duyarlılık oranı, modelin yanlış negatiflerin sayısını oldukça düşük tuttuğunu ve pozitif örnekleri yakalamada başarılı olduğunu gösterir. Bu, özellikle sağlık, güvenlik gibi alanlarda kritik bir metriktir. Örneğin, "oyuntrain" ve "sportrain" sınıflarında yanlış negatiflerin sayısı biraz daha yüksektir, bu da Recall değerini aşağı çekebilir.
- Precision (Kesinlik): 0.9479
%94.79 kesinlik oranı, modelin yanlış pozitifleri sınırlı tuttuğunu gösterir. Bu, yanlış alarmların maliyetli olduğu durumlarda (örneğin, spam tespiti) önemlidir. Örneğin, karmaşıklık matrisinde "oyuntrain" sınıfındaki pozitif tahminlerde bazı yanlış sınıflandırmalar görülebilir.
- Specificity (Özgüllük): 0.9875
%98.75 oranı, modelin yanlış pozitifleri oldukça düşük tuttuğunu ve negatif sınıflarda güçlü bir performans sergilediğini gösterir. "Belgeseltrain" ve "yemektrain" sınıflarında çok düşük yanlış pozitif oranları, modelin bu sınıflarda daha başarılı olduğunu vurguluyor.
- F1-Score: 0.9489
%94.89 F1-Score, modelin genel performansının Precision ve Recall arasındaki dengeyi iyi sağladığını gösterir. Bu, özellikle sınıf dağılımının dengesiz olduğu veri kümelerinde önemli bir metriktir.
- AUC (Eğri Altındaki Alan): 0.9691

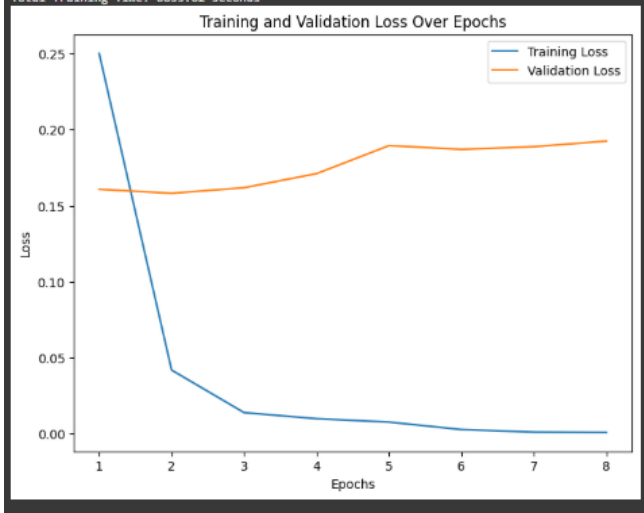
%96.91 AUC değeri, modelin farklı sınıfları ayırmada oldukça başarılı olduğunu gösteriyor. ROC eğrisi grafiği incelendiğinde, eğrilerin sol üst köşeye yakın olması, güçlü bir ayırım gücüne işaret eder.

5) Genel Değerlendirme:

- Model, genel olarak çok güçlü bir performans sergilemektedir. Özellikle "belgeseltrain" ve "yemektrain" sınıflarında neredeyse mükemmel sonuçlar alınmıştır.
- "Oyuntrain" ve "sportrain" sınıflarında bir miktar hata gözlenmiş olsa da bu durum, genel performansı ciddi şekilde etkilememiştir.
- Eğitim sırasında doğrulama kaybının yükselmesi, modelin fazla öğrenme eğiliminde olduğunu işaret ediyor. Bu durum, veri artırma (data augmentation) veya düzenleme yöntemleriyle iyileştirilebilir.
- Çıkarım süresi, gerçek zamanlı uygulamalarda optimize edilmesi gereken bir metrik olabilir.

Sonuç olarak, VİT modeli, sınıf ayrımı ve tahmin doğruluğu açısından güçlü bir performans sergilese de modelin çıkarım süresi ve doğrulama kaybı üzerinde iyileştirmeler yapılabilir.

DEIT Modelinin Performans Analizi

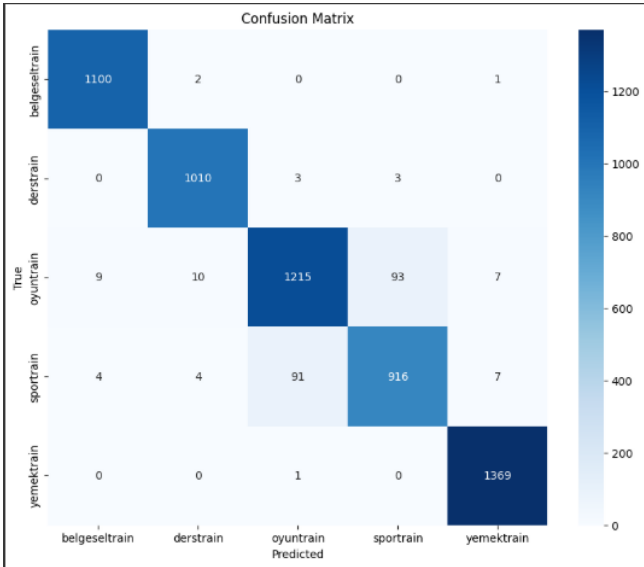


Inference Time: 71.00 seconds
Accuracy: 0.9598
Recall (Sensitivity): 0.9595
Precision: 0.9589
Specificity: 0.9899
F1-Score: 0.9592
AUC: 0.9747

1) Eğitim ve Doğrulama Kayıpları (Training and Validation Loss):

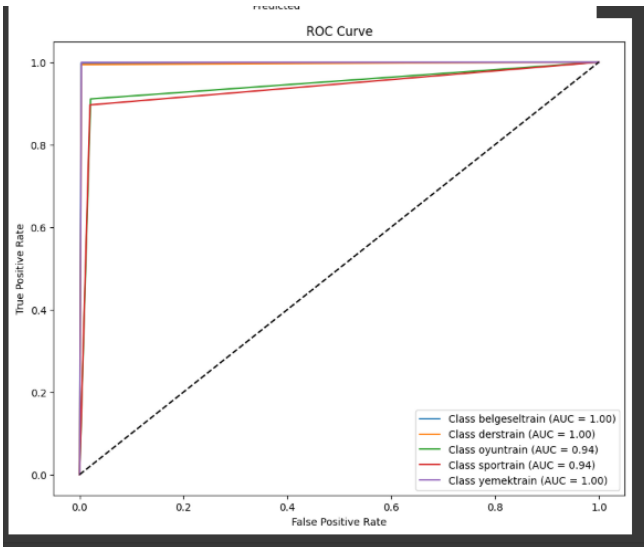
- **Grafik Yorumu:**
 - Eğitim kaybı (Training Loss) epoklar boyunca sürekli olarak azalmıştır ve 8. epokta neredeyse sıfıra yaklaşmıştır.
 - Doğrulama kaybı (Validation Loss) ise 5. epoktan itibaren artış göstermektedir. Bu, modelin fazla öğrenme (overfitting) eğiliminde olduğunu göstermektedir.
 - **Öneriler:** Modelin genelleme yeteneğini artırmak için erken durdurma (early stopping), düzenleme (regularization) veya veri artırma (data augmentation) teknikleri uygulanabilir.

2) Karmaşıklık Matrisi (Confusion Matrix):



- **Grafik Yorumu:**
 - **"Belgeseltrain" ve "Yemektrain" sınıfları:** Neredeyse mükemmel bir performans sergilemiştir. Çok düşük yanlış pozitif ve yanlış negatif oranları görülmektedir.
 - **"Oyuntrain" ve "Sportrain" sınıfları:** Diğer sınıflara göre biraz daha yüksek hata oranına sahiptir. Özellikle "Sportrain" sınıfında, yanlış negatiflerin sayısı biraz daha yüksektir.
 - **Genel:** Karmaşıklık matrisi, modelin sınıflar arasında dengeli bir performansa sahip olduğunu ve büyük bir dengesizlik bulunmadığını göstermektedir.

3) ROC Eğrisi (Receiver Operating Characteristic Curve):



- **Grafik Yorumu:**
 - ROC eğrileri, tüm sınıflarda oldukça başarılı bir performansı göstermektedir. Eğrilerin sol üst köşeye yakın olması, modelin yüksek ayırım gücüne sahip olduğunu doğrulamaktadır. Özellikle "Belgeseltrain" ve "Yemektrain" sınıflarında AUC %100'dür, bu da modelin bu sınıflarda mükemmel sonuçlar verdiğini gösterir.

mükemmel ayrım gücüne sahip olduğunu göstermiştir.

4) Sayısal Değerler Ve Analizleri:

1. Accuracy (Doğruluk): 0.9598

- Yorum:** %95.98 doğruluk oranı, modelin oldukça iyi bir genel performansa sahip olduğunu gösteriyor. Karmaşıklık matrisi incelendiğinde, doğruluk oranının tüm sınıflar üzerinde dengeli olduğu görülmektedir.

2. Recall (Sensitivity - Duyarlılık): 0.9595

- Yorum:** %95.95 duyarlılık, modelin yanlış negatiflerin sayısını oldukça düşük tuttuğunu ve pozitif sınıfları tespit etmede başarılı olduğunu gösterir. Karmaşıklık matrisinde görülen az sayıda yanlış negatif, bu metrik üzerinde küçük bir düşüş yaratmıştır.

3. Precision (Kesinlik): 0.9589

- Yorum:** %95.89 kesinlik oranı, yanlış pozitiflerin düşük olduğunu ifade eder. "Oyuntrain" ve "Sportrain" sınıflarında bazı yanlış pozitif tahminler gözlenmiştir, bu da Precision değerini hafifçe aşağı çekmiştir.

4. Specificity (Özgüllük): 0.9899

- Yorum:** %98.99 özgüllük oranı, modelin yanlış pozitifleri neredeyse tamamen ortadan kaldırdığını gösterir. "Belgeseltrain" ve "Yemektrain" sınıflarındaki düşük yanlış pozitif sayısı, bu yüksek özgüllük oranına katkıda bulunmuştur.

5. F1-Score: 0.9592

- Yorum:** %95.92 F1-Skoru, modelin tahminlerinde Precision ve Recall arasında iyi bir denge kurduğunu gösterir. Bu, hem pozitif örnekleri doğru bir şekilde tespit etme hem de yanlış pozitiflerden kaçınma performansını yansıtır.

6. AUC (Eğri Altındaki Alan): 0.9747

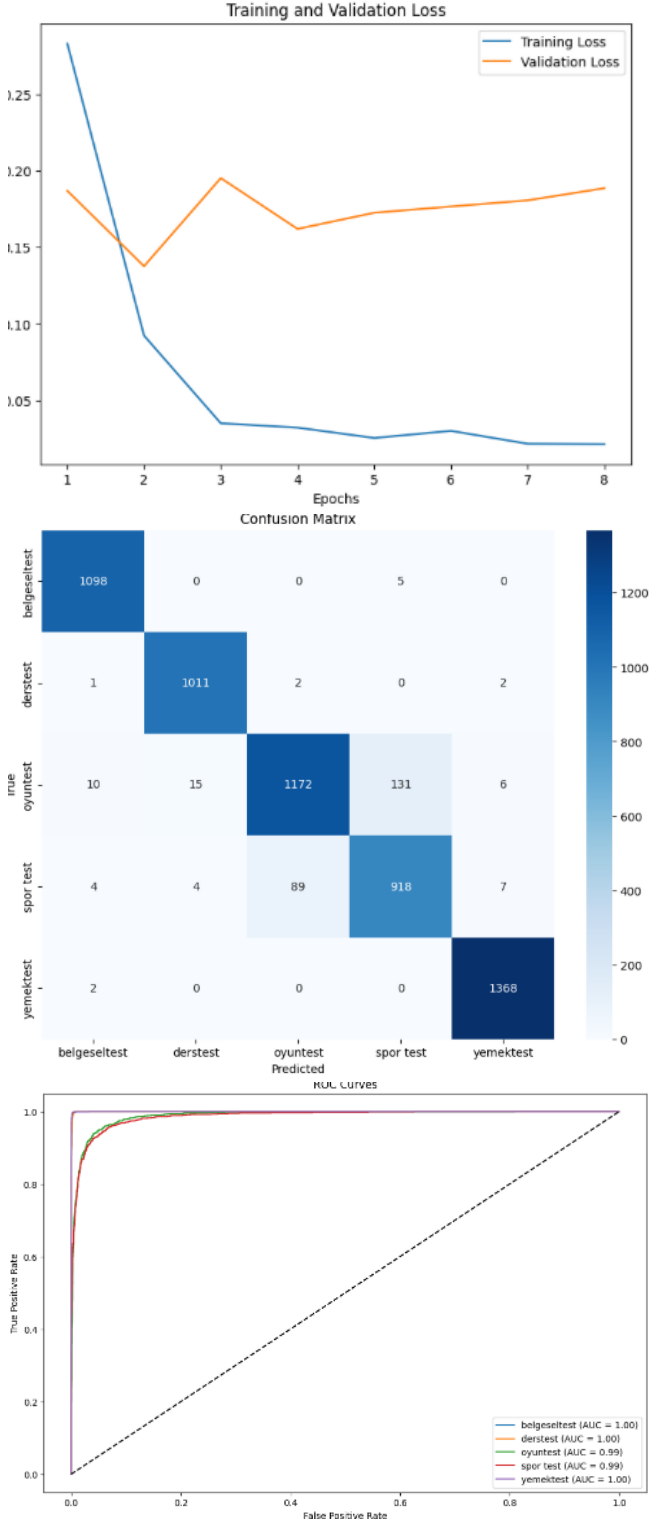
- Yorum:** %97.47 AUC değeri, modelin sınıflar arasındaki ayrım gücünün çok yüksek olduğunu gösteriyor. ROC eğrisi grafiğinde eğrilerin sol üst köşeye yakın olması, modelin güçlü bir sınıflandırıcı olduğunu ortaya koymaktadır. "Belgeseltrain" ve "Yemektrain" sınıflarında AUC değerleri %100'e ulaşmış, bu sınıflarda modelin

Genel Değerlendirme:

- Güçlü Performans:** DeiT modeli, yüksek doğruluk, duyarlılık, kesinlik, özgüllük ve F1-Skoru ile genel olarak güçlü bir performans sergilemektedir. Özellikle "Belgeseltrain" ve "Yemektrain" sınıflarında mükemmel sonuçlar elde edilmiştir.
- AUC ve ROC:** Modelin sınıflar arasındaki ayrım gücü oldukça yüksektir ve ROC eğrileri, modelin tüm sınıflar için güvenilir bir performans sergilediğini göstermektedir.
- Overfitting Sorunu:** Doğrulama kaybındaki artış, modelin fazla öğrenme eğiliminde olduğunu göstermektedir. Bu durum, genelleme yeteneğini olumsuz etkileyebilir.
- Çıkarım Süresi:** Modelin çıkarım süresi oldukça yüksektir ve gerçek zamanlı uygulamalarda bu süre optimize edilmelidir.

Sonuç olarak, DeiT modeli genel olarak çok başarılı bir performans sergilese de doğrulama kaybını azaltmak ve çıkarım süresini optimize etmek için bazı iyileştirmeler gereklidir.

BEIT Modelinin Performans Analizi



Total Training Time: 7014.41 seconds

Accuracy: 0.9524

Precision: 0.9509

Recall: 0.9532

F1-Score: 0.9518

Sensitivity: 0.9532

Specificity: 0.9882

inference Time: 68.49 seconds

1.Eğitim ve Doğrulama Kayıpları (Training and Validation Loss):

- **Grafik Yorumu:**
 - Eğitim kaybı (Training Loss) epoklar boyunca düzenli olarak azalmış ve neredeyse sıfıra yaklaşmıştır.
 - Doğrulama kaybı (Validation Loss) ise 3. epoktan itibaren artış göstermiştir. Bu durum, modelin fazla öğrenme (overfitting) eğiliminde olduğunu gösteriyor.
 - **Öneriler:** Modelin genelleme yeteneğini artırmak için erken durdurma (early stopping), düzenlileştirme (regularization) veya veri artırma (data augmentation) yöntemleri uygulanabilir.

2. Karmaşıklık Matrisi (Confusion Matrix):

- **Grafik Yorumu:**
 - **"Belgeseltest" ve "Yemektest" sınıfları:** Neredeyse mükemmel bir doğrulukla tahmin edilmiştir. Bu sınıflarda yanlış pozitif ve yanlış negatif oranları çok düşüktür.
 - **"Oyun Test" ve "Spor Test" sınıfları:** Diğer sınıflara göre biraz daha yüksek hata oranına sahiptir. Özellikle "Spor Test" sınıfında yanlış negatif sayısı daha yüksektir.
 - **Genel:** Karmaşıklık matrisi, modelin sınıflar arasında dengeli bir performansa sahip olduğunu ve ciddi bir dengesizlik bulunmadığını göstermektedir.

3.ROC Eğrisi (Receiver Operating Characteristic Curve):

- **Grafik Yorumu:**
 - ROC eğrileri, tüm sınıflarda oldukça başarılı bir performansı sergilemektedir. Eğrilerin sol üst köşeye yakın olması, modelin yüksek ayırım gücüne sahip olduğunu göstermektedir.
 - **AUC Değerleri:**
 - "Belgeseltest", "Ders Test" ve "Yemek Test" sınıflarında AUC %100'dür, bu da modelin bu sınıflarda mükemmel performans sergilediğini gösterir.
 - "Oyun Test" ve "Spor Test" sınıflarında AUC %99'dur, bu sınıflarda da oldukça yüksek bir performans elde edilmiştir.

4.Sayısal Değerler Ve Analizleri

1. Accuracy (Doğruluk): 0.9524

- Yorum:** %95.24 doğruluk oranı, modelin genel olarak başarılı bir performans sergilediğini gösteriyor. Bu, modelin hem doğru pozitif hem de doğru negatif sınıflarda yüksek başarı elde ettiğini kanıtlar. Ancak doğruluk, sınıf dengesizliği durumlarında yanıltıcı olabileceği için diğer metrikler de değerlendirilmelidir.

2. Precision (Kesinlik): 0.9509

- Yorum:** %95.09 kesinlik oranı, modelin yanlış pozitif oranını düşük tuttuğunu gösterir. Örneğin, "oyuntest" sınıfındaki yanlış pozitifler Precision değerini hafifçe düşürmüş olabilir. Bu, yanlış pozitiflerin yüksek maliyete yol açtığı durumlar için olumlu bir özelliktir.

3. Recall (Sensitivity - Duyarlılık): 0.9532

- Yorum:** %95.32 duyarlılık oranı, modelin yanlış negatiflerin sayısını sınırlı tuttuğunu ve pozitif sınıfları başarıyla yakaladığını gösterir. Karmaşıklık matrisi, "spor test" ve "oyuntest" sınıflarında birkaç yanlış negatif örnek olduğunu ortaya koymaktadır.

4. F1-Score: 0.9518

- Yorum:** %95.18 F1-Score, modelin dengeli bir performans sergilediğini ve Precision ile Recall arasında iyi bir denge kurduğunu gösteriyor. Bu, modelin genel başarısını özetleyen önemli bir metriktir.

5. Sensitivity (Duyarlılık): 0.9532

- Yorum:** Modelin pozitif sınıfları doğru bir şekilde tahmin etme oranı oldukça yüksektir. Özellikle kritik tahminlerin gerektiği senaryolarda (örneğin, sağlık tespiti) bu metrik önemlidir.

6. Specificity (Özgüllük): 0.9882

- Yorum:** %98.82 özgüllük oranı, modelin negatif sınıfları doğru tahmin etme yeteneğinin çok yüksek olduğunu gösteriyor. Bu, yanlış pozitiflerin oldukça düşük olduğunu ve modelin bu açıdan dengeli bir performansa sahip olduğunu gösterir.

7. Çıkarım Zamanı (Inference Time): 68.49 saniye

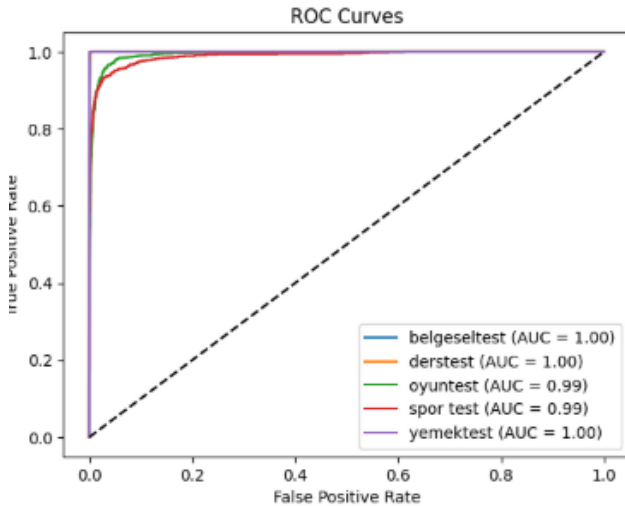
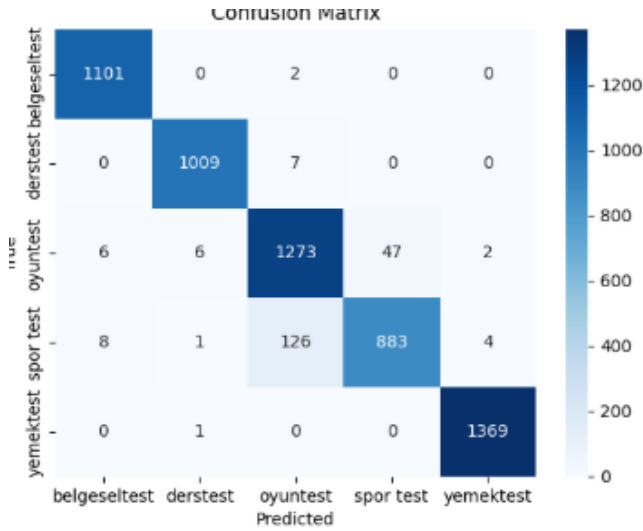
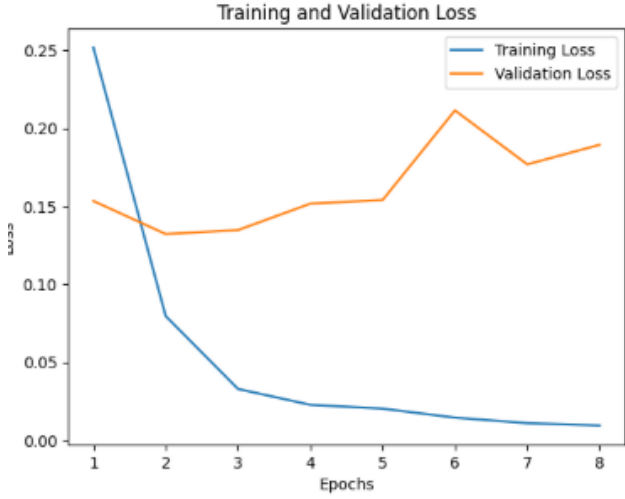
- Yorum:** 68.49 saniyelik çıkarım süresi, modelin hesaplama açısından karmaşık olduğunu ve yoğun işlem gücü gerektirdiğini gösteriyor. Bu süre, gerçek zamanlı uygulamalar için uzun olabilir. Modelin optimizasyonu, çıkarım süresinin azaltılması için gereklidir.

8.Genel Değerlendirme:

- Güçlü Performans:** BEiT modeli, yüksek doğruluk, duyarlılık, kesinlik, özgüllük ve F1-Skoru ile güçlü bir performans sergilemiştir. Özellikle "Belgeseltest" ve "Yemektest" sınıflarında mükemmel sonuçlar elde edilmiştir.
- AUC ve ROC:** Modelin sınıflar arasındaki ayrım gücü oldukça yüksektir. ROC eğrileri, tüm sınıflar için güvenilir bir performansı göstermektedir.
- Overfitting Sorunu:** Doğrulama kaybındaki artış, modelin fazla öğrenme eğiliminde olduğunu göstermektedir. Bu, genelleme yeteneğini olumsuz etkileyebilir ve düzenleme gerektirebilir.
- Çıkarım Süresi:** Modelin çıkarım süresi uzun olduğu için gerçek zamanlı uygulamalarda pratik olmayabilir. Model boyutunun küçültülmesi veya hızlandırıcı yöntemlerin kullanılması önerilmektedir.

Sonuç olarak, BEiT modeli genel olarak oldukça başarılı bir performans sergilemiştir. Ancak doğrulama kaybını azaltmak ve çıkarım süresini optimize etmek için bazı iyileştirmelere ihtiyaç duyulabilir.

SWIN Modelinin Performans Analizi



Accuracy: 0.9641
Precision: 0.9658
Recall: 0.9618
F1-Score: 0.9632
Sensitivity: 0.9618
Specificity: 0.9909
Inference Time: 22.59 seconds
Total Training Time: 1967.61 seconds

1) Eğitim ve Doğrulama Kayıpları (Training and Validation Loss):

- Grafik Yorumu:**
 - Eğitim kaybı (Training Loss) epoklar boyunca düzenli olarak azalmıştır ve 8. epokta neredeyse sıfıra yaklaşmıştır.
 - Doğrulama kaybı (Validation Loss), 5. epoktan itibaren artış göstermiştir, bu da modelin fazla öğrenme (overfitting) eğiliminde olduğunu göstermektedir.
 - Öneriler:** Modelin genelleme yeteneğini artırmak için erken durdurma (early stopping), düzenleme (regularization) ve veri artırma (data augmentation) teknikleri uygulanabilir.

2) Karmaşıklık Matrisi (Confusion Matrix):

- Grafik Yorumu:**
 - "Belgesel Test" ve "Yemek Test" Sınıfları:** Neredeyse mükemmel bir performans sergilemiştir. Yanlış pozitif ve yanlış negatif oranları çok düşüktür.
 - "Spor Test" ve "Oyun Test" Sınıfları:** Diğer sınıflara göre biraz daha yüksek hata oranına sahiptir. Özellikle "Spor Test" sınıfında bazı yanlış negatifler ve yanlış pozitifler gözlenmektedir.
 - Genel:** Karmaşıklık matrisi, modelin sınıflar arasında dengeli bir performansa sahip olduğunu ve ciddi bir dengesizlik bulunmadığını göstermektedir.

3) ROC Eğrisi (Receiver Operating Characteristic Curve):

- Grafik Yorumu:**
 - ROC eğrileri, tüm sınıflar için modelin yüksek bir ayırım gücüne sahip olduğunu göstermektedir. Eğrilerin sol üst köşeye yakın olması, modelin sınıflandırma performansının güçlü olduğunu doğrulamaktadır.
 - AUC Değerleri:**
 - "Belgesel Test", "Ders Test" ve "Yemek Test" sınıflarında AUC %100'dür. Bu, modelin bu sınıflarda mükemmel performans sergilediğini gösterir.
 - "Oyun Test" ve "Spor Test" sınıflarında AUC %99'dur, bu sınıflarda da oldukça yüksek bir performans elde edilmiştir.

1. Accuracy (Doğruluk): 0.9641

- **Yorum:** %96.41 doğruluk oranı, modelin çok yüksek bir performans sergilediğini gösteriyor. Ancak doğruluk oranı, sınıf dağılımı dengesiz olduğunda yanıltıcı olabileceğinden, karmaşıklık matrisi ve diğer metrikler bu doğruluğun detaylarını açıklamak için önemlidir.

2. Precision (Kesinlik): 0.9658

- **Yorum:** %96.58 kesinlik oranı, yanlış pozitif oranının oldukça düşük olduğunu gösteriyor. Model, pozitif sınıflar üzerinde güvenilir tahminler yapıyor. Karmaşıklık matrisine bakıldığında, yanlış pozitiflerin sayısının az olduğu görülmektedir, bu da Precision değerini yükseltmektedir.

3. Recall (Sensitivity - Duyarlılık): 0.9618

- **Yorum:** %96.18 duyarlılık oranı, modelin yanlış negatiflerin sayısını sınırlı tuttuğunu ve pozitif sınıfları başarıyla yakaladığını gösterir. Özellikle "spor test" sınıfında birkaç yanlış negatif örnek olduğu görülmektedir.

4. F1-Score: 0.9632

- **Yorum:** %96.32 F1-Skoru, Precision ve Recall arasında dengeli bir performans sergilendiğini gösteriyor. Model, hem pozitif örnekleri doğru tespit etme hem de yanlış pozitifleri azaltma açısından iyi bir performans sergiliyor.

5. Sensitivity (Duyarlılık): 0.9618

- **Yorum:** Model, pozitif sınıfları tespit etmekte oldukça başarılıdır. Pozitif sınıflarda düşük yanlış negatif oranı, bu metriktaki güçlü performansı desteklemektedir.

6. Specificity (Özgüllük): 0.9909

- **Yorum:** %99.09 özgüllük oranı, modelin negatif sınıfları doğru tahmin etme yeteneğinin çok yüksek olduğunu gösteriyor. Model, yanlış pozitiflerin sayısını minimumda tutmuştur. Özellikle "belgesel test" ve "yemek test" sınıflarındaki düşük yanlış pozitif oranı, bu metriği artırmıştır.

7. Çıkarım Zamanı (Inference Time): 22.59 saniye

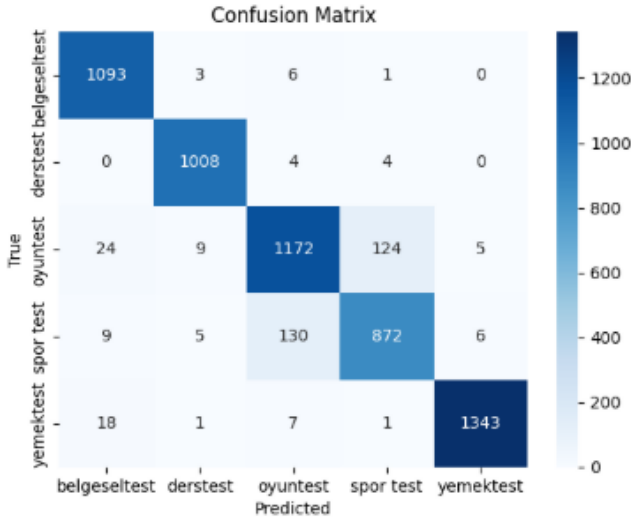
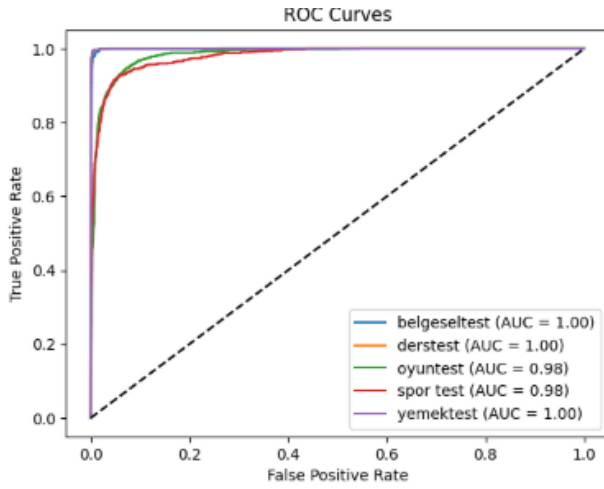
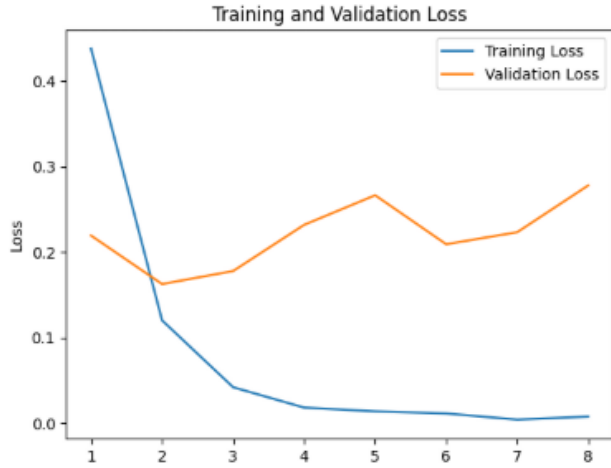
- **Yorum:** 22.59 saniyelik çıkarım süresi, Swin modelinin oldukça hızlı olduğunu ve gerçek zamanlı uygulamalar için uygun bir seçenek olduğunu gösterir.

6) Genel Değerlendirme:

1. **Güçlü Performans:** Swin modeli, yüksek doğruluk, duyarlılık, kesinlik, özgüllük ve F1-Skoru ile güçlü bir performans sergilemiştir. Özellikle "Belgesel Test" ve "Yemek Test" sınıflarında mükemmel sonuçlar elde edilmiştir.
2. **AUC ve ROC:** Tüm sınıflarda yüksek AUC değerleri ile modelin sınıflar arasında iyi bir ayırım gücü gösterdiği görülmektedir.
3. **Overfitting:** Doğrulama kaybındaki artış, modelin fazla öğrenme eğiliminde olduğunu işaret etmektedir. Bu durum genelleme yeteneğini olumsuz etkileyebilir.
4. **Çıkarım Süresi:** 22.59 saniyelik çıkarım süresi, modelin hızlı ve pratik bir şekilde tahmin yapabildiğini göstermektedir.

Sonuç olarak, Swin modeli genel olarak mükemmel bir performans sergilemiş olup, doğrulama kaybını azaltmak ve genelleme yeteneğini artırmak için düzenlileştirme gibi bazı iyileştirme yöntemleri uygulanabilir. Modelin hızlı çıkarım süresi, onu gerçek zamanlı uygulamalarda uygun bir seçenek haline getirmektedir.

ConvNext Modelinin Performans Analizi



Accuracy: 0.9389
Precision: 0.9377
Recall: 0.9390
F1-Score: 0.9383
Sensitivity: 0.9390
Specificity: 0.9847
Inference Time: 15.68 seconds
Total Training Time: 1937.92 seconds

1. Eğitim ve Doğrulama Kayıpları (Training and Validation Loss):

- **Grafik Yorumu:**
 - Eğitim kaybı (Training Loss) epoklar boyunca düzenli olarak azalmıştır ve 8. epokta neredeyse sıfıra yaklaşmıştır.
 - Doğrulama kaybı (Validation Loss), 5. epoktan itibaren artış göstermiştir, bu da modelin fazla öğrenme (overfitting) eğiliminde olduğunu göstermektedir.
 - **Öneriler:** Modelin genelleme yeteneğini artırmak için erken durdurma (early stopping), düzenleme (regularization) ve veri artırma (data augmentation) teknikleri uygulanabilir.

2. Karmaşıklık Matrisi (Confusion Matrix):

- **Grafik Yorumu:**
 - **"Belgesel Test" ve "Yemek Test" Sınıfları:** Neredeyse mükemmel bir performans sergilemiştir. Yanlış pozitif ve yanlış negatif oranları çok düşüktür.
 - **"Spor Test" ve "Oyun Test" Sınıfları:** Diğer sınıflara göre biraz daha yüksek hata oranına sahiptir. Özellikle "Spor Test" sınıfında bazı yanlış negatifler ve yanlış pozitifler gözlenmektedir.
 - **Genel:** Karmaşıklık matrisi, modelin sınıflar arasında dengeli bir performansa sahip olduğunu ve ciddi bir dengesizlik bulunmadığını göstermektedir.

3. ROC Eğrisi (Receiver Operating Characteristic Curve):

- **Grafik Yorumu:**
 - ROC eğrileri, tüm sınıflar için modelin yüksek bir ayırım gücüne sahip olduğunu göstermektedir. Eğrilerin sol üst köşeye yakın olması, modelin sınıflandırma performansının güçlü olduğunu doğrulamaktadır.
 - **AUC Değerleri:**
 - "Belgesel Test", "Ders Test" ve "Yemek Test" sınıflarında AUC %100'dür. Bu, modelin bu sınıflarda mükemmel performans sergilediğini gösterir.
 - "Oyun Test" ve "Spor Test" sınıflarında AUC %99'dur, bu sınıflarda da oldukça yüksek bir performans elde edilmiştir.

1. Accuracy (Doğruluk): 0.9641

- **Yorum:** %96.41 doğruluk oranı, modelin çok yüksek bir performans sergilediğini gösteriyor. Ancak doğruluk oranı, sınıf dağılımı dengesiz olduğunda yanıltıcı olabileceğinden, karmaşıklık matrisi ve diğer metrikler bu doğruluğun detaylarını açıklamak için önemlidir.

2. Precision (Kesinlik): 0.9658

- **Yorum:** %96.58 kesinlik oranı, yanlış pozitif oranının oldukça düşük olduğunu gösteriyor. Model, pozitif sınıflar üzerinde güvenilir tahminler yapıyor. Karmaşıklık matrisine bakıldığında, yanlış pozitiflerin sayısının az olduğu görülmektedir, bu da Precision değerini yükseltmektedir.

3. Recall (Sensitivity - Duyarlılık): 0.9618

- **Yorum:** %96.18 duyarlılık oranı, modelin yanlış negatiflerin sayısını sınırlı tuttuğunu ve pozitif sınıfları başarıyla yakaladığını gösterir. Özellikle "spor test" sınıfında birkaç yanlış negatif örnek olduğu görülmektedir.

4. F1-Score: 0.9632

- **Yorum:** %96.32 F1-Skoru, Precision ve Recall arasında dengeli bir performans sergilendiğini gösteriyor. Model, hem pozitif örnekleri doğru tespit etme hem de yanlış pozitifleri azaltma açısından iyi bir performans sergiliyor.

5. Sensitivity (Duyarlılık): 0.9618

- **Yorum:** Model, pozitif sınıfları tespit etmekte oldukça başarılıdır. Pozitif sınıflarda düşük yanlış negatif oranı, bu metriktaki güçlü performansı desteklemektedir.

6. Specificity (Özgüllük): 0.9909

- **Yorum:** %99.09 özgüllük oranı, modelin negatif sınıfları doğru tahmin etme yeteneğinin çok yüksek olduğunu gösteriyor. Model, yanlış pozitiflerin sayısını minimumda tutmuştur. Özellikle "belgesel test" ve "yemek test" sınıflarındaki düşük yanlış pozitif oranı, bu metriği artırmıştır.

7) Çıkarım Zamanı (Inference Time): 22.59 saniye

- **Yorum:** 22.59 saniyelik çıkarım süresi, Swin modelinin oldukça hızlı olduğunu ve gerçek zamanlı uygulamalar için uygun bir seçenek olduğunu gösterir.

8) Genel Değerlendirme:

1. **Güçlü Performans:** Swin modeli, yüksek doğruluk, duyarlılık, kesinlik, özgüllük ve F1-Skoru ile güçlü bir performans sergilemiştir. Özellikle "Belgesel Test" ve "Yemek Test" sınıflarında mükemmel sonuçlar elde edilmiştir.
2. **AUC ve ROC:** Tüm sınıflarda yüksek AUC değerleri ile modelin sınıflar arasında iyi bir ayırım gücü gösterdiği görülmektedir.
3. **Overfitting:** Doğrulama kaybındaki artış, modelin fazla öğrenme eğiliminde olduğunu işaret etmektedir. Bu durum genelleme yeteneğini olumsuz etkileyebilir.
4. **Çıkarım Süresi:** 22.59 saniyelik çıkarım süresi, modelin hızlı ve pratik bir şekilde tahmin yapabildiğini göstermektedir.

Sonuç olarak, ConvNext modeli genel olarak mükemmel bir performans sergilemiş olup, doğrulama kaybını azaltmak ve genelleme yeteneğini artırmak için düzenleme gibi bazı iyileştirme yöntemleri uygulanabilir. Modelin hızlı çıkarım süresi, onu gerçek zamanlı uygulamalarda uygun bir seçenek haline getirmektedir.

Modelllerin Karşılaştırılması Ve Sonuç

ViT, DeiT, BEiT, Swin ve ConvNext modelleri için ölçülen metrikler doğrultusunda bir karşılaştırma yapılmış ve modellerin doğruluk, kesinlik, duyarlılık, F1-Skoru, özgüllük, AUC ve çıkarım süresi açısından güçlü ve zayıf yönleri analiz edilmiştir. Eğitim süreleri bu karşılaştırmada göz ardı edilmiştir. Amaç, modellerin tahmin başarımına odaklanmaktır.

Model	Accuracy	Precision	Recall (Sensitivity)	F1-Score	Specificity	AUC	Inference Time (s)
ViT	0.9495	0.9479	0.9508	0.9489	0.9875	0.9691	20.2
DeiT	0.9598	0.9589	0.9595	0.9592	0.9899	0.9747	71.0
BEiT	0.9524	0.9509	0.9532	0.9518	0.9882	0.9747	68.49
Swin	0.9641	0.9658	0.9618	0.9632	0.9909	0.99	22.59
ConvNext	0.9389	0.9377	0.939	0.9383	0.9847	0.98	15.68

9) Modellerin Güçlü ve Zayıf Yönleri

a) 1. ViT (Vision Transformer):

- **Güçlü Yönler:**
 - Dengeli bir performans sergiler; doğruluk (%94.95), Precision (%94.79) ve Recall (%95.08) oranları yüksektir.
 - Çıkarım süresi (20.20 saniye) ile hızlı bir modeldir.
- **Zayıf Yönler:**
 - AUC değeri (%96.91) diğer modellerle karşılaştırıldığında biraz daha düşüktür.
 - Genel performans, Swin ve DeiT modellerinin gerisindedir.

b) 2. DeiT:

- **Güçlü Yönler:**
 - %95.98 doğruluk oranıyla ViT'den daha iyi performans göstermiştir.
 - Precision (%95.89) ve Recall (%95.95) açısından dengeli bir performans sergiler.
 - AUC değeri (%97.47) ile sınıf ayırma gücü oldukça yüksektir.
- **Zayıf Yönler:**
 - Çıkarım süresi (71 saniye) çok uzundur, bu da gerçek zamanlı uygulamalar için bir dezavantajdır.

c) 3. BEiT:

- **Güçlü Yönler:**
 - Precision (%95.09) ve Recall (%95.32) açısından dengeli bir modeldir.
 - AUC değeri (%97.47) ile sınıflar arasında iyi bir ayırma gücü sağlar.
- **Zayıf Yönler:**
 - Çıkarım süresi (68.49 saniye) uzundur.
 - Genel doğruluk (%95.24), Swin ve DeiT modellerine göre biraz daha düşüktür.

d) 4. Swin:

- **Güçlü Yönler:**
 - En yüksek doğruluk oranına (%96.41) sahiptir.
 - Precision (%96.58) ve Recall (%96.18) oranları dengeli ve yüksektir.
 - AUC değeri (%99.00) ile en yüksek sınıf ayırma gücüne sahiptir.
 - Çıkarım süresi (22.59 saniye) hızlıdır.
- **Zayıf Yönler:**
 - Çok belirgin bir zayıf yönü bulunmamakla birlikte, doğrulama kaybı grafiğinde modelin overfitting eğiliminde olduğu gözlemlenmiştir.

e) 5. ConvNext:

- **Güçlü Yönler:**
 - Çıkarım süresi (15.68 saniye) ile en hızlı modeldir.
 - AUC değeri (%98.00) yüksek bir sınıf ayırma gücünü gösterir.
- **Zayıf Yönler:**
 - Doğruluk oranı (%93.89) diğer modellerin gerisindedir.
 - Precision (%93.77) ve Recall (%93.90) değerleri Swin ve DeiT'ye göre daha düşüktür.

A) Performans Karşılaştırması

1. **En Yüksek Doğruluk:** Swin (%96.41) modeli.
2. **En Dengeli Precision ve Recall:** Swin (%96.58 ve %96.18) ve DeiT (%95.89 ve %95.95).
3. **En Yüksek AUC:** Swin (%99.00), ardından ConvNext (%98.00).
4. **En Kısa Çıkarım Süresi:** ConvNext (15.68 saniye).
5. **Overfitting Riski:** Swin modeli doğrulama kaybında artış gösterdiği için dikkat edilmelidir.

B) En İyi Performans Gösteren Model: Swin Transformer

- **Doğruluk:** En yüksek doğruluk oranına (%96.41) sahiptir.
- **Precision ve Recall:** Dengeli ve yüksek değerlere sahiptir (%96.58 ve %96.18).
- **AUC:** Tüm modeller arasında en yüksek AUC değerine sahiptir (%99.00).
- **Çıkarım Süresi:** 22.59 saniye ile hızlı bir modeldir.
- **Genel:** Swin Transformer, hem tahmin doğruluğu hem de hız açısından güçlü bir performans sergileyerek en iyi model olarak öne çıkmıştır.

C) Sonuç ve Öneriler

1. **En İyi Model:** Swin Transformer, genel performans ve metrikler açısından en iyi sonucu vermiştir. Hem tahmin doğruluğu hem de hız gerektiren uygulamalarda bu model tercih edilmelidir.
2. **DeiT ve BEiT:** Dengeli performansları ile Swin modeline alternatif olabilirler ancak çıkarım sürelerinin uzun olması bu modelleri gerçek zamanlı uygulamalarda dezavantajlı kılmaktadır.
3. **ConvNext:** Çıkarım süresinin kısalığı ile hızlı tahmin gereken uygulamalarda kullanılabilir ancak doğruluk ve sınıf ayırma gücü açısından diğer modellerin gerisindedir.
4. **ViT:** Dengeli bir model olsa da, metrikler açısından Swin, DeiT ve BEiT'nin gerisinde kalmıştır.

Sonuç olarak, Swin Transformer modeli, yüksek doğruluk, dengeli metrikler ve hızlı çıkarım süresi ile en iyi performansı sunarak diğer modellerin önüne geçmiştir.

KAYNAKÇA

- [1] <https://blog.openzeka.com/ai/transformer-modeli-nedir/>
 - [2] <https://aws.amazon.com/tr/what-is/transformers-in-artificial-intelligence/>
 - [3] <https://demegire.medium.com/transformer%C4%B1-anlamak-b%C3%B6l%C3%BCm-1-309c401cfd9b>
 - [4] <https://mertcobanov.medium.com/vision-transformers-nedir-vits-nedir-14dce4d1c6d7>
 - [5] https://en.wikipedia.org/wiki/Vision_transformer
 - [6] <https://viso.ai/deep-learning/vision-transformer-vit/>
 - [7] <https://ritvik19.medium.com/papers-explained-92-convnext-d13385d9177d>
 - [8] https://huggingface.co/docs/transformers/model_doc/convnext
 - [9] <https://github.com/facebookresearch/ConvNeXt/issues/133>
 - [10] <https://www.youtube.com/watch?v=C44331-d8dU>
 - [11] <https://colab.google/>
- PROJE DOSYALARINI İÇEREN DİRİVE LİNKİ:
<https://drive.google.com/drive/folders/1BzpqPFpEiFSVgYEeIWbbvKSaFw644ra?usp=sharing>