

# Duruş Bozukluğu Tespiti İçin Model Eğitimi

Ensar Akbaş  
Bilişim Sistemleri Mühendisliği  
Kocaeli Üniversitesi  
ensarr.akbas@gmail.com

Berkay Yüce  
Bilişim Sistemleri Mühendisliği  
Kocaeli Üniversitesi  
berkay.yce10@gmail.com

**Özet—** Bu projede, Vision Transformer (ViT), Swin Transformer ve BEiT modelleri kullanılarak beş farklı hastalık sınıfına ait görüntülerle sınıflandırma yapılmıştır. Veri artırma teknikleri ile modellerin performansı iyileştirilmiş ve her bir modelin doğruluk, duyarlılık, hassasiyet gibi performans metrikleri hesaplanarak karşılaştırılmıştır.

**Anahtar Kelimeler—** Vision Transformer, Swin Transformer, BEiT, görüntü sınıflandırma, performans ölçütleri.

## I. Giriş

Bu çalışmada, Vision Transformer (ViT), Swin Transformer ve BEiT modellerinin görüntü sınıflandırma üzerindeki performansları karşılaştırılmıştır. ViT, görüntüleri küçük parçalara ayırarak dikkat mekanizmalarıyla özellik çıkarımı yaparken, Swin Transformer yerel pencereler ve kaydırma mekanizması kullanarak daha etkin bir özellik öğrenimi sunmaktadır. BEiT modeli, görsel dil modellemesi üzerine kurulmuş olup, görselleri işleyiş şekliyle ViT ve Swin Transformer'a kıyasla farklı bir yaklaşım sunar. Projede, kifo, pektus ekskavatum, pektus karinatum, poland sendromu ve skolyoz gibi beş hastalık sınıfı üzerinde sınıflandırma yapılmıştır. Veri artırma teknikleriyle modellerin performansları artırılmış ve bu modeller arasındaki etkinlik farkları incelenmiştir.

## II. Selenium Nedir ? Nasıl Kullandık ?

Selenium, web tarayıcılarını otomatikleştirmek için kullanılan bir araçtır ve Python gibi programlama dilleri ile entegre çalışabilir. Veri toplama süreçlerinde, Selenium'u kullanarak Google Görseller üzerinden belirli anahtar kelimeler ile arama yapıp, görselleri indirme işlemini gerçekleştirdik. Sayfa içindeki görselleri bulmak için CSS seçicileri ve XPATH gibi seçicilerden faydalandık. Ardından, görselleri sayfanın aşağısına doğru kaydırarak yeni görsellerin yüklenmesini sağladık. Bu görsellerin URL'lerini topladık ve HTTP üzerinden indirdik. Böylece, belirli bir konuya dair geniş bir veri seti oluşturarak, sınıflandırma projeleri için görsel veriyi hazırladık. Selenium için gerekli kütüphaneleri aşağıdaki gibi kurduk. Ayrıca dosya işlemleri için 'os' modülünü de projemiz içine dahil ettik.

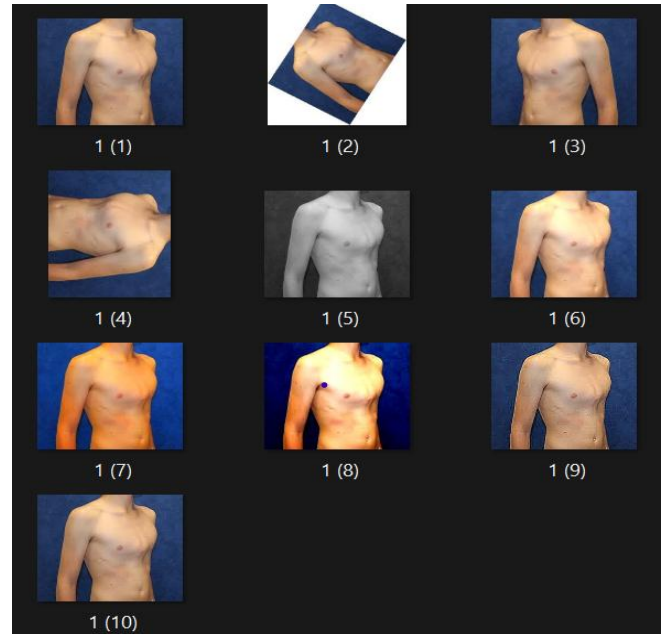
## III. Pillow Nedir ? Nasıl Kullandık ?

Pillow, Python'da görüntü işleme ve düzenleme için popüler bir kütüphanedir. Görsellerin boyutlandırılması, kontrast, parlaklık, keskinlik gibi özelliklerin ayarlanması ve filtrelerin uygulanması gibi işlemleri kolaylaştırır. Projede, Pillow ile indirilen görseller üzerinde veri ön işleme (data preprocessing) yaptık. Görselleri yeniden boyutlandırarak

model için daha uygun hale getirdik. Ayrıca, bu kütüphane sayesinde farklı renk ve parlaklık düzenlemeleri yaparak verilerimizi çeşitlendirdik. Pillow'un sağladığı işlevler sayesinde veri setimizin kalitesini ve homojenliğini artırdık. Pillow'u kullanabilmemiz için gerekli kütüphaneleri projemiz içine dahil ettik. Burada da dosyalama işlemlerini kullandığımız için 'os' modülünü tekrardan eklemiş olduk.

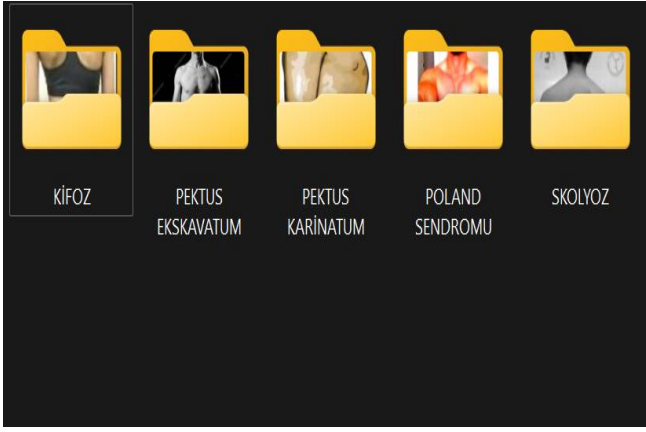
## IV. Veri Artırma (Data Augmentation) Nedir ? Nasıl Kullandık ? Önemi Nedir?

Veri artırma (data augmentation), mevcut veri setini artırmak için kullanılan bir tekniktir. Görseller üzerinde döndürme, çevirme, parlaklık ayarı, zoom gibi farklı işlemler uygulanarak yeni veriler oluşturulur. Bu projede, Pillow kütüphanesi ile data augmentation işlemleri gerçekleştirdik ve görsellerin çeşitliliğini artırdık. Bu işlem, modelin veri seti üzerindeki öğrenme kapasitesini genişletir, aşırı öğrenme (overfitting) riskini azaltır ve modelin daha çeşitli verilerle eğitim almasını sağlar.



## V. Veri Setimiz

Bu projede, görsel verilerin düzenlenmesinde ve çeşitlendirilmesinde kullanılan işlemler, modelin doğruluk ve genelleme kapasitesini artırmak adına yapılan adımlardır. Bu aşamalarda kullanılan teknikler ve kütüphaneler sayesinde elde edilen veri seti, makine öğrenmesi modelinin daha doğru ve kapsamlı sonuçlar vermesine katkıda bulunacaktır.



## VI. Eğitim ve Test Kümesi Ayrımı

Bu çalışmada kullanılan veri kümesi, beş farklı sınıfı (Kifoz, Pektus Ekskavatum, Pektus Karinatum, Poland Sendromu, Skolyoz) içermektedir. Başlangıçta toplamda 25.000 görüntüden oluşan bir veri kümesi kullanılmıştır. Ancak, veri kümesinin işlenmesi ve modelin eğitim sürecini hızlandırmak amacıyla veri kümesinin boyutu 13.153 görüntüye düşürülmüştür. Bu görüntüler, modelin eğitimi ve test işlemlerinde kullanılmak üzere eğitim ve test seti olarak ikiye ayrılmıştır.

**Eğitim Seti:** Modelin öğrenme sürecinde kullanılan toplam 10.521 görüntü içerir.

**Test Seti:** Modelin doğruluğunu ve genel başarımını değerlendirmek için kullanılan toplam 2.632 görüntü içerir.

Her bir sınıfın eğitim ve test setine ait görüntü sayıları resimdeki gibidir:

KİFOZ:	Eğitim seti: 2064 görüntü, Test seti: 516 görüntü
PEKTUS EKSİKAVATUM:	Eğitim seti: 2100 görüntü, Test seti: 526 görüntü
PEKTUS KARİNATUM:	Eğitim seti: 2164 görüntü, Test seti: 541 görüntü
POLAND SENDROMU:	Eğitim seti: 2078 görüntü, Test seti: 520 görüntü
SKOLYOZ:	Eğitim seti: 2115 görüntü, Test seti: 529 görüntü
<b>Toplam Eğitim görüntü sayısı:</b>	<b>10521</b>
<b>Toplam Test görüntü sayısı:</b>	<b>2632</b>
<b>Toplam Görüntü Sayısı:</b>	<b>13153</b>

### A. Vision Transformer (ViT)

Vision Transformer (ViT), görüntü sınıflandırma problemleri için kullanılan modern bir derin öğrenme modelidir. Bu model, geleneksel evrişimli sinir ağlarından (CNN) farklı olarak, görüntüleri yama (patch) tabanlı olarak işler ve her yamayı bir kelime gibi ele alarak Transformer mimarisi ile analiz eder. Bu yaklaşım, görüntülerin küresel bağlamını daha etkili bir şekilde öğrenmeye olanak tanır. ViT, özellikle büyük

veri kümelerinde güçlü bir performans sergilemesiyle öne çıkar ve derin öğrenme uygulamalarında giderek daha popüler hale gelmektedir.

### I. Model Eğitimi

Bu projede, ViT modeli kullanılarak belirli sınıflandırma problemlerinin çözümü hedeflenmiştir. Modelin eğitim süreci 10 epoch boyunca gerçekleştirilmiştir. Eğitim sırasında modelin doğruluk (accuracy) ve kayıp (loss) değerleri düzenli olarak takip edilmiştir. İlk epoch'ta modelin eğitim doğruluğu %23,77 iken doğrulama doğruluğu %34,08 olarak kaydedilmiştir. Eğitim ilerledikçe modelin performansı artış göstermiştir.

Son epoch'ta eğitim doğruluğu %91,03, doğrulama doğruluğu ise %78,95 olarak gözlemlenmiştir. Benzer şekilde, eğitim kaybı başlangıçta 1,7818 iken, son epoch'ta 0,2428'e düşmüştür. Doğrulama kaybı ise 1,4029'dan 0,6786'ya kadar azalmıştır.

Bu sonuçlar, modelin eğitim sırasında hem eğitim hem de doğrulama veri kümesine başarılı bir şekilde adapte olduğunu göstermektedir. Eğitim süresi toplamda 946 saniye olarak ölçülmüştür ve modelin doğrulama doğruluğundaki artış, modelin genelleme kapasitesinin arttığını ifade etmektedir. Eğitim süreci aşağıdaki gibidir:

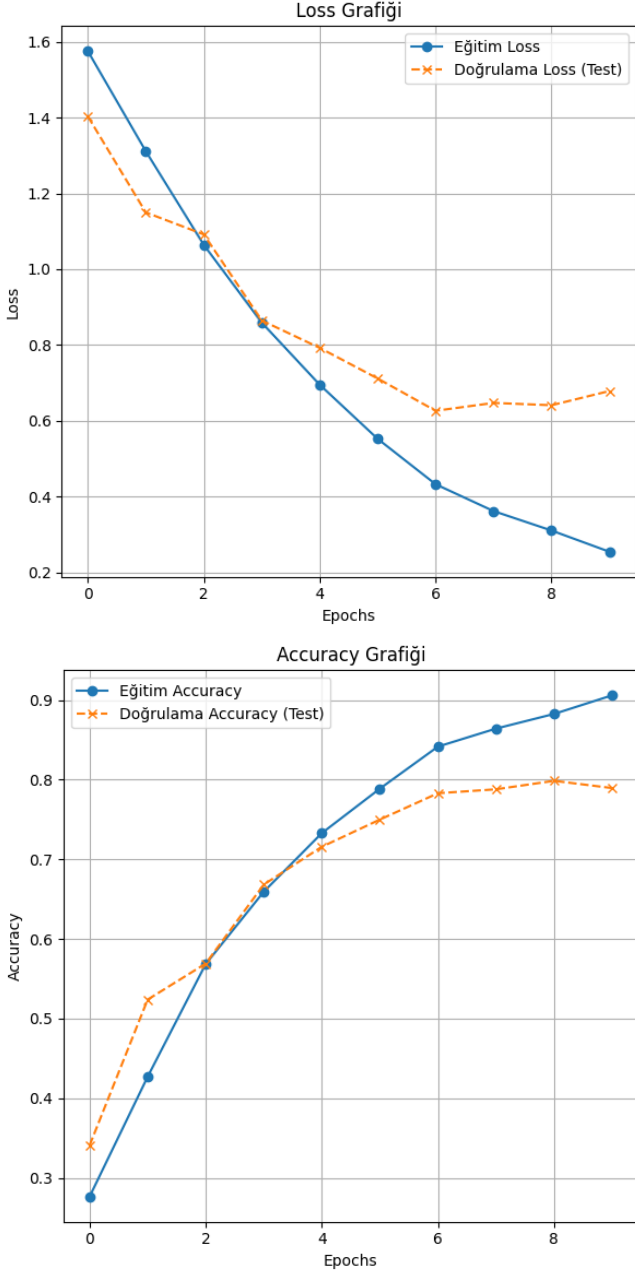
Epoch 1/10	62s 823ms/step - accuracy: 0.2377 - loss: 1.7818 - val_accuracy: 0.3408 - val_loss: 1.4029
Epoch 2/10	28s 43ms/step - accuracy: 0.3878 - loss: 1.3700 - val_accuracy: 0.5239 - val_loss: 1.1496
Epoch 3/10	29s 44ms/step - accuracy: 0.5597 - loss: 1.0935 - val_accuracy: 0.5691 - val_loss: 1.0916
Epoch 4/10	41s 44ms/step - accuracy: 0.6404 - loss: 0.8874 - val_accuracy: 0.6683 - val_loss: 0.8646
Epoch 5/10	41s 45ms/step - accuracy: 0.7225 - loss: 0.7069 - val_accuracy: 0.7158 - val_loss: 0.7923
Epoch 6/10	29s 44ms/step - accuracy: 0.7826 - loss: 0.5574 - val_accuracy: 0.7590 - val_loss: 0.7115
Epoch 7/10	29s 44ms/step - accuracy: 0.8452 - loss: 0.4278 - val_accuracy: 0.7831 - val_loss: 0.6263
Epoch 8/10	29s 45ms/step - accuracy: 0.8669 - loss: 0.3566 - val_accuracy: 0.7888 - val_loss: 0.6471
Epoch 9/10	41s 44ms/step - accuracy: 0.8859 - loss: 0.3001 - val_accuracy: 0.7986 - val_loss: 0.6410
Epoch 10/10	41s 45ms/step - accuracy: 0.9103 - loss: 0.2428 - val_accuracy: 0.7895 - val_loss: 0.6786
Eğitim Süresi:	946.01 saniye

### II. Modeli Test Etme

Model, test kümesi üzerinde değerlendirildiğinde 2632 görüntüden oluşan test veri kümesine uygulanmıştır. Test işlemi sırasında aşağıdaki sonuçlar elde edilmiştir. Kayıp, 0.6798 olarak doğruluk ise 0.7884 olarak karşımıza çıkmaktadır. Bu sonuçlar, modelin test kümesi üzerindeki performansını göstermektedir. Test doğruluğunun %78.84 olması, modelin sınıflandırma görevinde başarılı bir şekilde çalıştığını göstermektedir. Çıkarım süresinin kısa olması, modelin gerçek zamanlı uygulamalarda kullanılabilirliği açısından önemli bir avantajdır.

Found 2632 images belonging to 5 classes.
/usr/local/lib/python3.10/dist-packages/keras/src/trainers/data_adapters/py_dataset_adapter.py:100: UserWarning: self._warn_if_super_not_called()
83/83 — 19s 187ms/step - accuracy: 0.7813 - loss: 0.7063
Test Loss: 0.6798163652420044
Test Accuracy: 0.7883738875389099
83/83 — 16s 170ms/step

### III. Test Kümesi Üzerinde Grafikler



Test kümesi için loss (kayıp) ve accuracy (doğruluk) grafiklerinde modelin performansına dair önemli bilgiler yer almaktadır. Loss grafiği, modelin hata oranının epoch sayısına bağlı olarak azaldığını göstermektedir. Bu durum, modelin her epoch ile daha iyi bir şekilde öğrenme gerçekleştirdiğini ifade eder.

Accuracy grafiği ise modelin doğruluk oranının epoch sayısı arttıkça yükseldiğini göstermektedir. Bu yükseliş, modelin test veri kümesi üzerinde sınıflandırmayı başarılı bir şekilde gerçekleştirdiğinin bir göstergesidir.

Grafiklerdeki eğilimler, modelin eğitim sürecinin dengeli ve başarılı bir şekilde tamamlandığını desteklemektedir.

### IV. Performans Metrikleri

Çıkarım Süresi: 2.18 saniye

**Sensitivity (Recall):** {0: 0.7751937984496124, 1: 0.7566539923954373, 2: 0.7966728280961183, 3: 0.8769230769230769, 4: 0.7429111531190926}

**Specificity:** {0: 0.947069943289225, 1: 0.9140550807217473, 2: 0.953610712577714, 3: 0.9635416666666666, 4: 0.9586305278174037}

Sınıflandırma Raporu:

	precision	recall	f1-score	support
0(KIFOZ)	0.78	0.78	0.78	516
1(PEKTUS E.)	0.69	0.76	0.72	526
2(PEKTUS K.)	0.82	0.80	0.81	541
3(POLAND S.)	0.86	0.88	0.87	520
4(SKOLYOZ)	0.82	0.74	0.78	529
accuracy			0.79	2632
macro avg	0.79	0.79	0.79	2632
weighted avg	0.79	0.79	0.79	2632

Modelin performansını değerlendirmek için aşağıdaki metrikler hesaplanmıştır:

**Accuracy (Doğruluk):** Modelin toplam doğru tahmin oranını gösterir. Test setinde doğruluk değeri %79 olarak ölçülmüştür. Bu, modelin genel başarısını ifade eder.

**Recall (Duyarlılık/Sensitivity):** Her bir sınıf için doğru pozitif tahmin oranını temsil eder. Örneğin, Kifoz sınıfında %77.5, Pektus Ekskavatum sınıfında %75.6, Pektus Karinatum sınıfında %79.7 gibi sonuçlar elde edilmiştir.

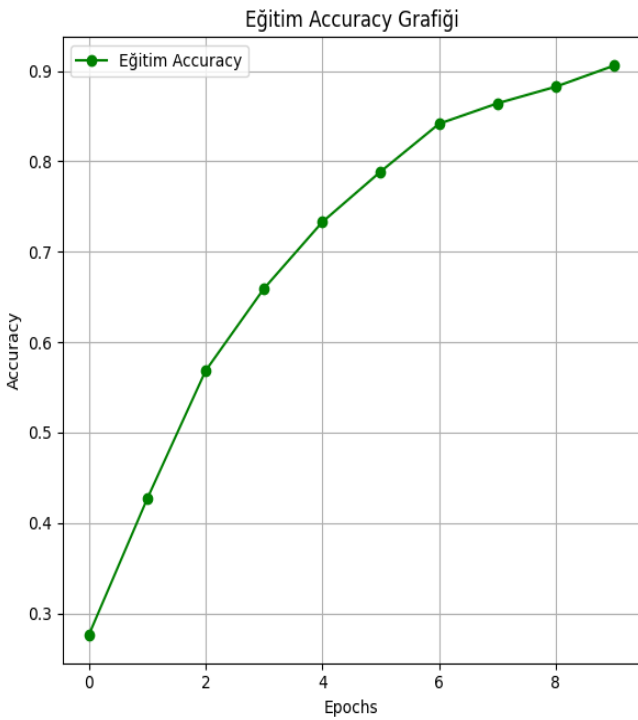
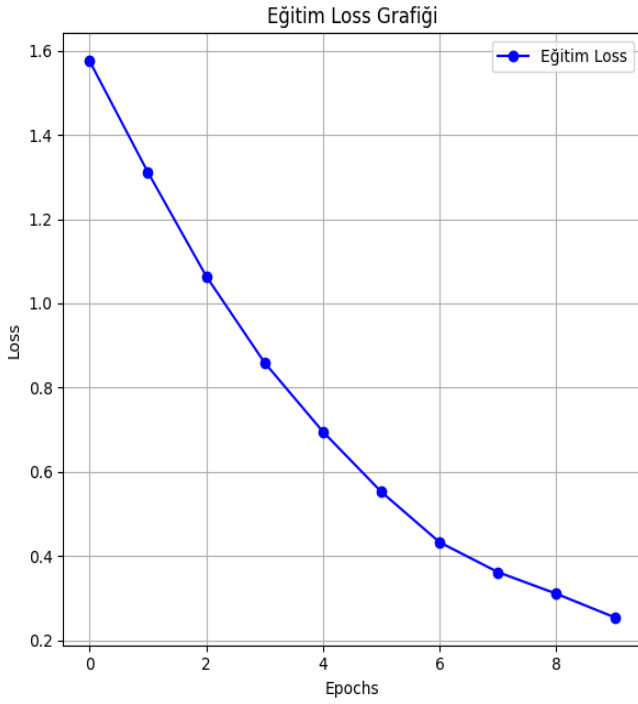
**Specificity (Özgüllük):** Yanlış pozitif oranını düşürme başarısını ölçer. Tüm sınıflar için yüksek özgüllük oranları (%94 ile %98 arasında) görülmüştür.

**Precision (Kesinlik):** Modelin tahmin ettiği pozitif örneklerin ne kadarının doğru olduğunu gösterir. Tüm sınıflarda %78 ile %86 arasında değerler elde edilmiştir.

**F1-Score:** Precision ve Recall'un dengesi alınarak hesaplanır. Dengeli bir model performansı ölçütü olarak, F1-Score değerleri %78 ile %87 arasında değişmektedir.

**Support (Destek):** Her bir sınıf için test setindeki örnek sayısını ifade eder. Örneğin, Kifoz sınıfında 516, Skolyoz sınıfında 529 görüntü bulunmaktadır.

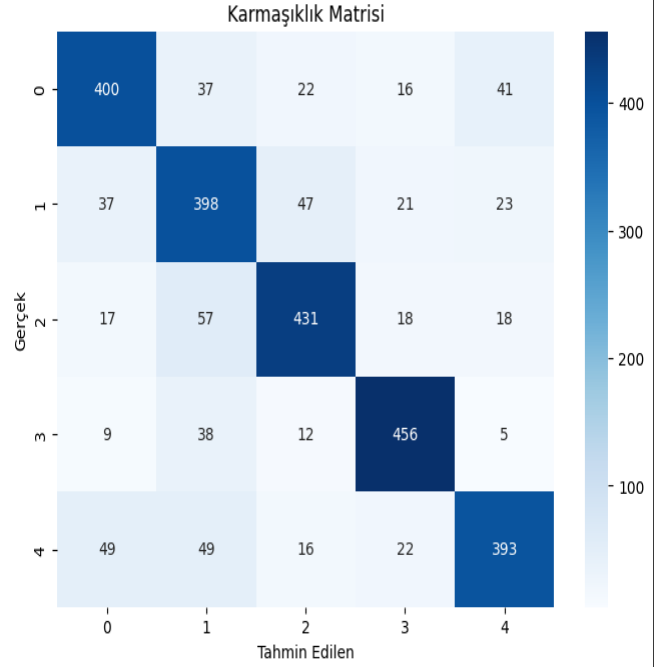
### V. Eğitim Kümesi Üzerinde Grafikler



Eğitim süreci boyunca modelin başarısını görselleştirmek için eğitim kaybı (loss) ve doğruluk (accuracy) grafiklerini inceledik. Eğitim kümesinde kayıp, modelin hatalarını ne kadar iyi öğrenebildiğini gösterirken, doğruluk ise modelin doğru tahmin yaptığı oranı temsil eder. Modelin ilk birkaç epoch'ta kaybı yüksek olup doğruluk değeri düşüktü; bu da modelin veriyi tam olarak öğrenemediğini gösteriyor. Ancak epoch sayısı arttıkça hem kayıp değeri düştü hem de doğruluk oranı artmaya başladı. Özellikle 10. epoch sonunda eğitim doğruluğu %91'e ulaşırken, doğruluk kaybı da önemli ölçüde azalmış oldu. Bu gelişim, modelin zamanla öğrenme kapasitesinin arttığını ve daha doğru tahminler

yapabileceğini göstermektedir. Validasyon setinde ise doğruluk %78.95 seviyelerine çıkmıştır, bu da modelin genelleme yeteneğinin oldukça iyi olduğunu ortaya koymaktadır.

## VI. Karmaşıklık Matrisi



Karmaşıklık matrisi, sınıflandırma modelinin tahmin sonuçlarını detaylı bir şekilde gösteren bir araçtır. Her satır gerçek sınıfı, her sütun ise modelin tahmin ettiği sınıfı temsil eder. Örneğin, ilk satırda (gerçek sınıf 0) modelin doğru tahmin ettiği 400 örnek (true positives, TP) bulunuyor. Diğer değerler ise modelin yanlış tahminlerini (false positives, FP ve false negatives, FN) gösteriyor.

Karmaşıklık matrisinde her bir hücre, modelin belirli bir sınıfa ait veriyi nasıl sınıflandırdığını gösterir. Bu matrisin okunması, modelin hangi sınıflarda daha iyi performans gösterdiğini veya hangi sınıflarda hatalar yaptığını anlamaya yardımcı olur. Örneğin, 0. sınıf için (kifoz), modelin çoğunlukla doğru tahminler yaptığı (400 doğru tahmin) görülmektedir, ancak 41 yanlış tahmin de yapılmıştır. Benzer şekilde, diğer sınıflarda da doğru tahminler yüksekken, bazı yanlış sınıflandırmalar da mevcuttur.

Karmaşıklık matrisi, modelin performansını değerlendirirken diğer metriklerle birlikte kullanıldığında, modelin güçlü ve zayıf yönlerini daha iyi anlamaya olanak sağlar.

## VII. ROC Eğrileri

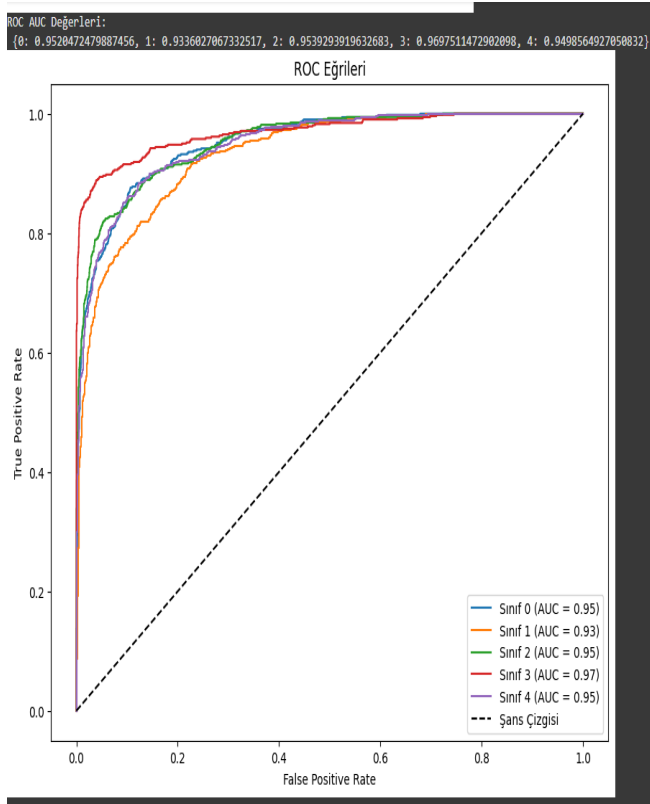
ROC (Receiver Operating Characteristic) eğrisi, sınıflandırma modelinin performansını görsel olarak değerlendirmeye yarayan bir araçtır. ROC eğrisinin yatay eksenini False Positive Rate (FPR), dikey eksenini ise True Positive Rate (TPR) değerini temsil eder. ROC eğrisinin ideal



konumu, sağ üst köşeye yakın olur, çünkü bu, modelin doğru tahminlerde yüksek bir doğruluk oranı gösterdiği anlamına gelir.

AUC (Area Under the Curve), ROC eğrisinin altındaki alanı ifade eder ve modelin genel performansını sayısal bir değerle ölçer. AUC değeri 0 ile 1 arasında değişir; 1'e yakın bir değer mükemmel bir modelin, 0.5'e yakın bir değer ise rasgele tahmin yapan bir modelin göstergesidir.

Çıktıda verilen AUC değerleri, her bir sınıf için modelin başarısını göstermektedir. Örneğin, sınıf 3 (ki bu pektus karinatum hastalığına karşılık geliyor) için AUC değeri 0.9697, modelin bu sınıfı çok doğru tahmin ettiğini gösteriyor. Diğer sınıflar için de AUC değerleri oldukça yüksek olup, modelin genel olarak başarılı bir performans sergilediğini ortaya koymaktadır. Yüksek AUC değerleri, modelin her sınıf için doğru tahmin yapma yeteneğini vurgular ve sınıflar arasındaki ayrımın güçlü olduğunu gösterir.

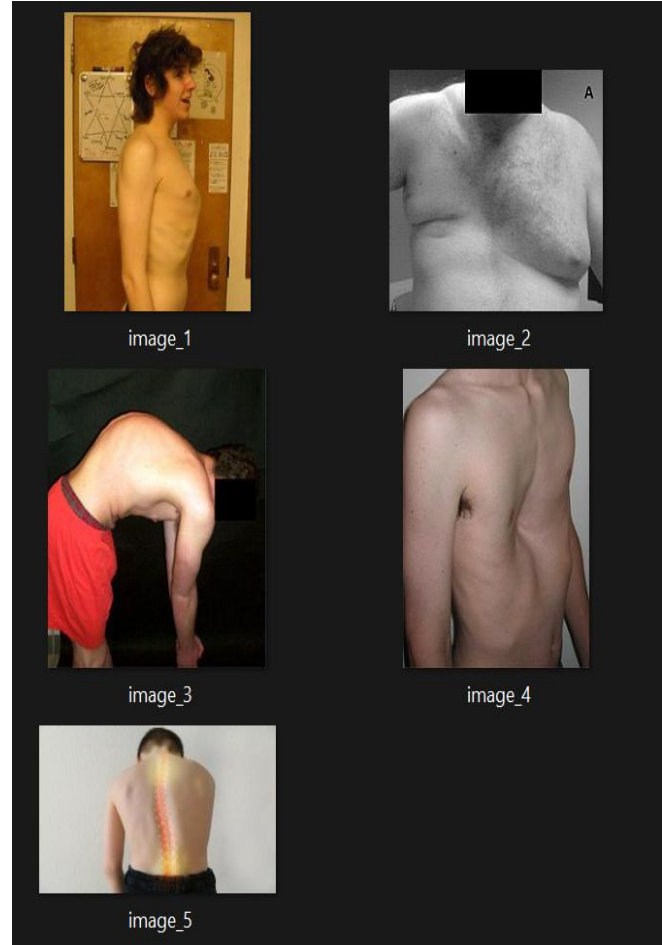


### VIII. Modeli Farklı Görüntüler ile Test Etme

Modelin test edilmesi süreci, modelin gerçek dünya verileriyle ne kadar iyi genellenebileceğini değerlendirmek için önemlidir. Verilen görseller üzerinde yapılan testlerde, her bir resim için modelin tahmin ettiği sınıf belirtilmiştir. Ancak, ilk görselde modelin yanlış tahmin yaptığı gözlemlenmiştir. Bu durum, modelin bazı durumlarda sınıflar arasında doğru ayrımı yapmada zorluk yaşayabileceğini gösterir.

Yanlış tahminler, modelin eğitim sürecindeki eksiklikler veya test verisinin modelin daha önce gördüğü verilerden farklı

olmasından kaynaklanabilir. Modelin doğruluğunu artırmak için, daha fazla eğitim verisi sağlamak, veri augmentasyonu yapmak veya modelin hiperparametrelerini optimize etmek gerekebilir. Bu tür testler, modelin geliştirilmesi için önemli geri bildirim sağlar.



1/1 — 0s 19ms/step  
1. Resim: Tahmin Edilen Sınıf: PEKTUS EKSKAVATUM  
1/1 — 0s 20ms/step  
2. Resim: Tahmin Edilen Sınıf: POLAND SENDROMU  
1/1 — 0s 22ms/step  
3. Resim: Tahmin Edilen Sınıf: KIFOZ  
1/1 — 0s 28ms/step  
4. Resim: Tahmin Edilen Sınıf: PEKTUS EKSKAVATUM  
1/1 — 0s 30ms/step  
5. Resim: Tahmin Edilen Sınıf: SKOLYOZ

### B. Shifted Window Transformer (Swin)

Swin Transformer, görüntü sınıflandırma ve diğer bilgisayarlı görü problemleri için geliştirilmiş modern bir derin öğrenme modelidir. Vision Transformer'dan (ViT)

farklı olarak, görüntüleri yerel pencerelere ayırarak işler ve bu pencereler arasında kaydırma (shift) yaparak daha iyi bir küresel bağlam öğrenimi sağlar. Bu yöntem, hesaplama verimliliğini artırırken yüksek performans sunar ve büyük veri kümelerinde üstün başarı sergiler.

## I. Model Eğitimi

```
Epoch 1/20 347s 938ms/step - accuracy: 0.2027 - loss: 2.2555 - val_accuracy: 0.2500 - val_loss: 1.5944
Epoch 2/20 185s 494ms/step - accuracy: 0.2439 - loss: 1.5881 - val_accuracy: 0.2917 - val_loss: 1.5726
Epoch 3/20 65s 492ms/step - accuracy: 0.2919 - loss: 1.5435 - val_accuracy: 0.2955 - val_loss: 1.5361
Epoch 4/20 83s 497ms/step - accuracy: 0.3112 - loss: 1.5221 - val_accuracy: 0.3523 - val_loss: 1.5181
Epoch 5/20 82s 498ms/step - accuracy: 0.3698 - loss: 1.4716 - val_accuracy: 0.3542 - val_loss: 1.4901
Epoch 6/20 66s 508ms/step - accuracy: 0.4081 - loss: 1.3931 - val_accuracy: 0.3220 - val_loss: 1.6241
Epoch 7/20 87s 537ms/step - accuracy: 0.4269 - loss: 1.3585 - val_accuracy: 0.3731 - val_loss: 1.4884
Epoch 8/20 66s 508ms/step - accuracy: 0.4655 - loss: 1.2821 - val_accuracy: 0.3920 - val_loss: 1.4477
Epoch 9/20 66s 497ms/step - accuracy: 0.4795 - loss: 1.2198 - val_accuracy: 0.4148 - val_loss: 1.4111
Epoch 10/20 88s 541ms/step - accuracy: 0.5225 - loss: 1.1213 - val_accuracy: 0.4413 - val_loss: 1.4515
Epoch 11/20 76s 497ms/step - accuracy: 0.6115 - loss: 0.9814 - val_accuracy: 0.4697 - val_loss: 1.3646
Epoch 12/20 66s 499ms/step - accuracy: 0.6436 - loss: 0.9099 - val_accuracy: 0.4375 - val_loss: 1.4947
Epoch 13/20 82s 497ms/step - accuracy: 0.6542 - loss: 0.8683 - val_accuracy: 0.4432 - val_loss: 1.5818
Epoch 14/20 81s 493ms/step - accuracy: 0.7111 - loss: 0.7448 - val_accuracy: 0.4318 - val_loss: 1.5327
```

Bu bölümde, modelin eğitim süreci boyunca doğruluk (accuracy) ve kayıp (loss) değerlerinin değişimi gözlemlenmiştir. İlk epoch'ta doğruluk %20.27 iken, ilerleyen epoch'larda modelin öğrenme başarısı artarak 14. epoch'ta doğruluk %71.11 seviyesine ulaşmıştır. Ancak, validasyon doğruluk değerlerinin (val\_accuracy) eğitim doğruluğuna oranla düşük olduğu ve zamanla dalgalanma gösterdiği fark edilmektedir.

## II. Modeli Test Etme

Modelin test kümesindeki performansı, doğruluk ve kayıp metrikleri ile değerlendirilmiştir. Ayrıca, gerçek ve tahmin edilen etiketlerin karşılaştırılması, modelin bazı sınıfları doğru tahmin etmede zorluk yaşadığını ortaya koymaktadır. Çıkarım süresi ise 5.62 saniye olarak hesaplanmıştır.

**Test Loss (Kayıp): 1.3646**

**Test Accuracy (Doğruluk): %46.97**

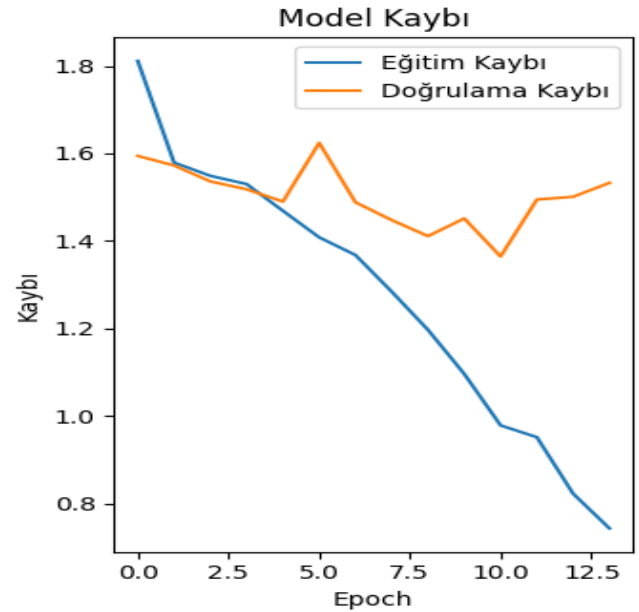
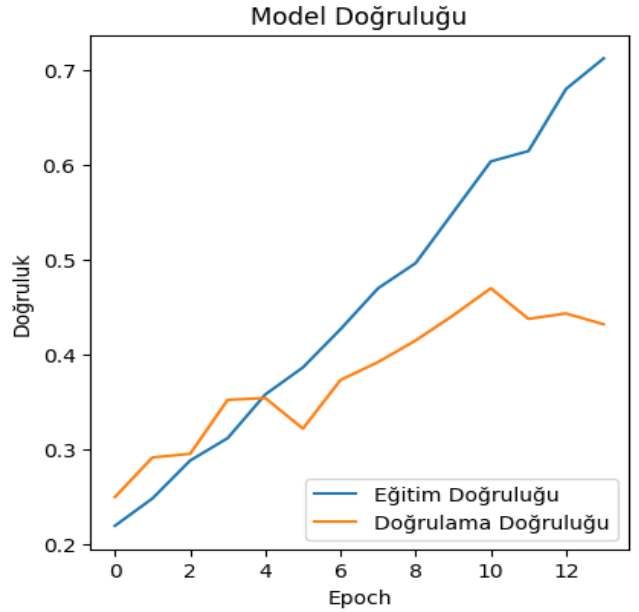
Bu sonuçlar, modelin test kümesi üzerindeki performansını göstermektedir.

```
33/33 6s 172ms/step - accuracy: 0.4648 - loss: 1.3519
Test doğruluğu: 0.4697, Test kaybı: 1.3646
1/1 0s 356ms/step
Gerçek Etiketler: tf.Tensor([3 2 0 2 1 1 1 1], shape=(10,), dtype=int32)
Tahmin Edilen Etiketler: tf.Tensor([3 1 1 2 2 4 1 1 0], shape=(10,), dtype=int64)
33/33 6s 164ms/step
Çıkarım Süresi: 5.62 saniye
```

## III. Grafikler

Test kümesi için loss (kayıp) ve accuracy (doğruluk) grafiklerinde modelin performansına dair önemli bilgiler yer almaktadır. Loss grafiği, modelin hata oranının epoch sayısına

bağlı olarak azaldığını göstermektedir. Bu durum, modelin her epoch ile daha iyi bir şekilde öğrenme gerçekleştirdiğini ifade eder. Accuracy grafiği ise modelin doğruluk oranının epoch sayısı arttıkça yükseldiğini göstermektedir.



## IV. Performans Metrikleri ve Karmaşıklık Matrisi

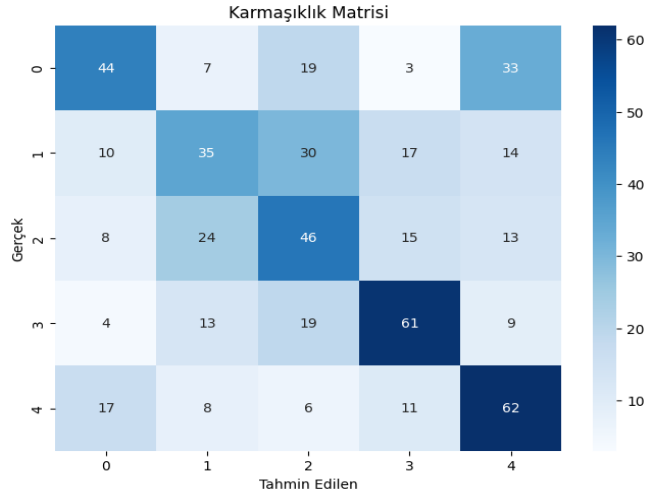
Karmaşıklık matrisi, modelin her sınıf için doğru ve yanlış tahminlerini göstermektedir. Sınıflandırma raporunda, sınıflar arasında performans farklılıkları olduğu görülmektedir. Özellikle sınıf 3 ve sınıf 4 için recall değerlerinin sırasıyla %57.5 ve %59.6 olması, bu sınıflarda nispeten daha başarılı bir duyarlılık oranına işaret etmektedir. Precision değerleri de farklılık göstermekte olup, modelin bazı sınıfları tahmin etmekte zorlandığını göstermektedir. Aşağıdaki şekilde performans metriklerinin daha detaylı hâli yer almaktadır. Aynı zamanda eğitim sonrasında ortaya çıkan karmaşıklık matrisi aşağıda verilmiştir.

Sensitivity (Recall): {0: 0.41509433962264153, 1: 0.330188679245283, 2: 0.4339622641509434, 3: 0.5754716981132075, 4: 0.5961538461538461}

Specificity: {0: 0.9075829383886256, 1: 0.8767772511848341, 2: 0.8246445497630331, 3: 0.8909952606635071, 4: 0.8372641509433962}

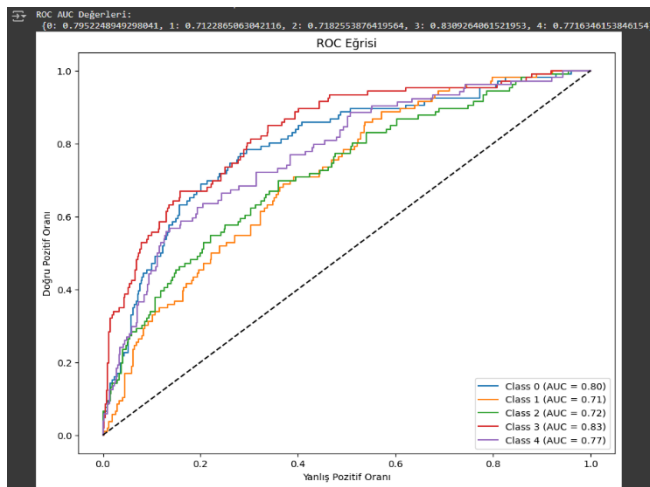
Sınıflandırma Raporu:

	precision	recall	f1-score	support
0	0.53	0.42	0.47	106
1	0.40	0.33	0.36	106
2	0.38	0.43	0.41	106
3	0.57	0.58	0.57	106
4	0.47	0.60	0.53	104
accuracy			0.47	528
macro avg	0.47	0.47	0.47	528
weighted avg	0.47	0.47	0.47	528



## V. ROC Eğrileri

ROC AUC değerleri, modelin her sınıf için ayırma kapasitesini ölçmektedir. Sınıf 3 için en yüksek ROC AUC değeri %83.09 olarak ölçülmüş, bu da modelin bu sınıfı diğerlerinden ayırt etmede nispeten daha başarılı olduğunu göstermektedir. En düşük ROC AUC değeri ise sınıf 1 için %71.22 olarak hesaplanmıştır.



## VI. Modeli Farklı Görüntüler ile Test Etme

Farklı görseller üzerinde yapılan testlerde, model her bir görsel için tahmin edilen sınıfları vermiştir. Bu testler,

modelin çeşitli görseller üzerinde sınıflandırma yeteneğini gözlemlemek için kullanılmıştır. Tahmin yapılan görüntüler ViT modelinde kullanılan görüntüler ile aynıdır. Model beş resim üzerinden iki tanesini doğru tahmin etmiştir. Sonuç aşağıdaki gibidir.

```
1/1 ————— 0s 234ms/step
1. Resim: Tahmin Edilen Sınıf: skolyoz
1/1 ————— 0s 31ms/step
2. Resim: Tahmin Edilen Sınıf: skolyoz
1/1 ————— 0s 30ms/step
3. Resim: Tahmin Edilen Sınıf: poland_sendromu
1/1 ————— 0s 31ms/step
4. Resim: Tahmin Edilen Sınıf: pektus_ekskavatum
1/1 ————— 0s 32ms/step
5. Resim: Tahmin Edilen Sınıf: skolyoz
```

## C. BERT Pre-Training of Image Transformers (BEiT)

BEiT (Bidirectional Encoder representation from Image Transformers), görsel görevler için tasarlanmış bir transformatör modelidir. BEiT, doğal dil işleme alanındaki BERT modeline benzer şekilde, görsellerin maskelenmiş bölümlerini tahmin ederek ön eğitim gerçekleştirir. Görselleri parçalara (patch) ayırır ve bu parçalar arasındaki ilişkileri öğrenmek için dikkat mekanizmasını (attention) kullanır. BEiT, özellikle görüntü sınıflandırma, nesne tespiti ve segmentasyon gibi görevlerde yüksek performans göstermesiyle öne çıkar.

### I. Model Eğitimi

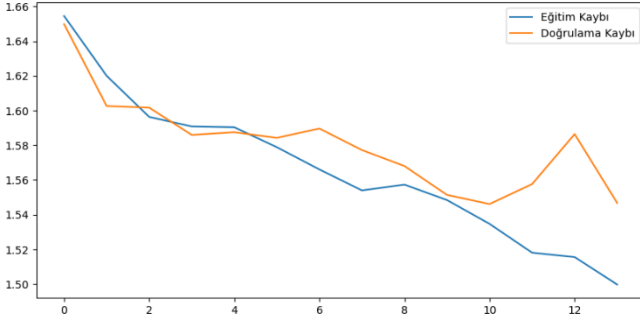
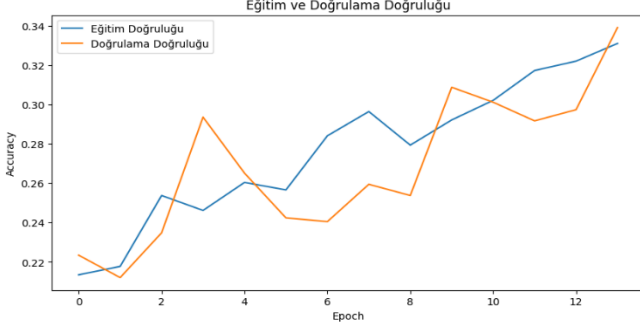
Model, 20 epoch boyunca eğitilmiştir. Başlangıçta eğitim doğruluğu (%20.67) ve doğrulama doğruluğu (%22.35) oldukça düşüktür. Eğitim sürecinin ilerleyen safhalarında model doğruluğu artış göstermiş ancak doğrulama doğruluğu dalgalanmıştır.

14. epoch sonunda doğrulama doğruluğu %33.90'a ulaşmıştır. Eğitim kaybı sürekli düşüş göstermiştir; ancak doğrulama kaybında tam bir iyileşme gözlenmemiştir, bu da modelin doğrulama kümesinde genelleme yapma yeteneğinin sınırlı olduğunu gösterir. Toplam eğitim süresi 1788.80 saniyedir.

```
Epoch 1/20
132/132 ————— 145s 798ms/step - accuracy: 0.2067 - loss: 1.7041 - val_accuracy: 0.2235 - val_loss: 1.6498
Epoch 2/20
132/132 ————— 102s 773ms/step - accuracy: 0.2097 - loss: 1.6268 - val_accuracy: 0.2121 - val_loss: 1.6026
Epoch 3/20
132/132 ————— 108s 820ms/step - accuracy: 0.2678 - loss: 1.5923 - val_accuracy: 0.2348 - val_loss: 1.6017
Epoch 4/20
132/132 ————— 139s 801ms/step - accuracy: 0.2449 - loss: 1.5952 - val_accuracy: 0.2936 - val_loss: 1.5859
Epoch 5/20
132/132 ————— 147s 834ms/step - accuracy: 0.2567 - loss: 1.5912 - val_accuracy: 0.2652 - val_loss: 1.5875
Epoch 6/20
132/132 ————— 110s 834ms/step - accuracy: 0.2629 - loss: 1.5719 - val_accuracy: 0.2474 - val_loss: 1.5843
Epoch 7/20
132/132 ————— 143s 837ms/step - accuracy: 0.3086 - loss: 1.5495 - val_accuracy: 0.2405 - val_loss: 1.5896
Epoch 8/20
132/132 ————— 109s 830ms/step - accuracy: 0.2979 - loss: 1.5456 - val_accuracy: 0.2595 - val_loss: 1.5772
Epoch 9/20
132/132 ————— 145s 853ms/step - accuracy: 0.2750 - loss: 1.5539 - val_accuracy: 0.2538 - val_loss: 1.5680
Epoch 10/20
132/132 ————— 154s 949ms/step - accuracy: 0.2902 - loss: 1.5563 - val_accuracy: 0.3087 - val_loss: 1.5514
Epoch 11/20
132/132 ————— 117s 888ms/step - accuracy: 0.3052 - loss: 1.5316 - val_accuracy: 0.3011 - val_loss: 1.5461
Epoch 12/20
132/132 ————— 115s 870ms/step - accuracy: 0.3420 - loss: 1.5069 - val_accuracy: 0.2917 - val_loss: 1.5576
Epoch 13/20
132/132 ————— 111s 845ms/step - accuracy: 0.3124 - loss: 1.5190 - val_accuracy: 0.2973 - val_loss: 1.5864
Epoch 14/20
132/132 ————— 143s 853ms/step - accuracy: 0.3247 - loss: 1.5141 - val_accuracy: 0.3390 - val_loss: 1.5467
Eğitim Zamanı: 1788.80 saniye
```

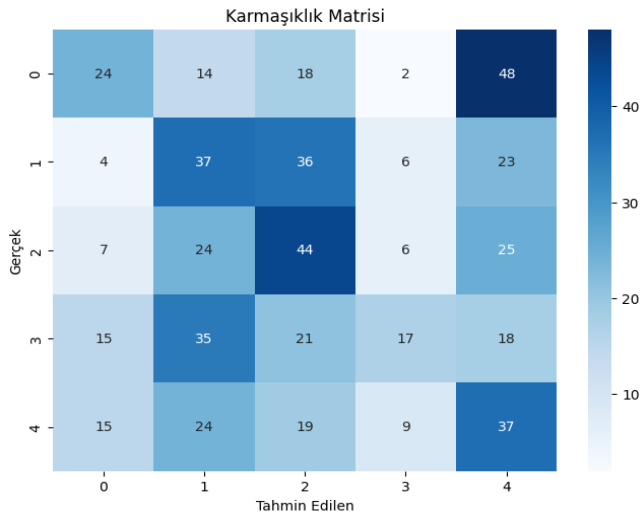
## II. Grafikler

Eğitim doğruluğu zamanla artış gösterirken kayıp değeri azalmaktadır. Ancak doğrulama doğruluğu ve kaybı arasında belirgin bir dalgalanma mevcuttur. Bu durum, modelin aşırı öğrenme (overfitting) riski taşıdığını veya doğrulama verilerinde uygun performansı sağlayamadığını gösterebilir.



## III. Performans Metrikleri ve Karmaşıklık Matrisi

Karmaşıklık Matrisi, modelin her sınıf için doğru ve yanlış tahminlerini gösterir. Örneğin, Sınıf 0'dan 24 örnek doğru tahmin edilirken 48 tanesi Sınıf 4 olarak yanlış tahmin edilmiştir. Duyarlılık (Sensitivity): Sınıf 2 (%41.51) en yüksek, Sınıf 3 (%16.03) en düşük. Özgüllük (Specificity): Sınıf 0 (%90.28) en yüksek doğruluğa sahiptir. Genel Performans: Doğruluk %30, en iyi F1 skoru (%36) Sınıf 2'ye aittir. Model sınıflar arasında dengeli performans göstermemiştir. Daha fazlası aşağıdaki görsellerdedir:



Sensitivity (Recall): {0: 0.22641509433962265, 1: 0.3490566037735849, 2: 0.41509433962264153, 3: 0.16037735849056603, 4: 0.3557692307692308}

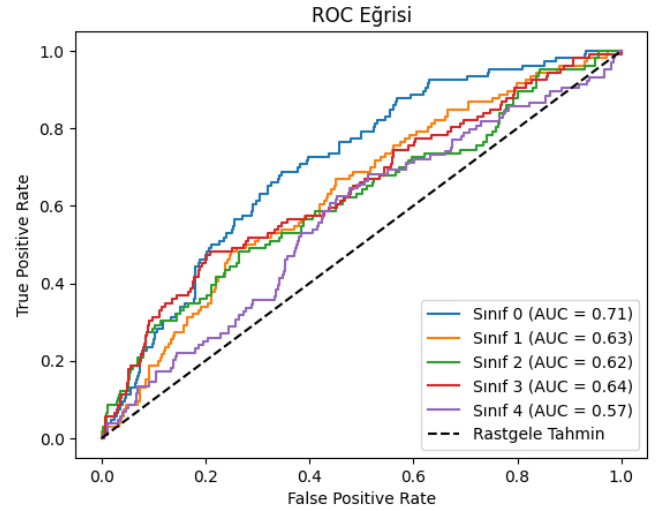
Specificity: {0: 0.9028436018957346, 1: 0.7701421800947867, 2: 0.7772511848341233, 3: 0.9454976303317536, 4: 0.7311320754716981}

Sınıflandırma Raporu:

	precision	recall	f1-score	support
0	0.37	0.23	0.28	106
1	0.28	0.35	0.31	106
2	0.32	0.42	0.36	106
3	0.42	0.16	0.23	106
4	0.25	0.36	0.29	104
accuracy			0.30	528
macro avg	0.33	0.30	0.29	528
weighted avg	0.33	0.30	0.29	528

## IV. ROC Eğrileri

ROC eğrisi, sınıflar için ayırım gücünü ölçer. AUC (Eğri Altındaki Alan) değerleri, sınıf ayrıştırma performansını gösterir. Sınıf 0 AUC=0.71 ile en iyi ayırım gücüne sahip. Sınıf 4 AUC=0.57 ile en zayıf ayırım gücüne sahip. Bu sonuçlar, modelin bazı sınıflar arasında ayrımı iyi yapamadığını işaret eder.



## V. Modeli Farklı Görüntüler ile Test Etme

ViT model testinde de kullandığımız görüntüler ile BeiT modelini test ettik. Beş görüntüden ikisini doğru bildi. Sonuçlar modelin sınıfları ayırtma zorluk çektiğini ve daha fazla optimizasyon gerektiğini gösteriyor. Sonuçlar aşağıdaki gibidir:

```
1/1 ————— 0s 55ms/step
1. Resim: Tahmin Edilen Sınıf: pektus_karinatum
1/1 ————— 0s 38ms/step
2. Resim: Tahmin Edilen Sınıf: skolyoz
1/1 ————— 0s 35ms/step
3. Resim: Tahmin Edilen Sınıf: pektus_karinatum
1/1 ————— 0s 100ms/step
4. Resim: Tahmin Edilen Sınıf: pektus_ekskavatum
1/1 ————— 0s 55ms/step
5. Resim: Tahmin Edilen Sınıf: kifo
```



## KAYNAKÇA

- [1] [Selenium WebDriver with Python Tutorial - javatpoint](#)
  - [2] <https://www.youtube.com/watch?v=MM10SYKfqwA&list=PLbFzTYWXNIJ4BrHcQ5dVhy9xhLtLg2Kcr&index=1>
  - [3] <https://www.youtube.com/watch?v=1OlakgRTF4E>
  - [4] [https://www.tutorialspoint.com/python\\_pillow/index.htm](https://www.tutorialspoint.com/python_pillow/index.htm)
  - [5] <https://github.com/nikhilroxtomar/Flower-Image-Classification-using-Vision-Transformer>
  - [6] <https://github.com/nikhilroxtomar/Vision-Transformer-ViT-in-TensorFlow>
  - [7] [https://keras.io/examples/vision/image\\_classification\\_with\\_vision\\_transformer/](https://keras.io/examples/vision/image_classification_with_vision_transformer/)
  - [8] <https://www.tensorflow.org/>
  - [9] [https://huggingface.co/docs/transformers/model\\_doc/vit](https://huggingface.co/docs/transformers/model_doc/vit)
  - [10] <https://www.youtube.com/watch?v=Fb1xsTXT4P8&t=25s>
  - [11] <https://www.youtube.com/watch?v=Ssndsjh1Zqk&list=PL6EUEMk2A71ZRh88Fe9u4WbvaQEAkofAw>
  - [12] [https://www.youtube.com/watch?v=i2\\_zJ0ANrw0](https://www.youtube.com/watch?v=i2_zJ0ANrw0)
- 
- [Veri Seti \(25.677 Veri\)](#)
  - [Eğitim&Test Ayırımı \(Google Colab Notebook\)](#)
  - [ViT Model Eğitimi \(Google Colab Notebook\)](#)
  - [Swin Model Eğitimi \(Google Colab Notebook\)](#)
  - [BeiT Model Eğitimi \(Google Colab Notebook\)](#)
  - [GitHub Linki I](#)
  - [GitHub Linki II](#)