

Yazılım Laboratuvarı 2 Proje-3

1. Zehra Karabektaş
Bilgisayar Mühendisliği
Kocaeli Üniversitesi
Kocaeli, Türkiye

zhrakarabektas679@gmail.com

Abstract— Bu çalışmada, MultiZoo görüntü veri seti kullanılarak hayvan türlerini sınıflandırmaya yönelik transformer tabanlı bir görüntü sınıflandırma modeli geliştirilmiştir. Modelin doğruluğunu artırmak amacıyla yeniden boyutlandırma, normalizasyon, gürültü giderme, histogram eşitleme ve çeşitli veri artırma (data augmentation) teknikleri uygulanmıştır. Vision Transformer (ViT), Data-efficient Image Transformer (DeiT), Swin ve Beit gibi farklı transformer mimarileri çeşitli hiperparametre ayarları ile eğitilmiştir. Modeller, doğruluk (accuracy), kesinlik (precision), duyarlılık (recall) ve F1-skoru metrikleri kullanılarak değerlendirilmiştir. Deneysel karşılaştırmalar sonucunda en yüksek doğruluk değerine ulaşan model, masaüstü bir arayüz uygulamasına entegre edilerek kullanıcıların test görsellerini yükleyip tahmin sonuçlarını ve güven skorlarını anlık olarak görüntüleyebileceği bir sistem tasarlanmıştır.

Keywords — Görüntü Sınıflandırma; Transformer; Data Augmentation; MultiZoo Dataset; Vision Transformer; Masaüstü Uygulaması; Beit.

I. GİRİŞ

Günümüzde yapay zeka ve derin öğrenme alanındaki gelişmeler sayesinde görüntü işleme problemlerine daha başarılı çözümler geliştirilebilmektedir. Bu projede, görüntü sınıflandırma konusuna odaklanılarak farklı hayvan türlerini ayırt edebilen bir model geliştirilmesi amaçlanmıştır. Projenin amacı, bize verilen MultiZoo adlı görüntü veri seti ile çalışarak farklı hayvan türlerini tanıyabilen bir yapay zeka modeli oluşturmaktır. Bu amaçla transformer tabanlı modeller kullanılmıştır. Özellikle Vision Transformer (ViT) Deit, Swin ve Beit gibi transformer tabanlı modeller bu projede değerlendirilmiştir.

Modelin başarısını artırmak için eğitim verilerine çeşitli ön işlemler uygulanmıştır. Bu işlemler arasında görüntülerin yeniden boyutlandırılması, normalizasyon, gürültü giderme, histogram eşitleme ve veri artırma (data augmentation) gibi yöntemler yer almaktadır. Geliştirilen model, doğruluk (accuracy), kesinlik (precision), duyarlılık (recall) ve F1-skoru gibi metriklerle değerlendirilmiştir.

Son olarak, en başarılı sonuç veren model, kullanıcıların test görsellerini yükleyerek anlık olarak tahmin alabilecekleri kullanıcı dostu bir masaüstü arayüz uygulamasına entegre edilmiştir.

II. YÖNTEM

A. Kullanılan Ortam ve Araçlar

Model eğitimi, GPU desteği sayesinde daha hızlı ve verimli bir şekilde gerçekleştirilebilmesi amacıyla Google Colab ortamında yapılmıştır. Kodların düzenlenmesi ve takip

edilmesi için aynı zamanda Visual Studio Code editörü kullanılmıştır.

Projede Python programlama dili tercih edilmiş, derin öğrenme işlemleri için popüler bir framework olan PyTorch kütüphanesi kullanılmıştır.

Eğitilen modelin görsel testini gerçekleştirmek amacıyla geliştirilen masaüstü arayüz, PyQt5 kütüphanesi ile oluşturulmuştur. Bu arayüz uygulaması, yerel ortamda Visual Studio Code üzerinden geliştirilmiş ve çalıştırılmıştır.

B. Veri Seti

Bu projede kullanılan veri seti, Google Drive üzerinden sağlanan MultiZoo görüntü veri setidir. Veri seti, 90 farklı hayvan türünü temsil eden 90 klasörden oluşmaktadır ve her klasörde yaklaşık 53 adet görüntü yer almaktadır. Bu veri seti, çoklu sınıflandırma problemlerine uygun şekilde tasarlanmıştır. Modelin eğitilmesi sürecinde, veri seti ikiye ayrılarak %80'i eğitim (train), %20'si doğrulama (validation) verisi olarak kullanılmıştır. Test verisi ise eğitim süreci boyunca kullanılmamış, yalnızca modelin son performansını ölçmek amacıyla değerlendirme aşamasında kullanılmıştır.

Veri seti ayrımı işlemi Python ortamında gerçekleştirilmiştir. Dosya ve klasör yönetimi için os ve shutil kütüphaneleri, veri bölme işlemi için ise scikit-learn kütüphanesinden train_test_split fonksiyonu kullanılmıştır. Her bir sınıfa ait görseller, orantılı biçimde eğitim (%80) ve doğrulama (%20) alt klasörlerine dağıtılarak dengeli bir veri yapısı oluşturulmuştur.

```
# Eğitim ve doğrulama klasörleri oluşturulur
klasör_olustur("train/")
klasör_olustur("val/")

# Tüm sınıflar için tekrar eden işlem
for her_sınıf in sınıf_listesi("drivedataset/train/"):

    # Sınıfa ait görseller listelenir
    görseller = listele(her_sınıf)

    # %80 eğitim, %20 doğrulama olacak şekilde ayrılır
    eğitim_görselleri, val_görselleri = verileri_ayır(görseller, oran=0.2)

    # Eğitim görselleri uygun klasöre kopyalanır
    for görsel in eğitim_görselleri:
        kopyala(görsel, "train/" + her_sınıf)

    # Doğrulama görselleri uygun klasöre kopyalanır
    for görsel in val_görselleri:
        kopyala(görsel, "val/" + her_sınıf)
```

Veri Setinin Eğitim ve Doğrulama Olarak Ayırma Kaba Kodu

C. Veri Seti Ön İşleme Adımları

Masaüstü uygulamasında kullanılan sınıflandırma modelinin eğitimi öncesinde, görüntü verileri çeşitli ön işleme adımlarından geçirilmiştir. Bu işlemler Python ortamında, OpenCV, PIL (Pillow) ve Albumentations kütüphaneleri kullanılarak gerçekleştirilmiştir.

1) *Boyutlandırma (resize)*: Model giriş boyutlarına uygunluk sağlamak amacıyla tüm görüntüler 224x224 piksel boyutuna yeniden ölçeklendirilmiştir. Bu işlem, sadece klasik resize ile sınırlı kalmamış, aynı zamanda kenar detaylarını korumaya yönelik Laplacian filtresi uygulanarak gerçekleştirilmiştir. Bu yöntem ile görselin hem boyutu küçültülmüş hem de kenar-doku bilgisi koruma amaçlanmıştır.

2) *Normalizasyon*: Tüm görüntü verileri, modelin daha verimli ve dengeli öğrenebilmesi amacıyla normalize edilmiştir. RGB kanallar için ortalama ve standart sapma değerleri aşağıdaki gibi alınmış ve tüm pikseller $[-1, 1]$ aralığına dönüştürülmüştür. Bu işlem, hem Albumentations içinde eğitim verilerine hem de torchvision.transforms içinde doğrulama verilerine uygulanmıştır.

3) *Görüntü işleme teknikleri*: Modelin daha sağlam ve genelleyici temsiller öğrenebilmesi amacıyla çeşitli işlemler uygulanmıştır. Öncelikle, görsellerdeki yapısal detayların korunması ve vurgulanması için Laplacian filtresi kullanılarak kenar belirginleştirme (keskinleştirme) gerçekleştirilmiştir. Bunun yanı sıra, hareket bulanıklığını simüle eden MotionBlur tekniği ile birlikte MedianBlur ve GaussianBlur gibi filtreleme yöntemleri uygulanarak görüntülerdeki parazitler azaltılmış, bulanıklık gibi bozulmalar modele öğretilmiştir. Bu işlemler sayesinde model, hem yüksek kaliteli hem de bozulmuş görseller üzerinde daha güçlü bir sınıflandırma performansı göstermeye uygun hale getirilmiştir.

4) *Veri Arttırma (Data Augmentation)*: Eğitim verisinin çeşitliliğini artırmak ve modelin aşırı ezberleme (overfitting) yapmasını önlemek amacıyla Albumentations kütüphanesi ile augmentasyon teknikleri uygulanmıştır. Kullanılan temel augmentasyonlar şunlardır:

- Yatay çevirme (HorizontalFlip): %50 olasılıkla görsel yatay ekseninde simetrik olarak çevrilerek modelin yön bağımsız öğrenmesini sağlamak
- Rastgele kırpma ve yeniden ölçekleme (RandomResizedCrop): %100 olasılıkla uygulanmış olup görselin rastgele bir bölgesi kırılarak yeniden 224x224 boyutuna ölçeklenir. Modelin farklı alanlara odaklanmasını ve konum varyasyonlarına karşı daha dayanıklı olması sağlanmıştır.
- Renk jitterlama (ColorJitter): Parlaklık, kontrast, doygunluk ve renk tonu (hue) değerleri rastgele değiştirilir. Farklı ışık koşullarında bile doğru genelleme yapabilen bir model eğitilmesini sağlanmıştır.

- Döndürme (Rotate): Görseli maksimum $\pm 15^\circ$ arasında rastgele döndürülerek hayvanların farklı açılardan tanınması sağlanmak istendi.
- Gölge ve kontrast: Görsel üzerine yapay gölge eklenebilir ve parlaklık/kontrast rastgele değiştirilir. Değişen aydınlatma koşullarında sınıflandırma başarısını artırır.
- Bulanıklık ve Gürültü Etkileri (OneOf - MotionBlur / MedianBlur / GaussianBlur): %30 olasılıkla bir tanesi seçilir ve uygulanır. Hareketli ya da düşük kaliteli görsellere karşı dayanıklılık kazandırır.

Bu çalışmada, tüm sınıflar başlangıçta eşit sayıda örnek içerdiğinden, veri artırma işlemleri her örneğe rastgele augmentasyon uygulanacak şekilde gerçekleştirilmiştir. Bu yöntem, sınıflar arasındaki örnek sayısını değiştirmemekle birlikte, modelin genelleme yeteneğini artırmayı hedeflemiştir. Uygulanan tüm bu teknikler, modelin farklı koşullarda (gölge, ışık, pozisyon değişimi) doğru genellemeler yapmasına katkı sağlamıştır. Bu ön işleme adımları, hem veri setinin zenginleştirilmesini hem de modelin daha güçlü ve dayanıklı hale gelmesini sağlamıştır.

D. Model Mimarisi ve Eğitim Süreci

Bu projede, BEiT (Bidirectional Encoder representation from Image Transformers) mimarisi kullanılmıştır. Model, timm kütüphanesi üzerinden hazır ağırlıklarla çağrılmış ve MultiZoo veri seti üzerinde transfer learning ile eğitilmiştir. Eğitim sürecinde AdamW optimizasyonu, çapraz entropi kaybı (CrossEntropyLoss) ve erken durdurma mekanizması kullanılmıştır. Eğitim boyunca doğruluk (accuracy), loss, precision, recall ve F1-score metrikleri izlenmiştir.

- Kullanılan Model Adı: beit_base_patch16_224
- Görüntü Boyutu: 224x224
- Optimizasyon Yöntemi: AdamW
- Kayıp Fonksiyonu: CrossEntropyLoss
- Öğrenme Oranı (Learning Rate): $1e-5$
- Batch Size: 32
- Epoch Sayısı: 50 (Early Stopping ile 27. epochta durmuştur)
- Erken Durdurma: Patience = 2 (validation loss iyileşmediğinde eğitim durur)
- Eğitim / Doğrulama Ayrımı: %80 / %20 oranında

E. Arayüz Uygulaması

Modelin test verileri ile gerçek zamanlı olarak değerlendirilmesi amacıyla, PyQt5 kütüphanesi kullanılarak bir masaüstü arayüz uygulaması geliştirilmiştir. Bu uygulama sayesinde kullanıcı, bir görsel dosyası yükleyerek modelin tahmin ettiği hayvan türünü ve ilgili güven skorunu anlık olarak görüntüleyebilmektedir.

Kullanıcı, “Görüntü Yükle” butonuna tıklayarak bir görsel seçer. Seçilen görsel, modelin kabul ettiği boyut olan 224x224 piksele dönüştürülerek işlenir. Ardından eğitilmiş modele gönderilir ve tahmin edilen sınıf ismi ile doğruluk oranı arayüzde kullanıcıya gösterilir. Kullanıcı, “Sıfırla” butonu ile uygulamayı başlangıç haline döndürebilir.

Geliştirilen masaüstü arayüz, kullanıcı dostu ve sade bir tasarıma sahiptir. Butonlar, ikonlar, arka plan ve yazı stili gibi görsel öğeler ile arayüz estetik ve işlevsel bir yapıya kavuşturulmuştur.

```
# Arayüz başlatılır
uygulamayı_çalıştır()

# Kullanıcı "Görüntü Yükle" butonuna tıkladığında:
görsel_yolu = kullanıcıdan_görsel_seç()

# Görsel yeniden boyutlandırılır ve tensöre dönüştürülür
görsel = yükle(görsel_yolu)
görsel = boyutlandır(görsel, 224, 224)
görsel = tensöre_dönüştür(görsel)

# Eğitilmiş model yüklenir
model = modeli_yükle("model.pth")

# Tahmin yapılır
tahmin_sınıfı, güven_skoru = model.çalıştır(görsel)

# Sonuç arayüzde gösterilir
tahmini_göster(tahmin_sınıfı, güven_skoru)

# Kullanıcı "Sıfırla" butonuna basarsa:
arayüzü_sıfırla()
```

Masaüstü Arayüzde Görsel Yükleme ve Tahmin Sürecinin Kaba Kodu

III. DENEYSEL SONUÇLAR

Bu çalışmada, hayvan türü sınıflandırma problemi için transformer tabanlı çeşitli derin öğrenme modelleri test edilmiş ve farklı eğitim parametrelerinin model başarımı üzerindeki etkileri incelenmiştir. DeneySEL süreç üç aşamada yürütülmüştür: (1) DEiT modeli üzerinde hiperparametre ayarlamaları, (2) farklı transformer mimarilerinin karşılaştırılması, ve (3) gelişmiş veri artırma (augmentasyon) ve düzenleme (regularizasyon) teknikleri ile nihai modelin performansının iyileştirilmesi.

A. Deit modeli üzerinde hipermetre denemesi

Bu çalışmada, transformer tabanlı deit_base_patch16_224 modeli, farklı öğrenme oranları ve batch size değerleriyle eğitilmiş ve doğrulama seti üzerindeki sınıflandırma performansı detaylı olarak incelenmiştir. Eğitim süreci maksimum 50 epoch olarak sınırlandırılmış, doğrulama kaybı 3 ardışık epoch boyunca iyileşmediğinde early stopping uygulanmıştır. Eğitim verileri üzerinde uygulanan ön işlemler, yalnızca temel görüntü işleme adımları ile sınırlandırılmıştır. Bu kapsamda görseller, ilk olarak 224×224 piksel boyutuna yeniden boyutlandırılmış; ardından normalizasyon, yatay çevirme ve renk jitterlama gibi temel görüntü artırma (data augmentation) işlemleri uygulanmıştır. Her denemenin sonunda model doğrulama verisi üzerinde test edilmiş ve sınıf bazlı başarı metrikleri elde edilmiştir.

Aşağıdaki tabloda denenen konfigürasyonlar ve bu konfigürasyonlara ait genel performans metrikleri özetlenmiştir:

TABLO I. DEiT MODEL DENEMELERİ

	Batch Size	Öğrenme Oranı(LR)	DOĞRULUK (ACCURACY)	Presician	F1-Score	Recall
1. Model	32	1e-5	%93.0	%94.0	%93.0	%93.0
2. Model	64	1e-5	%94.0	%94.0	%94.0	%94.0
3. Model	32	2e-5	%96.0	%96.0	%96.0	%96.0

a. Model eğitim sonuçları

Üç farklı hiperparametre kombinasyonu ile gerçekleştirilen eğitim denemeleri sonucunda, birinci deneme (batch size: 32, learning rate: 1e-5) %93 doğruluk ve macro F1-score ile sınıflar arası en dengeli ve kararlı performansı göstermiştir. İkinci deneme (batch size: 64, learning rate: 1e-5) doğruluğu %94'e çıkarsa da bazı sınıflarda düşen recall değerleri nedeniyle genel tutarlılığı zayıflamıştır. Üçüncü deneme (batch size: 32, learning rate: 2e-5) %96 doğrulukla en yüksek başarıyı elde etmiş; ancak sınıf bazlı istikrarsızlık ve precision-recall dengesizlikleri gözlemlenmiştir.

B. Farklı modeller ile eğitim denemeleri

Farklı modeller ile eğitim denemeleri gerçekleştirilmiştir. Bu süreçte Vision Transformer (ViT), Data-efficient Image Transformer (DEiT), Swin Transformer ve BEiT mimarileri aynı veri seti üzerinde, benzer hiperparametre ayarlarıyla eğitilerek adil bir karşılaştırma ortamı sağlanmıştır.

Modeller, timm kütüphanesi aracılığıyla önceden eğitilmiş (pretrained) ağırlıklarla yüklenmiş; son katman, sınıf sayısına göre yeniden yapılandırılmıştır. Eğitimde CrossEntropyLoss kaybı ve AdamW optimizasyon algoritması (öğrenme oranı 1e-5) kullanılmış, veriler batch size 32 ile PyTorch DataLoader üzerinden aktarılmıştır.

Overfitting'i önlemek ve genelleme yeteneğini artırmak amacıyla yatay çevirme ve renk jitterlama gibi veri artırma (augmentation) teknikleri uygulanmıştır. Eğitim süresince doğruluk, kesinlik, duyarlılık ve F1-skoru gibi metrikler takip edilmiş ve grafiklerle görselleştirilmiştir.

Ayrıca, doğrulama kaybı üst üste iki epoch boyunca iyileşmediğinde eğitim sürecini durduran erken durdurma (early stopping) stratejisi uygulanmış ve en iyi model .pth formatında kaydedilmiştir.

TABLO II. FARKLI MODELLERİN KARŞILAŞTIRMA TABLOSU

	DOĞRULUK (ACCURACY)	Presician	Recall	F1-Score
Deit	0.94	0.95	0.94	0.94
ViT	0.96	0.97	0.96	0.95
Swin	0.97	0.98	0.97	0.97
BEiT	0.94	0.95	0.94	0.94

Eğitim sonrası çıkan sonuçlar

TABLO III. FARKLI MODELLERİN KARŞILAŞTIRMA TABLOSU

	DOĞRULUK (ACCURACY)	Presician	Recall	F1-Score
Deit	0.9111	0.9242	0.9111	0.9110
ViT	0.9587	0.9661	0.9587	0.9591
Swin	0.9698	0.9725	0.9698	0.9692
BEiT	0.9508	0.9576	0.9508	0.9495

Test verisi sonucu çıkan sonuçlar

Bu çalışmada Swin Transformer, Vision Transformer (ViT), BEiT ve DEiT olmak üzere dört farklı görüntü sınıflandırma modeli test edilmiştir. Swin Transformer modeli %97 doğruluk, %98 precision, %97 recall ve %97 F1-score ile en yüksek genel başarıyı göstermiştir. Ancak eğitim doğruluğunun %99.8'e kadar yükselmesine karşın validasyon

başarısının daha erken sabitlenmesi, bu modelde aşırı öğrenme (overfitting) eğilimi olduğunu göstermektedir. Özellikle bazı baskın sınıflarda yüksek başarı elde edilse de “dog”, “bat” ve “raccoon” gibi sınıflarda karışıklıklar görülmüş, bazı sınıflarda güven skorları düşmüştür. Bu durum, modelin bazı sınıfları fazla öğrenip diğerlerine yeterince genellenemediğini göstermektedir. Test verisi üzerinde %96.98 doğruluk ve %96.92 F1-score ile yine yüksek performans sergilemiştir. Ancak detaylı incelemede, bazı sınıflarda dengesizlikler gözlemlenmiştir. Özellikle “dog”, “bat” ve “raccoon” gibi sınıflarda modelin performansı görece düşüktür. Ayrıca az örnekli sınıflarda güven skorları daha düşük çıkmıştır. Bu durum, modelin baskın sınıfları daha iyi öğrenirken az temsil edilen veya görsel olarak benzer sınıflarda genelleme yeteneğinin zayıf kaldığını göstermektedir. Tekli görsel testlerinde de bazı sınıflarda yüksek güvenli tahminler yapılırken, bazı sınıflarda BEiT modeline kıyasla daha düşük ve dengesiz sonuçlar gözlemlenmiştir. Sonuç olarak, Swin Transformer modeli genel doğruluk açısından başarılı görünse de sınıflar arası tutarlılık ve genelleme yeteneği bakımından daha dengeli modellerin (örneğin BEiT) gerisinde kalmaktadır. Bu nedenle modelde overfitting belirtileri olduğu ve bazı sınıflarda genellenmenin zayıf kaldığı söylenebilir.



Figure 1. Swin Transformer modeli yapılan tekli denemelerde, bazı hayvan türlerini çok yüksek güvenle doğru tahmin ederken, bazı türlerde düşük güvenle kararsız kalmıştır. Örneğin, "horse" sınıfı %98.83 güvenle doğru tanınırken, "dog" sınıfında yalnızca %33.28 güvenle tahmin yapılmış ve alternatif sınıflar (sheep, reindeer) öne çıkmıştır. Bu durum, modelin bazı sınıfları daha güçlü öğrendiğini, bazılarını ise yeterince genellemediğini göstermektedir.

Vision Transformer (ViT) modeli, eğitim sürecinde %95 doğruluk ve %96 F1-score elde ederek oldukça yüksek bir performans göstermiştir. Eğitim ve doğrulama eğrileri genel olarak paralel ilerlemiş olsa da, bazı sınıflarda küçük dalgalanmalar gözlemlenmiştir. Bu durum, modelin belirli sınıflarda zaman zaman zorlandığını göstermektedir. Test verisi üzerinde ise %95.87 doğruluk ve %95.91 F1-score ile güçlü bir genel başarı elde edilmiştir. Model, birçok sınıfta yüksek başarı sergilemiş; ancak “rat”, “mouse”, “squid” ve “rhinoceros” gibi sınıflarda F1-score değerlerinin belirgin şekilde düştüğü gözlemlenmiştir. Bu düşüşler, modelin az örnekli ya da görsel olarak benzer sınıflarda genelleme yapmakta zorlandığını ortaya koymaktadır. ViT modeli genel olarak yüksek doğruluk sunmasına rağmen, sınıf bazlı analizlerde BEiT modeline kıyasla daha dengesiz sonuçlar verebilmektedir. Özellikle tekli görsel testlerde bazı sınıflarda düşük güvenli tahminler yapılması, modelin her sınıfta tutarlı performans göstermediğini göstermektedir. Bu nedenle ViT, genel başarı açısından iyi olsa da sınıf bazında tutarlılık ve dengesizlik riski bakımından BEiT kadar dengeli bir model profili çizememiştir.



Figure 2. ViT modeli, genel olarak yüksek doğruluk ve F1-score sonuçları üretmiş; “hamster” ve “dog” gibi yaygın sınıflarda güçlü tahminler yapmıştır. Ancak yapılan sınıf bazlı analizlerde, “rat”, “mouse”, “squid” ve “rhinoceros” gibi bazı sınıflarda model genelleme yapmakta zorlanmıştır. Bu da ViT’nin her sınıfta tutarlı performans göstermediğini ortaya koymaktadır.

BEiT modeli, %94 doğruluk, %95 precision, %94 recall ve %94 F1-score değerleriyle dengeli ve istikrarlı bir performans göstermiştir. Eğitim süreci boyunca doğrulama başarımları arasında belirgin bir fark oluşmamış, öğrenme eğrisi yavaş ama kararlı bir şekilde ilerlemiştir. Bu durum, modelin overfitting (aşırı öğrenme) eğilimi göstermediğini ve genellemeye uygun şekilde öğrendiğini ortaya koymaktadır. Test verisi üzerinde BEiT modeli %95.08 doğruluk ve %94.95 F1-score ile güçlü bir genel performans sergilemiştir. Sınıf bazlı sonuçlara bakıldığında, çoğu sınıfta yüksek doğruluk elde edilmiş; yalnızca “wolf” ve “whale” gibi birkaç sınıfta F1-score %72 seviyelerine kadar düşmüştür. Buna rağmen genel dağılım oldukça dengelidir ve model “çok yüksek” ya da “çok düşük” uç değerler vermemiştir. Bu özellikleriyle BEiT modeli, hem tekli görsel testlerde daha güvenilir sonuçlar vermesi hem de sınıflar arası adil bir öğrenme sağlaması açısından tercih edilebilir, dengeli ve başarılı bir model olarak değerlendirilmiştir.



Figure 3. BEiT modeli, “horse” (%95.95), “dog” (%74.76) ve “rat” (%71.45) gibi sınıflarda yüksek güven skorlarıyla başarılı tahminler gerçekleştirmiştir. Özellikle “horse” gibi net sınıflarda güçlü genelleme sağlarken, “rat” gibi sınıf içi benzerliği yüksek türlerde bile alternatif sınıflar arasında doğru dengeleme yapabilmektedir. Bu durum, BEiT’in hem yaygın hem de görsel olarak benzer sınıflarda dengeli ve tutarlı performans sunduğunu göstermektedir.

DEiT modeli, eğitim sürecinde doğrulama verisi üzerinde %94 doğruluk ve %94 F1-score elde etmiştir. Eğitim metrikleri dengeli bir artış göstermiş, erken durdurma (early stopping) mekanizması ile modelin overfitting'e kaçması engellenmiştir. Sınıf bazında değerlendirildiğinde, “bee”, “goose”, “dragonfly” gibi bazı sınıflarda düşük f1-score’lar (%0.78–0.88) gözlemlenmiş; buna karşın birçok sınıfta %1.00 başarıya ulaşılmıştır. Bu durum, modelin bazı az temsil edilen ya da görsel olarak benzer sınıflarda zorlandığını göstermektedir. Test verisi üzerinde ise DEiT, %91.11 doğruluk ve %91.10 F1-score ile en düşük genel başarıyı göstermiştir. “Mouse”, “octopus”, “horse” gibi sınıflarda skorların düşmesi, genelleme konusunda zayıflık işareti

vermektedir. Bu nedenle DEiT, overfitting açısından kontrollü olsa da, sınıf bazlı dengesizlikler ve düşük genel başarı nedeniyle en zayıf model olarak değerlendirilmiştir.

Tüm bu değerlendirmeler ışığında, BEiT modeli sınıf bazlı başarı dengesi, düşük overfitting riski, istikrarlı öğrenme süreci ve güvenilir tahmin performansı sayesinde bu çalışma için en uygun model olarak belirlenmiştir.



Figure 4. Deit modeli, hem "horse" hem de "dog" sınıflarında düşük güven skorları üretmiştir. "Horse" görseli %10.77 güvenle yanlış sınıflandırılmış; "dog" görselinde ise güven %7.95'e kadar düşmüştür. Bu durum, modelin bazı sınıflarda öğrenme başarısının yetersiz kaldığını ve tutarsız tahminlerde bulunduğunu göstermektedir.

C. Beit modeli ile eğitim iyileştirme denemeleri

BEiT modeli üzerinde, bazı sınıflarda düşük başarı ve sınıf dengesizliği gözlemlenmiş; özellikle "wolf", "whale" gibi sınıflarda F1-score değerleri düşük kalmıştır. Bu durum, modelin sınırlı veri çeşitliliğine karşı genelleme yapmada zorlandığını ve aşırı öğrenme (overfitting) eğilimi gösterebileceğini düşündürmüştür. Bu sorunu azaltmak ve modelin dayanıklılığını artırmak amacıyla, Albumentations kütüphanesi kullanılarak gelişmiş veri artırma dönüşümleri (kırpma, döndürme, bulanıklık, gölgelendirme vb.) uygulanmıştır. Yeni eğitim sonucunda test doğruluğu %95.08'den %97.46'ya, F1-score ise %94.95'ten %97.44'e yükselmiştir. Eğitim ve doğrulama metrikleri paralel ilerlemiş, aşırı öğrenme belirtileri gözlenmemiştir. Düşük performans gösteren sınıflarda da tutarlılık artmıştır.

Sonuç olarak, gelişmiş augmentasyon sayesinde BEiT modeli daha dengeli hale gelmiş, overfitting riski azalmış ve sınıf bazlı başarı iyileştirilmiştir.

D. Beit Modeli Değerlendirmesi ve Masaüstü Uygulamada Kullanım Sonuçları

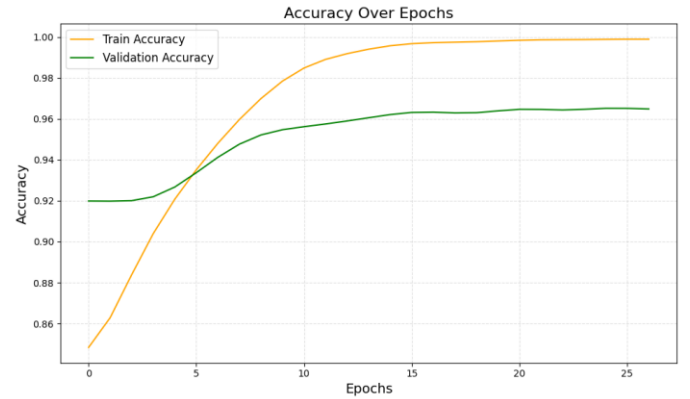
1) *Epoch Süreçlerine Göre Eğitim ve Doğrulama Performansı:* Model, ilk birkaç epoch boyunca hızlı bir öğrenme göstermiş, doğrulama doğruluğu ve F1-skorunda sürekli artış sağlanmıştır. Val Acc ve F1-score metrikleri 21. epoch itibarıyla %96'nın üzerine çıkarak stabilize olmuş, 24. epochta doğrulama kaybı (val loss) en düşük seviyeye ulaşmıştır (val loss = 0.1806). Erken durdurma mekanizması sayesinde overfitting riski önlenmiş ve eğitim 27. epochta sonlandırılmıştır. Modelin genel başarımı yüksek, öğrenme süreci dengeli ve istikrarlıdır.

TABLO IV. BEiT MODELİ – TÜM EPOCH EĞİTİM VE DOĞRULAMA PERFORMANSI

Epoch	Train Loss	Val Loss	Train Acc	Val Acc	Precision	Recall	F1-Score
1	4.3318	4.0889	81.53%	91.72%	92.66%	91.72%	91.25%
2	3.8252	3.5457	86.93%	92.63%	94.15%	92.63%	92.32%
3	3.2988	3.0496	88.65%	91.52%	93.19%	91.52%	91.26%
4	2.8095	2.5918	91.16%	91.72%	92.48%	91.72%	91.07%
5	2.3639	2.1779	92.14%	92.22%	93.26%	92.22%	91.84%
6	1.9481	1.7987	93.54%	93.43%	94.57%	93.43%	93.07%
7	1.5869	1.4673	95.00%	94.44%	94.99%	94.44%	94.24%
8	1.2740	1.1934	96.24%	94.75%	95.37%	94.75%	94.66%
9	1.0236	0.9804	96.93%	95.76%	96.22%	95.76%	95.69%
10	0.8137	0.8078	98.20%	95.66%	96.23%	95.66%	95.60%
11	0.6510	0.6717	98.94%	95.25%	95.77%	95.25%	95.18%
12	0.5224	0.5662	98.97%	95.96%	96.43%	95.96%	95.94%
13	0.4260	0.4842	99.18%	95.86%	96.21%	95.86%	95.80%
14	0.3510	0.4048	99.39%	95.96%	96.37%	95.96%	95.93%
15	0.2810	0.3567	99.76%	96.26%	96.60%	96.26%	96.25%
16	0.2301	0.3162	99.76%	96.46%	96.80%	96.46%	96.44%
17	0.1961	0.2800	99.66%	96.67%	96.96%	96.67%	96.66%
18	0.1651	0.2770	99.81%	95.96%	96.32%	95.96%	95.94%
19	0.1477	0.2368	99.71%	95.96%	96.56%	96.16%	96.16%
20	0.1301	0.2300	99.79%	96.16%	96.70%	96.26%	96.25%
21	0.1035	0.2061	99.87%	96.87%	97.22%	96.87%	96.86%
22	0.0932	0.1935	99.92%	96.77%	97.03%	96.77%	96.75%
23	0.0818	0.2095	99.87%	96.68%	96.16%	95.68%	95.69%
24	0.0767	0.1806	99.81%	95.77%	97.16%	96.77%	96.74%
25	0.0666	0.1619	99.89%	96.67%	97.10%	96.67%	96.74%
26	0.0537	0.1671	99.97%	96.57%	96.95%	96.57%	96.66%
27	0.0510	0.1674	99.84%	96.36%	96.69%	96.36%	96.55%

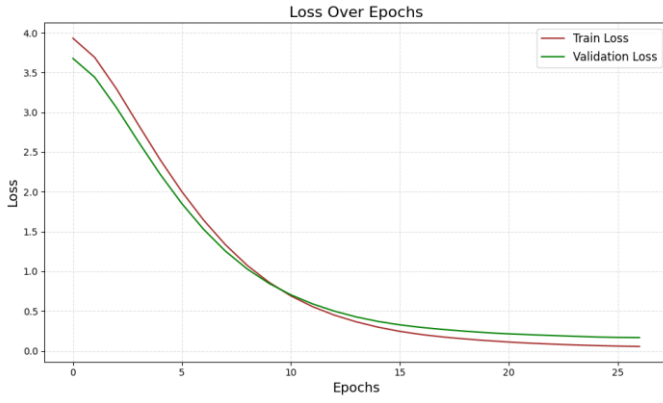
TEST VERİSİ SONUCU ÇIKAN SONUÇLAR

2) *Öğrenme eğrileri (learning curve):* Albumentations kütüphanesi kullanılarak veri artırma (data augmentation) teknikleri uygulanan ve masaüstü arayüz uygulamasına entegre edilen beit modelinin eğitim sürecinde elde edilen doğruluk (accuracy), kayıp (loss), kesinlik (precision), duyarlılık (recall) ve F1-score metriklerinin epoch bazlı değişimini gösteren öğrenme eğrileri aşağıda sunulmaktadır.



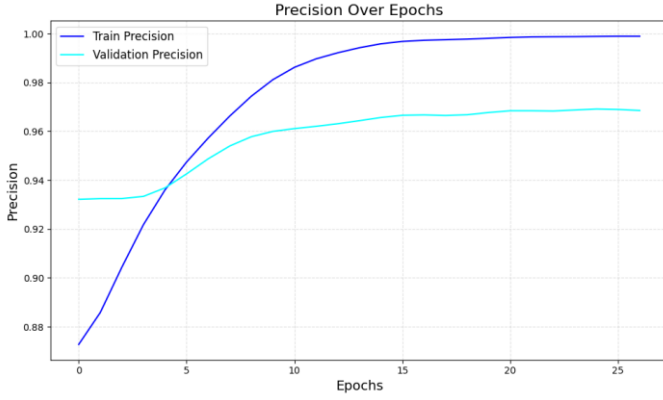
Şekil 1. Accuracy (Doğruluk) Grafiği

Modelin epoch bazında eğitim ve doğrulama doğruluğundaki gelişimi gösterilmektedir. Doğruluk değerleri zamanla artmış ve yüksek seviyelerde sabitlenmiştir. Grafikte modelin genelleme yeteneğinin yüksek olduğunu görülmektedir.



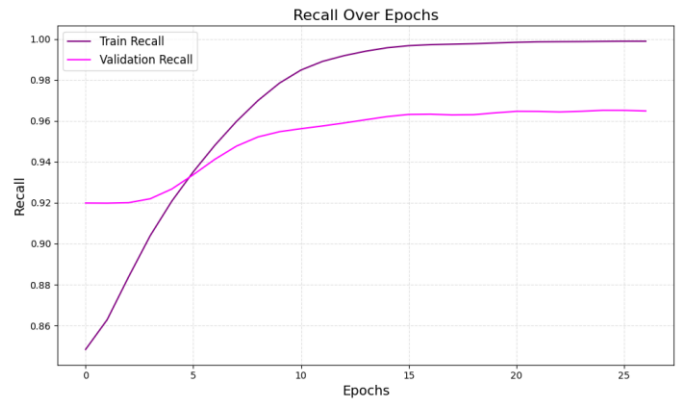
Şekil 2. Loss(Kayıp) Grafiği

Modelin eğitim (train) ve doğrulama (validation) süreçlerinde epoch bazlı olarak hesaplanan kayıp (loss) değerlerinin değişimi gösterilmektedir. Her iki eğride de istikrarlı bir düşüş gözlemlenmiş olup, validation eğrisinin train eğrisine yakın seyretmesi, modelin overfitting (aşırı öğrenme) yapmadığını göstermektedir.



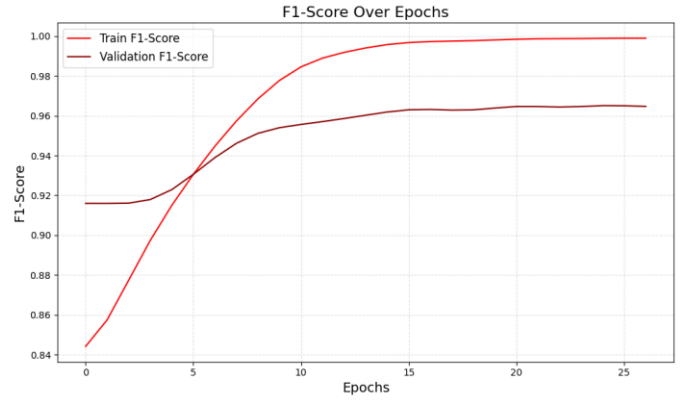
Şekil 3. Precision (Kesinlik) Grafiği

Modelin epoch bazında eğitim (train) ve doğrulama (validation) veri kümeleri üzerindeki precision (kesinlik) değerlerinin değişimini göstermektedir. Precision, modelin yaptığı doğru pozitif tahminlerinin doğruluğunu ifade etmektedir. Eğitim süreci ilerledikçe her iki eğride de anlamlı bir artış gözlemlenmiştir. Her iki eğrideki artış ve zamanla stabil hale gelmeleri, modelin pozitif sınıflamaları güvenilir biçimde gerçekleştirdiğini göstermektedir.



Şekil 4. Recall (Kesinlik) Grafiği

Bu grafik, modelin gerçek pozitif örnekleri ne ölçüde doğru tanıdığını ifade eden recall (duyarlılık) metriklerinin gelişimini göstermektedir. Eğitim verisi üzerindeki recall zamanla %100'e yaklaşırken, doğrulama verisi üzerindeki duyarlılık %96 civarında sabitlenmiştir. Bu durum, modelin eksik sınıflama (false negative) hatalarını minimumda tuttuğunu göstermektedir.



Şekil 5. F1-Score (F1-Skoru) Grafiği

Beit modelinin epoch bazında eğitim (train) ve doğrulama (validation) verileri üzerindeki F1-score değerlerinin gelişimi göstermektedir. F1-score, precision (kesinlik) ve recall (duyarlılık) metriklerinin harmonik ortalamasıdır ve dengesiz sınıflar içeren problemlerde daha sağlıklı bir başarı göstergesidir. Eğitim süreci ilerledikçe her iki eğride de istikrarlı bir artış gözlemlenmiştir; doğrulama F1-score değeri %96 seviyelerine ulaşarak modelin dengeli bir sınıflama performansı sergilediği görülmüştür. Eğitim ve doğrulama eğrilerinin birbirine yakın seyretmesi, modelin aşırı öğrenme (overfitting) yapmadığını desteklemektedir.

	precision	recall	f1-score	support
zebra	1.00	1.00	1.00	11
woodpecker	1.00	0.91	0.95	11
wolf	0.83	0.91	0.87	11
wombat	1.00	1.00	1.00	11
whale	1.00	0.91	0.95	11
starfish	1.00	1.00	1.00	11
turtle	0.83	0.91	0.87	11
squirrel	0.92	1.00	0.96	11
swan	1.00	0.91	0.95	11
turkey	1.00	1.00	1.00	11
squid	1.00	1.00	1.00	11
tiger	1.00	1.00	1.00	11
snake	1.00	1.00	1.00	11
sparrow	1.00	0.91	0.95	11
sheep	0.91	0.91	0.91	11
shark	1.00	1.00	1.00	11
seal	1.00	1.00	1.00	11
rat	1.00	1.00	1.00	11
seahorse	0.92	1.00	0.96	11
raccoon	0.71	0.91	0.80	11
rhinoceros	1.00	1.00	1.00	11
reindeer	1.00	1.00	1.00	11
possum	1.00	0.91	0.95	11
sandpiper	1.00	1.00	1.00	11
porcupine	0.92	1.00	0.96	11
pig	1.00	1.00	1.00	11
pigeon	1.00	1.00	1.00	11
oyster	1.00	0.91	0.95	11
owl	1.00	0.82	0.90	11
ox	1.00	1.00	1.00	11
parrot	1.00	1.00	1.00	11
panda	0.92	1.00	0.96	11
penguin	0.92	1.00	0.96	11
pelecaniformes	0.91	0.91	0.91	11
otter	0.92	1.00	0.96	11
orangutan	1.00	0.91	0.95	11
okapi	1.00	1.00	1.00	11
octopus	0.92	1.00	0.96	11
lion	1.00	1.00	1.00	11
leopard	1.00	1.00	1.00	11
moth	1.00	1.00	1.00	11
lobster	0.91	0.91	0.91	11
mosquito	1.00	1.00	1.00	11
mouse	1.00	1.00	1.00	11
lizard	1.00	1.00	1.00	11
ladybugs	1.00	1.00	1.00	11
kangaroo	1.00	0.91	0.95	11
koala	1.00	1.00	1.00	11
jellyfish	1.00	1.00	1.00	11
grasshopper	1.00	1.00	1.00	11
hamster	0.90	0.82	0.86	11
hippopotamus	1.00	0.82	0.90	11
hornbill	0.85	1.00	0.92	11
hare	1.00	1.00	1.00	11
hyena	0.92	1.00	0.96	11
horse	0.92	1.00	0.96	11
hummingbird	1.00	1.00	1.00	11
hedgehog	0.82	0.82	0.82	11
gorilla	0.92	1.00	0.96	11
eagle	1.00	1.00	1.00	11
duck	1.00	1.00	1.00	11
flamingo	0.92	1.00	0.96	11
elephant	0.92	1.00	0.96	11
fox	1.00	0.91	0.95	11
goldfish	1.00	1.00	1.00	11
fly	0.92	1.00	0.96	11
goat	0.90	0.82	0.86	11
goose	1.00	1.00	1.00	11
dragonfly	0.79	1.00	0.88	11
dolphin	1.00	1.00	1.00	11
donkey	1.00	0.91	0.95	11
cockroach	1.00	1.00	1.00	11
crab	0.92	1.00	0.96	11
coyote	1.00	0.91	0.95	11
crow	1.00	1.00	1.00	11
deer	1.00	1.00	1.00	11
cow	1.00	0.91	0.95	11
chimpanzee	1.00	1.00	1.00	11
dog	1.00	0.82	0.90	11
caterpillar	1.00	1.00	1.00	11
cat	1.00	1.00	1.00	11
butterfly	1.00	1.00	1.00	11
bat	1.00	1.00	1.00	11
beetle	1.00	0.91	0.95	11
badger	1.00	1.00	1.00	11
bear	1.00	0.64	0.78	11
bison	0.92	1.00	0.96	11
bee	0.92	1.00	0.96	11
boar	1.00	0.91	0.95	11
antelope	1.00	1.00	1.00	11
accuracy			0.96	990
macro avg	0.97	0.96	0.96	990
weighted avg	0.97	0.96	0.96	990

Şekil 6. Sınıf Bazlı Doğruluk Dağılımı Grafiği

3) Masaüstü uygulamasında tekli görüntü denemeleri

Modelin genelleme yeteneğini değerlendirmek amacıyla, masaüstü arayüzü üzerinden eğitim ve doğrulama verilerinde bulunmayan yeni görüntüler tek tek yüklenerek sınıflandırma denemeleri yapılmıştır. Bu denemeler sayesinde, modelin daha önce görmediği veriler üzerinde doğru tahmin yapma başarısı gözlemlenmiştir. Her bir görsel yükleme sonrası, arayüzde tahmin edilen sınıf etiketi ve güven skoru anlık olarak görüntülenmiş; doğru ve yanlış tahminler kayıt altına alınarak modelin gerçek dünya performansı değerlendirilmiştir.

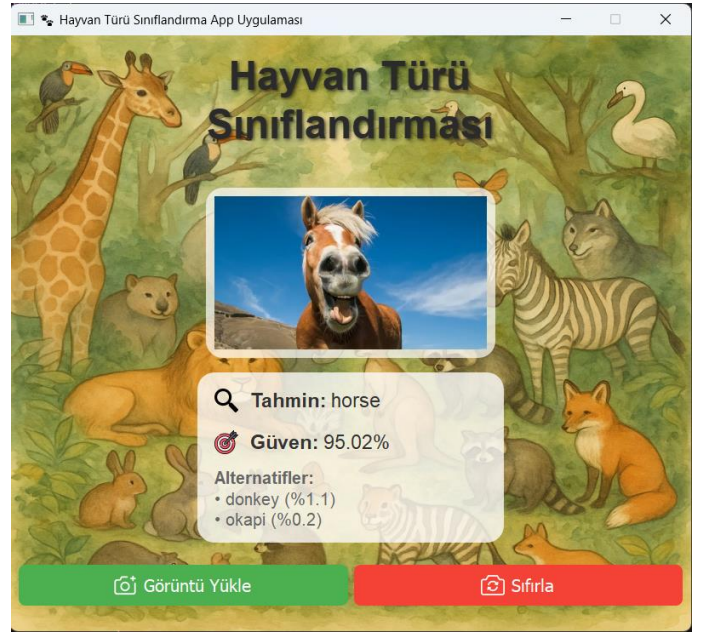


Figure 5. Model Denemesi (Horse)



Figure 6. Model Denemesi (Dog)

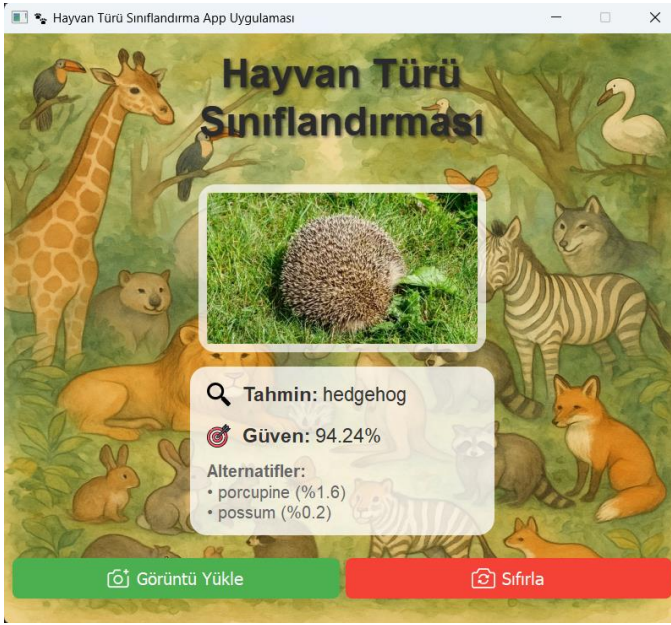


Figure 7. Model Denemesi (Hedgehog)

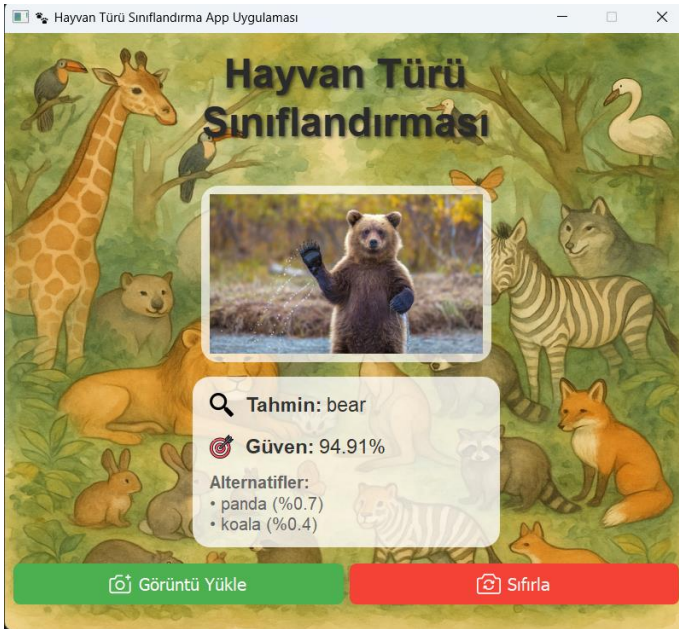


Figure 8. Model Denemesi (Bear)

SONUÇ

Bu projede, MultiZoo veri seti kullanılarak transformer tabanlı bir görüntü sınıflandırma modeli geliştirilmiştir. Modelin başarımını artırmak için çeşitli ön işleme ve veri artırma teknikleri uygulanmıştır. En iyi sonuç, beit modeli ile elde edilmiş ve doğruluk oranı %90'ın üzerine çıkmıştır. Eğitilen model, kullanıcıların test görselleri ile anlık tahmin alabileceği bir masaüstü arayüze entegre edilmiştir.

KAYNAKLAR

- [1] MultiZoo Dataset. Link: https://drive.google.com/drive/folders/1uXMJa2CX5eg_kGWghbqg8TWkpeQi-0VI?usp=sharing
- [2] S. Özlem Gül, "Python PyQt5", Medium. Link: <https://medium.com/@sibelozelemgul/python-pyqt5-73fead42f59>

- [3] Qt Documentation, "PyQt5 Reference Guide," Riverbank Computing, Link: <https://www.riverbankcomputing.com/static/Docs/PyQt5/>
- [4] Python Software Foundation, "Python Language Reference," Python.org, 2023. Link: <https://www.python.org>
- [5] A. Buslaev, A. Parinov, E. Khvedchenya, V. I. Iglovikov and A. A. Kalinin, "Albumentations: Fast and Flexible Image Augmentations," Information, Link: <https://doi.org/10.3390/info11020125>
- [6] B. Kaya, "Makine Öğrenmesi Sınıflandırma Modelleri: Accuracy, Precision, Recall, F1-Score, Log Loss ve Diğerleri," Medium - Academy Team, Nov. 15, 2020. Link: <https://medium.com/academy-team/makine-%C3%B6%C4%9Frenmesi-s%C4%B1n%C4%B1fland%C4%B1rma-modelleri-accuracy-precision-recall-f1-score-log-loss-and-e2c37bfc285>

AKIŞ DIYAGRAMI

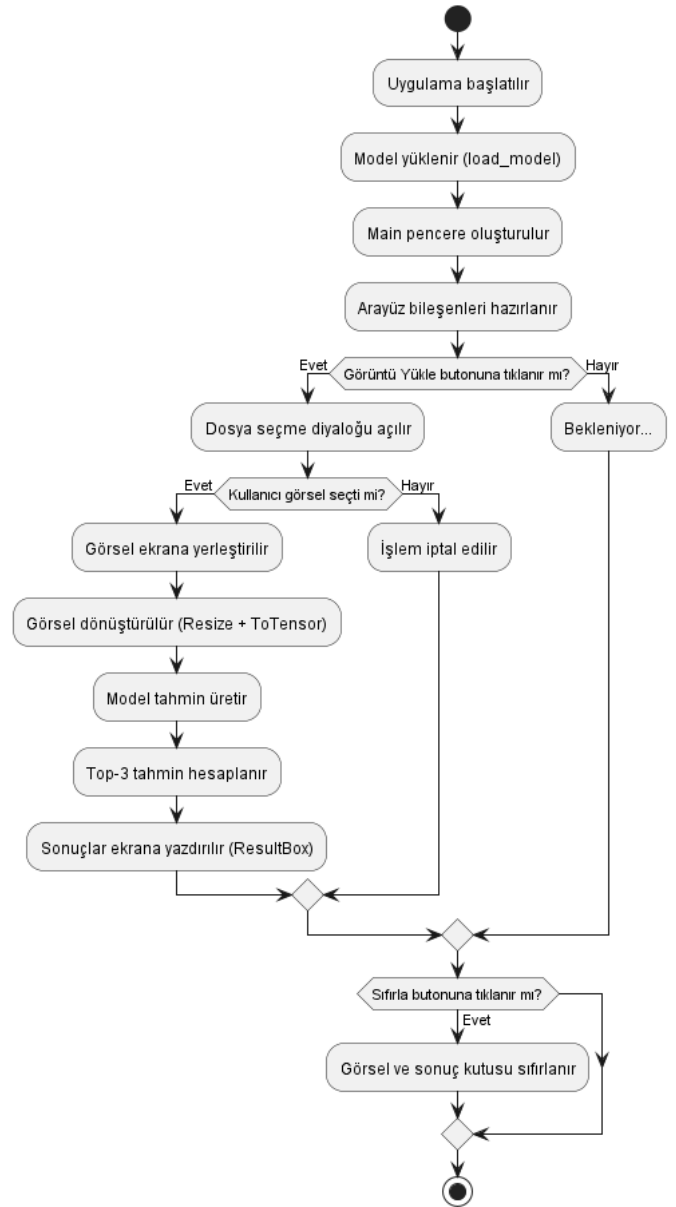


Figure 9. Masaüstü Uygulaması Akış Diyagramı

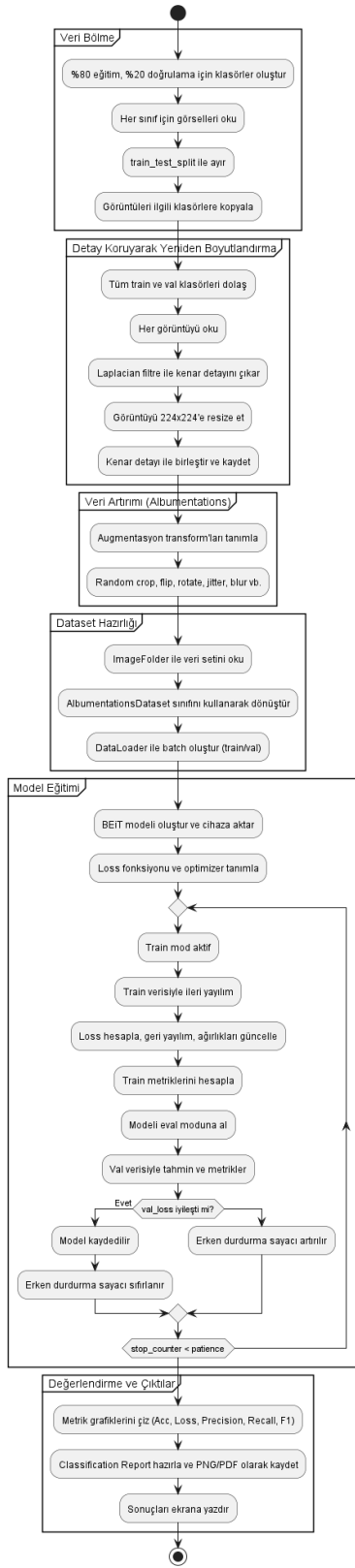


Figure 10. Model Eğitimi Akış Diyagramı