T.C.

SAKARYA ÜNİVERSİTESİ

BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ

BSM 342 DERİN ÖĞRENME VE EVRİŞİMLİ SİNİR AĞLARI

GRUP ÜYELERİ

B211210085 - Vedat ÖZTÜRK

B211210040 - Berkay SARAY

B211210574 - Mahdi Hojjati

G221210382 - Yetgin Akcan

Deniz Hayvanlarının Sınıflandırılması

2024-2025 Bahar Dönemi

1. Giriş

Model, önceden eğitilmiş EfficientNetB3 modeli üzerine inşa edilmiş ve transfer öğrenme yöntemiyle özelleştirilmiştir. Amaç, her bir görseli ait olduğu canlı türüne en yüksek doğrulukla atayabilen bir sınıflandırma modeli geliştirmektir.

2. Veri Hazırlığı

Veriler './archive' klasörü altında yer almakta olup, her bir alt klasör bir deniz canlısı sınıfını temsil etmektedir. Kodlama aşamasında, Python *os* ve *shutil* kütüphaneleri kullanılarak veriler organize edilmiştir. Görseller rastgele karıştırılarak eğitim (%75), doğrulama (%10) ve test (%15) setlerine ayrılmıştır.

3. Veri Yükleme ve Ön İşleme

TensorFlow'un *image_dataset_from_directory* fonksiyonu ile görseller yüklenmiştir. Görseller 300x300 çözünürlüğe ölçeklenmiş, sınıflar ise, indisler ile belirlenmiştir.

4. Model Mimarisi ve Kurulumu

EfficientNetB3 modeli, ImageNet üzerinde önceden eğitilmiş biçimde kullanılmış, son 3 katmanı, modelin fine-tune edilmesi için eğitime kapatılmamıştır. Sınıflandırma için yeni eklenen katmanlar: GlobalAveragePooling2D, Dense(32, ReLU), Dropout(0.1) ve Dense(23, softmax). Toplamda 23 sınıf için sınıflandırma yapılmaktadır. Optimizer olarak *Adam*, loss hesabı için *sparse_categorical_crossentropy* ve doğruluk için *accuracy* kullanılmıştır.

5. Model Eğitimi

Model 10 epoch boyunca eğitim setinde eğitilmiş, doğrulama seti üzerinden izlenmiştir. EarlyStopping ve ModelCheckpoint gibi callback fonksiyonları kullanılmıştır.

6. Sonuçların Görselleştirilmesi ve Değerlendirme

Eğitim ve doğrulama süreçlerine ait doğruluk ve kayıp değerleri matplotlib kullanılarak görselleştirilmiştir. Test seti ile modelin genel başarımı ölçülmüş, sınıflandırma sonuçlarının güvenilir olduğu gözlemlenmiştir.

7. Grad-CAM ile Açıklanabilir Yapay Zeka

Modelin karar mekanizmasını anlamak için Grad-CAM yöntemi kullanılmıştır. Bu teknik, modelin belirli bir sınıfa karar verirken görselin hangi bölümlerine odaklandığını gösteren bir ısı haritası üretir.

8. Sonuç

EfficientNet mimarisi ve transfer öğrenme kullanılarak geliştirilen bu derin öğrenme modeli, deniz canlılarının görüntülerini %88'e yakın bir doğrulukla sınıflandırabilmektedir.

Transfer Learning ile Görüntü Sınıflandırma: VGG16 Tabanlı Model Raporu

1. Giriş

Bu çalışmada, önceden eğitilmiş derin öğrenme modellerinden biri olan **VGG16** mimarisi kullanılarak görüntü sınıflandırma görevi gerçekleştirilmiştir. Model, **ImageNet** veri kümesi üzerinde eğitilmiş VGG16'nın öğrenilmiş görsel özniteliklerinden yararlanılarak transfer learning (aktarılmış öğrenme) yaklaşımıyla geliştirilmiştir. Sıfırdan bir model eğitmek yerine bu yöntem kullanılarak hem eğitim süresi kısaltılmış hem de genel başarı oranı artırılmıştır.

2. Model Mimarisi

2.1. Taban Model

Modelin omurgasını oluşturan **VGG16** mimarisi, include_top=False parametresi ile son sınıflayıcı katmanlar çıkarılmış şekilde kullanılmıştır. Bu sayede yalnızca convolutional katmanlar alınmış ve bu katmanlar dondurularak (trainable=False) yeniden eğitilmeleri engellenmiştir. Böylece model, daha önce öğrendiği temel öznitelikleri korumuştur.

- **Model:** VGG16(weights='imagenet', include_top=False)
- **Dondurulan Katmanlar:** Tüm convolution katmanları

2.2. Girdi ve Ön İşleme

- **Girdi Boyutu:** 300x300x3 (renkli görüntüler)
- **Ön İşleme:** Piksel değerleri 0 ile 1 arasına normalize edilmiştir: Rescaling(1./255)

2.3. Eklenmiş Sınıflayıcı Katmanlar

VGG16'nın üzerine aşağıdaki katmanlar eklenmiştir:

1. **GlobalAveragePooling2D:** Özellik haritalarını sıkıştırmak ve sabit uzunlukta bir vektör üretmek için kullanılmıştır.
2. **Dense(128, activation='relu')**: Öğrenilebilir 128 nöronluk tam bağlantılı katman.

3. **Dropout(0.3):** Aşırı öğrenmeyi (overfitting) engellemek amacıyla %30 oranında nöronlar rastgele devre dışı bırakılmıştır.
4. **Dense(23, activation='softmax')**: 23 sınıflı çoklu sınıflandırma için çıkış katmanı.

3. Eğitim Süreci

3.1. Parametreler

- **Optimizer:** Adam(learning_rate=0.001, beta_1=0.9)
- **Loss Function:** sparse_categorical_crossentropy
  (Etiketler integer formatında verildiği için bu kayıp fonksiyonu seçilmiştir.)
- **Metric:** accuracy

3.2. Eğitim Yapılandırması

- **Epoch Sayısı:** 25
- **EarlyStopping:**
  - monitor='loss'
  - patience=5
    (Model, kayıpta gelişme gözlenmediğinde erken durdurma uygulanmıştır.)
- **Model Checkpoint:**
  - Her epoch sonrası model .keras formatında kaydedilmiştir.

4. Sonuç ve Değerlendirme

Transfer learning yöntemi ile VGG16 tabanlı bir model oluşturularak sınıflandırma görevi başarıyla gerçekleştirilmiştir. Eğitim süresi klasik modellere göre kısalmış, yüksek doğruluk oranı elde edilmiştir. Özellikle sınırlı veri setlerinde bu yöntem, düşük kaynak tüketimiyle etkili sonuçlar vermektedir.

Custom CNN Modeli Raporu

1. Giriş

Bu projede, görüntü sınıflandırma problemi için özel olarak tasarlanmış bir **Convolutional Neural Network (CNN)** modeli geliştirilmiştir. Model, 300x300 boyutundaki RGB (3 kanallı) görüntüleri

giriş olarak almakta ve 23 farklı sınıfa ayırmaktadır. Eğitim süreci boyunca hem modelin mimarisi hem de optimizasyon parametreleri dikkatle yapılandırılmıştır.

2. Model Mimarisi

Model, derin öğrenmede yaygın olarak kullanılan katmanlardan oluşturulmuş ve aşağıdaki şekilde yapılandırılmıştır:

2.1 Convolutional Bloklar

- Modelde toplam **4 adet Conv2D bloğu** bulunmaktadır.
- Her blok şunları içerir:
    - **Conv2D** katmanı: Özellik çıkarımı için temel yapıtaşıdır.
    - **BatchNormalization**: Öğrenme sürecini kararlı hale getirir, overfitting'i azaltır.
    - **MaxPooling2D**: Özellik haritalarını küçülterek hem boyut hem de hesaplama maliyeti düşürülmüştür.

2.2 Aktivasyon Fonksiyonu

- **Swish** aktivasyon fonksiyonu tercih edilmiştir.
- Swish, f(x)=x·sigmoid(x)f(x) = x \cdot \text{sigmoid}(x)f(x)=x·sigmoid(x) formülüyle çalışır ve ReLU'ya kıyasla daha akıcı bir gradyan geçişi sunarak daha yüksek doğruluk sağlamıştır.

2.3 Yoğun (Dense) Katmanlar

- **128 → 64** nöron yapısına sahip iki Dense katman kullanılmıştır.
- Bu katmanlar, konvolüsyonel katmanlar sonrası öğrenilen özellikleri sınıflandırma için anlamlı hale getirir.

2.4 Çıkış Katmanı

- Çıkış katmanında **Softmax** aktivasyonu kullanılarak 23 sınıf için olasılık tahmini yapılmaktadır.

3. Eğitim Süreci

3.1 Optimizasyon

- Eğitimde **Adam optimizörü** kullanılmıştır.
- **beta_1** parametresi, önceki gradyanlara ağırlık vererek momentum etkisi sağlar ve modelin kararlı bir şekilde öğrenmesini kolaylaştırır.

3.2 Kayıp Fonksiyonu

- Çok sınıflı sınıflandırma problemi olduğu için **sparse_categorical_crossentropy** kayıp fonksiyonu tercih edilmiştir.

3.3 Epoch ve Kayıt

- Model, **30 epoch** boyunca eğitilmiştir.
- Her epoch sonunda, modelin o ana kadarki durumu **.keras formatında checkpoint** dosyasına kaydedilmiştir. Bu sayede en iyi model sonradan yüklenebilir durumdadır.

4. Model Performansı

4.1 Değerlendirme

- Eğitim sonrası **en iyi model (epoch 30)** yüklenmiş ve test verisi üzerinde değerlendirilmiştir.

4.2 Test Sonuçları

- **Doğruluk (Accuracy)**: %71.12
- **Kayıp (Loss)**: 0.9973

Bu sonuçlar, modelin genel olarak iyi bir sınıflandırma performansı sunduğunu, ancak gelişim için hâlâ alan olduğunu göstermektedir.

Transfer Learning ile ResNet50 Modeli Raporu

1. Giriş

Bu çalışmada, deniz hayvanlarının sınıflandırılması için önceden eğitilmiş ResNet50 mimarisi kullanılmıştır. ResNet50, derin ağlarda eğitim sırasında oluşabilecek performans düşüşlerini engellemek için "residual connection" (artık bağlantı) mekanizmasını kullanan etkili bir derin öğrenme modelidir. Transfer öğrenme yöntemiyle, ImageNet veri kümesinde öğrenilmiş özelliklerden yararlanılarak modelin daha hızlı ve etkili bir şekilde eğitilmesi sağlanmıştır.

2. Model Mimarisi

2.1. Taban Model

- **Model**: ResNet50(weights='imagenet', include_top=False, input_shape=(227, 227, 3))
- **Dondurulan Katmanlar**: Taban modelin tüm katmanları başlangıçta dondurulmuştur (trainable=False). Bu sayede önceden öğrenilmiş özellikler korunmuştur.

2.2. Özelleştirilmiş Katmanlar

Taban modelin üzerine aşağıdaki katmanlar eklenmiştir:

1. **GlobalAveragePooling2D**: Özellik haritalarını sabit boyutlu bir vektöre dönüştürmek için kullanılmıştır.
2. **Dense(512, activation='relu')**: Öğrenilebilir 512 nöronlu tam bağlantılı katman.
3. **BatchNormalization**: Aktivasyon dağılımlarını normalize ederek eğitimi stabilize etmiştir.
4. **Dropout(0.5)**: Aşırı öğrenmeyi engellemek için %50 oranında nöronlar rastgele devre dışı bırakılmıştır.
5. **Dense(256, activation='relu')**: İkinci öğrenilebilir tam bağlantılı katman.
6. **BatchNormalization**: İkinci normalizasyon katmanı.
7. **Dropout(0.3)**: İkinci dropout katmanı.
8. **Dense(23, activation='softmax')**: 23 sınıflı çoklu sınıflandırma için çıkış katmanı.

3. Eğitim Süreci

3.1. Optimizasyon

- **Optimizer**: Adam(learning_rate=1e-4)
- **Loss Function**: sparse_categorical_crossentropy
- **Metric**: accuracy

3.2. Callback'ler

1. **EarlyStopping**:
    - monitor='val_loss'
    - patience=8
    - restore_best_weights=True
2. **ModelCheckpoint**:
    - filepath='best_resnet50_model.h5'
    - monitor='val_loss'
    - save_best_only=True

3. **ReduceLROnPlateau**:
    - monitor='val_loss'
    - factor=0.3
    - patience=4
    - min_lr=1e-6

4. Sonuçlar ve Değerlendirme

4.1. Performans Metrikleri

- **Test Accuracy**: %78.87
- **Test Loss**: 0.9973

4.2. Confusion Matrix

Confusion matrix analizi, modelin özellikle benzer görünümlü deniz canlıları arasında (örneğin farklı balık türleri) karıştırma yapabildiğini göstermiştir. Ancak genel olarak sınıflar arasında iyi bir ayrım yapabilmiştir.

5.Avantajlar ve Dezavantajlar

5.1.Avantajlar

- Residual connection (artık bağlantı) mekanizması sayesinde, derin ağlarda bile eğitim sırasında performans düşüşü yaşanmamıştır. Bu, modelin karmaşık özellikleri öğrenmesini kolaylaştırmıştır.
- ImageNet veri kümesinde önceden eğitilmiş olması, modelin temel görsel özellikleri hızlı bir şekilde öğrenmesini sağlamıştır. Bu sayede, sınırlı veriyle bile yüksek performans elde edilebilmiştir.

5.2.Dezavantajlar

- Modelin derinliği ve parametre sayısı, küçük veri setlerinde overfitting (aşırı öğrenme) riskini artırabilir.
- ResNet50, derin bir mimariye sahip olduğu için eğitim ve çıkarım sırasında yüksek hesaplama gücü gerektirir. Bu, düşük kaynaklı sistemlerde bir dezavantaj oluşturabilir