T.C Kütahya Dumlupınar Üniversitesi

Bilgisayar Mühendisliği Bölümü

Yüksek Düzey Programlama

Cleaned vs Dirty Classifier

Ad-Soyad : Nurullah Özatak

Öğrenci No: 202013171038

Giriş

Bu projenin amacı, Kaggle'dan elde edilen "Cleaned vs Dirty" veri setini kullanarak temiz ve kirli görüntüleri sınıflandırmaktır. Bu problem, görüntü işleme ve derin öğrenme alanlarında yaygın olarak karşılaşılan bir sınıflandırma problemidir. Proje kapsamında veri ön işleme, model eğitimi, değerlendirme ve sonuç analizi yapılmıştır.

Veri Seti ve Önişleme

Veri Seti

Bu projede kullanılan veri seti, Kaggle'daki "Cleaned vs Dirty" veri setidir. Veri seti, temiz (cleaned) ve kirli (dirty) olmak üzere iki sınıftan oluşmaktadır. Her bir sınıfta bulunan görseller, günlük yaşamda farklı yüzeylerde karşılaşılabilecek temiz ve kirli durumları temsil etmektedir. Veri seti aşağıdaki klasör yapısına sahiptir:

- train/ klasörü: Modelin eğitimi için kullanılan görüntüler.
- test/ klasörü: Modelin doğrulama ve test işlemleri için kullanılan görüntüler.
- sample_submission.csv: Test verileri için etiketlerin yer aldığı CSV dosyası.

Veri seti boyutları:

- Eğitim verisi (train): 40 görüntü
- Test verisi (test): 744 görüntü

Veri setinde sınıflar arasında ciddi bir dengesizlik bulunmaktadır. Örneğin, dirty sınıfındaki görseller sayıca cleaned sınıfına göre oldukça fazladır. Bu nedenle sınıf dengesizliğini gidermek için çeşitli veri artırma (data augmentation) teknikleri uygulanmıştır.

Veri Ön İşleme Adımları

Veri seti üzerinde yapılan ön işleme adımları aşağıdaki gibi sıralanabilir:

1. Görsellerin Yeniden Boyutlandırılması ve Normalizasyon:
    - Eğitim ve test görselleri, modelin girdisiyle uyumlu olacak şekilde 224x224 piksel boyutlarına küçültülmüştür.
    - Görsellerin piksel değerleri [0, 1] aralığına normalize edilmiştir. Bu işlem, modelin daha hızlı ve daha doğru öğrenmesini sağlamaktadır.
2. Etiketlerin Hazırlanması:
    - Test görsellerine karşılık gelen etiketler sample_submission.csv dosyasından alınmıştır.
    - CSV dosyasındaki id sütunu, test görsellerinin isimlerini temsil etmektedir. Bu görsellerin isimleri ile klasördeki dosyalar arasında eşleşme sağlanmıştır.
    - Etiketler (cleaned, dirty) sayısal değerlere dönüştürülmüştür:
        - cleaned -> 0
        - dirty -> 1
3. Veri Artırımı (Data Augmentation):
    - cleaned sınıfındaki görsellerin sayısı az olduğu için bu sınıfa çeşitli veri artırımı teknikleri uygulanmıştır. Uygulanan teknikler şunlardır:
        - Döndürme (Rotation): Görseller rastgele 10°-20° aralığında döndürülmüştür.
        - Yakınlaştırma (Zoom): Görseller rastgele ölçeklendirilmiştir.
        - Aynalama (Horizontal Flip): Görseller yatay olarak çevrilmiştir.
    - Bu işlem, cleaned sınıfında modelin çeşitliliği öğrenmesini sağlamıştır.
4. Sınıf Dengesizliğinin Giderilmesi:
    - Eğitim sırasında sınıflar arasındaki dengesizlik, class_weights parametresi kullanılarak giderilmiştir. Bu, modelin azınlık sınıf (cleaned) üzerinde daha iyi performans göstermesini sağlamıştır.
5. Görsellerin Ayrıştırılması ve Gruplandırılması:
    - Eğitim ve test verileri, NumPy dizilerine dönüştürülmüştür. Böylece görseller

modelin girişine uygun bir formatta hazırlanmıştır:

- Eğitim görüntüleri: (40, 224, 224, 3) boyutunda bir dizi.
- Test görüntüleri: (744, 224, 224, 3) boyutunda bir dizi.

- Etiketler ise (40,) ve (744,) boyutunda tek boyutlu dizilere dönüştürülmüştür.

Veri ön işleme adımları:

1. Görsellerin Yeniden Boyutlandırılması ve Normalizasyon

Tüm görseller, modelin giriş formatına uygun olarak **224x224 piksel** boyutlarına yeniden boyutlandırılmıştır.

Görsellerin piksel değerleri [0, 255] aralığından [0, 1] aralığına normalize edilmiştir. Bu işlem, modelin daha hızlı ve stabil öğrenmesini sağlar.

2. Etiketlerin Hazırlanması

Test görsellerinin etiketleri, sample_submission.csv dosyasından alınmıştır. Dosyada her görselin ID'si (id) ve etiketi (label) bulunmaktadır.

cleaned etiketi 0, dirty etiketi ise 1 olarak sayısal değerlere dönüştürülmüştür.

Görseller ve etiketler, NumPy dizileri şeklinde saklanarak model için giriş verisi formatına uygun hale getirilmiştir.

3. Veri Artırımı (Data Augmentation)

Veri setinde sınıflar arasında ciddi bir dengesizlik olduğu için **cleaned** sınıfına yönelik veri artırımı yapılmıştır:

Döndürme (Rotation): Görseller rastgele açılarda döndürülerek çeşitlilik artırılmıştır.

Yakınlaştırma (Zoom): Görseller rastgele ölçeklendirilmiştir.

Aynalama (Flip): Görseller yatay olarak çevrilerek veri setine yeni görseller eklenmiştir.

Bu adımlar, azınlık sınıfındaki verinin model tarafından daha iyi öğrenilmesini sağlamıştır.

4. Sınıf Dengesizliğinin Giderilmesi

Sınıf ağırlıkları kullanılarak model, azınlık sınıf olan cleaned görsellerine daha fazla önem verecek şekilde eğitilmiştir.

Bu işlem, modelin iki sınıf arasında dengeli performans göstermesini sağlamıştır.

5. Eksik veya Hatalı Verilerin Kontrolü

Test klasöründeki görseller ile CSV dosyasındaki ID'ler arasında eşleşme kontrol edilmiştir.

Eksik veya hatalı dosyalar tespit edilerek gerekli düzenlemeler yapılmıştır.

6. Veri Ayrıştırma ve Gruplandırma

Eğitim verileri (40, 224, 224, 3) boyutunda bir diziye dönüştürülmüş ve etiketleri (40,) boyutunda tutulmuştur.

Test verileri ise (744, 224, 224, 3) boyutunda bir diziye ve etiketleri (744,) boyutunda bir diziye dönüştürülmüştür.

Model ve Yöntemler

1. Kullanılan Model

Proje kapsamında temel bir Convolutional Neural Network (CNN) modeli oluşturulmuştur. Bu model, görüntülerin özelliklerini çıkarıp sınıflandırma yapabilmek için yaygın olarak kullanılan bir derin öğrenme mimarisidir.

Modelin Mimari Yapısı:

- Giriş Katmanı:
  - 224x224 boyutunda ve 3 kanallı (RGB) görselleri giriş olarak alır.
- Convolutional Katmanlar:
  - Görsellerden özellik çıkarımı yapılmasını sağlar.
  - Her bir Convolutional katmandan sonra ReLU (Rectified Linear Unit) aktivasyon fonksiyonu uygulanmıştır.
- MaxPooling Katmanları:
  - Özellik haritalarındaki önemli bilgileri koruyarak boyut küçültmesi yapılmıştır.
- Flatten Katmanı:
  - Çok boyutlu özellik haritaları, tam bağlantılı katmanlar (fully connected layers) için tek boyutlu hale getirilmiştir.
- Tam Bağlantılı (Fully Connected) Katmanlar:
  - Çıkış sınıflarını tahmin etmek için kullanılmıştır.
  - Son katmanda, Softmax aktivasyon fonksiyonu ile iki sınıfa (cleaned, dirty) ait olasılıklar tahmin edilmiştir.

---

2. Eğitim Ayarları

Modelin eğitimi sırasında kullanılan hiperparametreler ve yöntemler şunlardır:

- Optimizer:
  - Adam Optimizer kullanılmıştır. Bu, hızlı öğrenme ve stabilite sağlamak için popüler bir optimizasyon yöntemidir.
- Loss Fonksiyonu:
  - Sparse Categorical Crossentropy kullanılmıştır. Bu fonksiyon, çok sınıflı sınıflandırma problemlerinde yaygın olarak tercih edilir.
- Epoch Sayısı:
  - Model, 10 epoch boyunca eğitilmiştir. Her bir epoch sırasında hem eğitim hem de doğrulama verisi üzerinde performans değerlendirilmiştir.
- Batch Size:
  - Her bir eğitim adımında 16 görüntü kullanılmıştır.
- Sınıf Ağırlıkları:
  - Veri setindeki dengesizliği telafi etmek için cleaned ve dirty sınıflarına özel ağırlıklar atanmıştır.

---

3. Veri Artırımı

Modelin daha iyi genelleme yapabilmesi için cleaned sınıfına veri artırımı uygulanmıştır. Aşağıdaki yöntemler kullanılmıştır:

- Görsellerin döndürülmesi (rotation).
- Görsellerin rastgele ölçeklendirilmesi (zoom).
- Görsellerin yatay çevrilmesi (horizontal flip).

Bu adımlar, veri setindeki az sayıda cleaned görselin çeşitlendirilmesini sağlamıştır.

---

4. Model Değerlendirme

Modelin performansını değerlendirmek için aşağıdaki yöntemler kullanılmıştır:

- Accuracy (Doğruluk):
  - Eğitim ve doğrulama verisi üzerinde modelin doğruluk oranı hesaplanmıştır.
- Confusion Matrix:
  - Modelin hangi sınıfları doğru veya yanlış sınıflandırdığı analiz edilmiştir.
- Loss (Kayıp):
  - Eğitim ve doğrulama kaybı takip edilmiştir. Modelin aşırı öğrenme (overfitting) yapıp yapmadığını anlamak için kayıp değerleri incelenmiştir.

Sonuçlar ve Değerlendirme

Modelin genel doğruluğu %97.3 olarak hesaplanmıştır. Aşağıdaki confusion matrix ile modelin performansı görselleştirilmiştir: