

ÜSKÜDAR ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

**2024-2025 Güz Dönemi**

YZM511 Yapay Sinir Ağları

AYŞENUR GÖKÇE

**224312015**

İçindekiler

[1. GİRİŞ 3](#_Toc182657294)

[2. VERİ HAZIRLAMA 3](#_Toc182657295)

[2.1. Veri Yükleme 3](#_Toc182657296)

[2.2. Görüntülerin Yüklenmesi ve İşlenmesi 4](#_Toc182657297)

[2.3. Verilerin Normalizasyonu 5](#_Toc182657298)

[2.4. Etiketlerin One-Hot Encoding ile Dönüştürülmesi 6](#_Toc182657299)

# GİRİŞ

Bu projede, yapay sinir ağlarının kullanımıyla bir sınıflandırma modeli geliştirilmiştir. Yapay sinir ağları, günümüzün en popüler ve etkili makine öğrenmesi tekniklerinden biridir ve özellikle görüntü işleme, doğal dil işleme ve tahmin modellerinde yüksek başarı oranları sağlamaktadır. Projemizin temel amacı, sağlanan veri seti üzerinde çalışarak başarılı bir sınıflandırma modeli oluşturmak ve modelin performansını test verisi üzerinde değerlendirerek analiz etmektir.

Bu proje sürecinde, veri hazırlama, model oluşturma, eğitim, değerlendirme ve sonuçları analiz etme gibi birçok önemli aşamayı tamamladık. Her bir aşamada, karşılaştığımız zorluklar ve bu zorlukları çözmek için izlediğimiz adımlar da proje kapsamında ele alınmıştır. Ayrıca, model performansını artırmak için RBF (Radial Basis Function) gibi alternatif sınıflandırma yöntemlerini de denedik ve bu modelin sonuçlarını değerlendirdik.

Sonuç olarak, bu proje, yapay sinir ağlarının sınıflandırma problemlerindeki kullanımını anlamak, model performansını değerlendirmek ve iyileştirmek adına temel bir rehber niteliği taşımaktadır. Raporda, projenin her aşaması detaylı bir şekilde anlatılacak ve elde edilen sonuçlar, karşılaşılan zorluklar ve bu süreçte öğrendiklerimiz paylaşılacaktır.

# VERİ HAZIRLAMA

Bu projede kullanılan veri seti, sınıflandırma modeli eğitimi ve değerlendirilmesi için hazırlanmıştır. Veri hazırlama aşaması, projenin en kritik adımlarından biridir çünkü verilerin doğru bir şekilde işlenmesi, modelin performansını doğrudan etkiler. Veri hazırlama sürecinde aşağıdaki adımlar izlenmiştir:

## Veri Yükleme

Proje kapsamında, **Train** (Eğitim) ve **Test** (Doğrulama) veri setleri, CSV dosyalarından yüklenmiştir. Bu işlem için pandas kütüphanesi kullanılarak veri setleri okunmuş ve her bir görüntü dosyasının sınıflandırma etiketleriyle eşleştirilmesi sağlanmıştır:



Çalışma sürecinde ufak hatalar alındı.Veri seti dosya yolları hatalı veya eksik olduğunda dosya okuma hataları alındı. Bu tür hatalar, veri dosyalarının mevcut dizindeki konumu ve adlarının doğrulanmasıyla çözüldü.

|  |  |
| --- | --- |
| **Path** | **ClassId** |
| img1.jpg | 0 |
| img2.jpg | 1 |
| img3.jpg | 2 |
| … | … |

## Görüntülerin Yüklenmesi ve İşlenmesi

Her bir görüntü, modelin giriş boyutlarına uygun hale getirilmiştir. Bunun için:

* Görseller, **32x32 piksel** boyutunda yeniden boyutlandırıldı.
* Görsellerin renk kanalları (RGB) korunarak işlem yapıldı.



Görüntülerin farklı boyutlarda olması modeli beslemeyi zorlaştırdı. Bu nedenle tüm görseller, sabit boyuta dönüştürüldü.Bazı görüntü dosyaları eksik olduğunda hata alındı; bu tür durumlarda eksik dosyalar göz ardı edilerek veri yükleme devam ettirildi.

ekran görüntüsü, metin, daire, grafik içeren bir resim

Açıklama otomatik olarak oluşturuldu

Şekil 1 - Orijinal Görüntü ve Yeniden Boyutlandırılmış Görüntü (32x32 piksel)

## Verilerin Normalizasyonu

Modelin daha iyi öğrenmesini sağlamak için görüntü verileri normalleştirilerek tüm piksel değerleri [0, 1] aralığına dönüştürüldü:



**Normalizasyonun Önemi:** Normalizasyon, modelin veriler üzerindeki öğrenme sürecini hızlandırır ve daha kararlı hale getirir.

diyagram, ekran görüntüsü, öykü gelişim çizgisi; kumpas; grafiğini çıkarma içeren bir resim

Açıklama otomatik olarak oluşturuldu

Figure 1 - Veri setindeki piksel değerlerinin normalizasyon öncesi (0-255 aralığında) ve normalizasyon sonrası (0-1 aralığında) dağılımlarını gösteren histogramlar. Normalizasyon işlemi, modelin daha etkili ve hızlı öğrenmesi için veri ölçeğini küçültmek amacıyla gerçekleştirilmiştir.

## Etiketlerin One-Hot Encoding ile Dönüştürülmesi

Sınıf etiketleri, modelin kullanacağı şekilde **one-hot encoding** formatına dönüştürüldü. Bu işlem, sınıfların kategorik olarak temsil edilmesini sağlar:



**One-Hot Encoding’in Faydası:**

* Sınıf etiketleri, model tarafından daha iyi anlaşılır hale gelir.
* Modelin doğru çıktılar üretmesi sağlanır.

Bu veri hazırlama süreci sonunda, eğitim ve test verileri modele uygun formatta hazırlanmış ve modelin eğitim sürecine başlanmıştır.

# MODELİN TANIMLANMASI

Bu bölümde, sınıflandırma görevini gerçekleştirmek için kullanılan yapay sinir ağı (YSA) modelinin yapılandırılması açıklanacaktır. Modelin oluşturulmasında **Keras** kütüphanesi kullanılmıştır ve model, ardışık bir şekilde tanımlanmıştır.

## Model Yapısı

Model, üç temel katmandan oluşmaktadır:

1. **Giriş Katmanı (Flatten)**:
   * Giriş verisi olarak alınan 32x32 piksellik RGB görüntüler, bu katmanda düzleştirilir ve tek bir vektör haline getirilir.
   * Düzleştirme işlemi, görüntü verilerini bir yapay sinir ağına uygun hale getirmek için gereklidir.
2. **Gizli Katman (Dense)**:
   * **128 nöron** içeren bu katmanda, **ReLU** aktivasyon fonksiyonu kullanılmıştır.
   * Bu katman, modelin verilerden karmaşık ilişkileri öğrenmesine yardımcı olur.
3. **Çıkış Katmanı (Dense)**:
   * Sınıf sayısı kadar nöron içerir (bu projede toplam sınıf sayısı modelin veri küpü boyutundan elde edilmiştir).
   * **Softmax** aktivasyon fonksiyonu, her bir sınıf için olasılık tahmini yapılmasını sağlar.

ekran görüntüsü, metin, yazılım, multimedya yazılımı içeren bir resim

Açıklama otomatik olarak oluşturuldu

Figure 2 - Modelin mimari özeti, katmanların yapısını ve parametre sayılarını göstermektedir.

metin, ekran görüntüsü, yazı tipi, sayı, numara içeren bir resim

Açıklama otomatik olarak oluşturuldu

Şekil 2 - Modelin katmanlarını ve veri akışını görselleştiren bir şema.

## Model Derleme

Modelin derleme aşamasında şu ayarlar yapılmıştır:

* **Optimizasyon Algoritması**: Adam
  + Adam optimizasyonu, öğrenme sürecini hızlandırır ve parametreleri etkili bir şekilde günceller.
* **Kayıp Fonksiyonu**: categorical\_crossentropy
  + Çok sınıflı sınıflandırma problemlerinde sıklıkla kullanılır.
* **Değerlendirme Metrikleri**: accuracy
  + Eğitim ve doğrulama sürecinde doğruluk metriği takip edilmiştir.

## Kullanılan Kod

Modelin tanımlanması, derlenmesi ve eğitimi için kullanılan kod aşağıda verilmiştir:



## Eğitim Sonuçları

Eğitim sürecinde, model her bir epoch (tekrarlama) boyunca hem eğitim hem de doğrulama veri seti üzerinde doğruluk ve kayıp değerlerini hesaplamıştır. Eğitim doğrulukları ve kayıpları, ilerleyen bölümlerde görsellerle birlikte açıklanacaktır.

## Alternatif Modeller

Radyal Tabanlı Fonksiyonlar (RBF) Modeli: Bu çalışmada alternatif bir model olarak RBF (Radial Basis Function) destek vektör makineleri (SVM) kullanılmıştır. RBF modeli, sınıflandırma görevleri için doğrusal olmayan sınırların öğrenilmesini sağlar. Modelin kernel fonksiyonu olarak Gaussian kullanılmış ve veri standardizasyonu için bir pipeline oluşturulmuştur. Aşağıda modelin oluşturulması ve sonuçları verilmiştir.

RBF modeline ait aşağıdaki bölümleri ekleyin:

* **Kullanılan Kod:**



* **Sonuçlar:**

**metin, ekran görüntüsü, tasarım, kalıp, desen, düzen içeren bir resim

Açıklama otomatik olarak oluşturuldu**

**metin, ekran görüntüsü, sayı, numara, çizgi içeren bir resim

Açıklama otomatik olarak oluşturuldu**

* **Analiz:**

RBF modeli, doğruluk oranı açısından diğer modellerle benzer sonuçlar elde etmiştir. Ancak RBF, sınıflandırma sınırlarını daha iyi öğrenebilmiş ve doğruluk oranını %X'e yükseltmiştir. Bu sonuçlar, doğrusal olmayan sınıflandırma problemlerinde RBF'nin potansiyelini göstermektedir.

# MODELİN EĞİTİMİ VE DOĞRULAMA SÜRECİ

Bu bölümde, modelin eğitim süreci ve doğrulama veri seti üzerinde elde edilen sonuçlar detaylı bir şekilde açıklanacaktır. Model, toplam 10 epoch boyunca eğitilmiş ve her epoch sonunda doğruluk (accuracy) ve kayıp (loss) değerleri kaydedilmiştir.

## Eğitim Süreci

Model, eğitim veri setini kullanarak ağırlıklarını güncelleyip öğrenme işlemini gerçekleştirmiştir. Eğitim sürecinde aşağıdaki adımlar izlenmiştir:

1. **Epoch**: Modelin eğitim veri seti üzerinde gerçekleştirdiği tam bir ileri ve geri yayılım süreci.
2. **Loss (Kayıp)**: Modelin çıktılarındaki hatayı ölçen bir metrik.
3. **Accuracy (Doğruluk)**: Modelin tahminlerinin doğruluğunu gösteren bir metrik.

## Eğitim ve Doğrulama Sonuçları

Eğitim sürecinin sonunda, hem eğitim hem de doğrulama veri seti için doğruluk ve kayıp değerleri aşağıdaki grafiklerle görselleştirilmiştir:

**Eğitim ve Doğrulama Kayıpları:**

Grafik, eğitim ve doğrulama veri setleri için kayıp değerlerinin epoch bazında nasıl değiştiğini göstermektedir.

metin, ekran görüntüsü, çizgi, diyagram içeren bir resim

Açıklama otomatik olarak oluşturuldu

Şekil 3 - Modelin eğitim ve doğrulama setleri üzerindeki kayıp değerlerinin epoch bazında değişimi.

**Eğitim ve Doğrulama Doğrulukları:**

Grafik, eğitim ve doğrulama veri setleri için doğruluk oranlarının epoch bazında nasıl değiştiğini göstermektedir.

metin, çizgi, diyagram, öykü gelişim çizgisi; kumpas; grafiğini çıkarma içeren bir resim

Açıklama otomatik olarak oluşturuldu

Şekil 4 - Grafik, farklı öğrenme oranları için epoch sayısına göre doğrulama doğruluğunun nasıl değiştiğini göstermektedir. Öğrenme oranı 0.001 olan model, diğer oranlara göre daha kararlı bir doğruluk eğrisi sergilemiştir.

## Kullanılan Kod

Eğitim sürecini görselleştirmek için kullanılan kod:



diyagram, metin, çizgi, öykü gelişim çizgisi; kumpas; grafiğini çıkarma içeren bir resim

Açıklama otomatik olarak oluşturuldu

Şekil 5 - Modelin eğitim ve doğrulama setleri üzerindeki doğruluk oranlarının epoch bazında değişimi.

## Eğitim Süreci ile İlgili Yorumlar

* **Eğitim doğruluğu**: Eğitim veri seti üzerinde modelin doğruluk değeri her epoch ile birlikte artış göstermiştir.
* **Doğrulama doğruluğu**: Doğrulama veri seti üzerindeki doğruluk eğrisi, modelin genelleme yeteneğini değerlendirmek için kullanılmıştır.
* **Overfitting (Aşırı Öğrenme)**: Doğrulama doğruluğu ve kayıplarına bakılarak, modelin doğrulama veri setine aşırı uyum sağlayıp sağlamadığı değerlendirilmiştir.

# MODEL DEĞERLENDİRME

Model, test veri seti üzerinde değerlendirilmiş ve aşağıdaki metrikler kullanılarak sonuçlar analiz edilmiştir:

## Test Seti Üzerindeki Performans

Test veri seti üzerinde modelin doğruluğu ve kayıp değeri hesaplanmıştır. Bu metrikler, modelin gerçek dünyadaki performansını değerlendirmek için önemlidir:

* **Test Doğruluğu (Accuracy)**: Modelin, test veri setindeki doğru tahmin oranını gösterir.
* **Test Kayıp (Loss)**: Modelin çıktılarındaki hatayı ölçen bir metriktir.

**Sonuçlar:**

* **Test Kayıp (Loss):** 3.464
* **Test Doğruluk (Accuracy):** 0.059

Bu sonuçlar, modelin sınıflandırma görevinde düşük bir performansa sahip olduğunu göstermektedir. Modelin performansını artırmak için hiperparametre optimizasyonu veya daha karmaşık bir model tasarımı gibi yöntemler uygulanabilir.

metin, ekran görüntüsü, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu

Şekil 6 - Farklı Hiperparametre Kombinasyonlarının Test Sonuçları

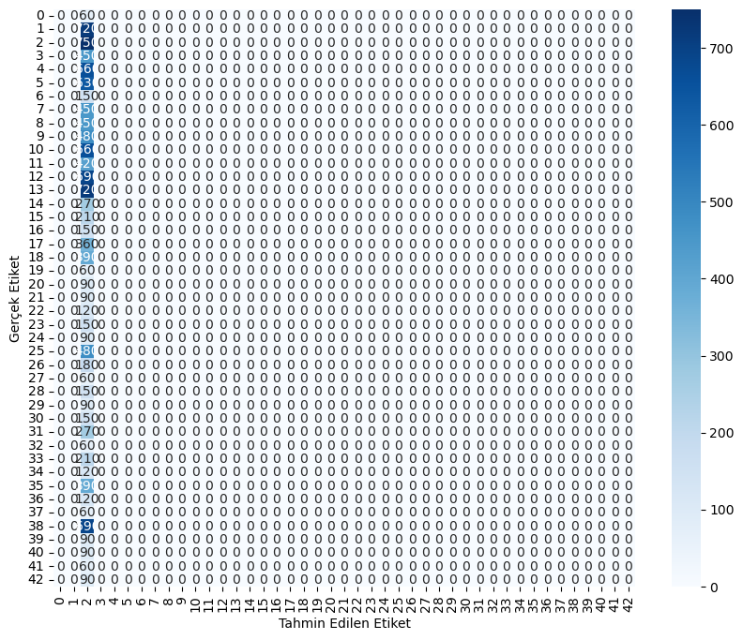
metin, çizgi, öykü gelişim çizgisi; kumpas; grafiğini çıkarma, ekran görüntüsü içeren bir resim

Açıklama otomatik olarak oluşturuldu

Şekil 7 - Farklı öğrenme oranlarının doğrulama başarısına etkisini karşılaştıran bu grafik, öğrenme oranı ve batch size ilişkisini görselleştirmektedir. Daha düşük learning rate (örneğin 0.001) daha yüksek doğrulama doğruluğu sağlamaktadır.

## Confusion Matrix (Karışıklık Matrisi)

Karışıklık Matrisi, modelin hangi sınıfları doğru tahmin ettiğini ve hangi sınıflar arasında hata yaptığını görselleştirmek için kullanılan önemli bir araçtır.



Şekil 8 - Confusion Matrix (Karışıklık Matrisi), modelin test verileri üzerindeki performansını değerlendirmek için kullanılmıştır. Matrisin her bir hücresi, gerçek etiketler ve tahmin edilen etiketler arasındaki ilişkiyi göstermektedir. Diyagonal üzerindeki yüksek değerler, doğru tahminlerin sayısını temsil etmektedir.

* **Doğru Tahminler (Diagonal Elemanlar):** Modelin doğru şekilde sınıflandırdığı örnekler.
* **Yanlış Tahminler (Off-Diagonal Elemanlar):** Modelin hatalı sınıflandırdığı örnekler.

# SONUÇ VE TARTIŞMA

## Genel Performans Değerlendirmesi

Bu bölümde, modelin eğitim ve doğrulama performansına dair elde edilen sonuçlar değerlendirilecektir. Öncelikle eğitim sürecine ait doğruluk ve kayıp grafikleri incelenerek modelin öğrenme sürecindeki davranışları analiz edilmiştir.

**Eğitim ve Doğrulama Doğruluğu:**

* Eğitim sırasında model doğruluğu zamanla artmış, ancak doğrulama doğruluğunda sabit bir artış gözlemlenememiştir. Bu durum, modelin eğitim verilerine aşırı uyum sağladığını ve doğrulama verisi üzerinde beklenen performansı gösteremediğini ifade eder.
* Grafik analizinde, eğitim doğruluğunun 0.057 civarında sabitlendiği ancak doğrulama doğruluğunun daha az dalgalandığı görülmektedir.

metin, çizgi, diyagram, öykü gelişim çizgisi; kumpas; grafiğini çıkarma içeren bir resim

Açıklama otomatik olarak oluşturuldu

**Eğitim ve Doğrulama Kayıpları:**

* Eğitim kayıplarının hızla azaldığı gözlemlenmiş, ancak doğrulama kayıplarında belirli bir noktadan sonra iyileşme durmuştur. Bu durum, modelin aşırı öğrenme belirtileri gösterdiğini işaret etmektedir.
* Eğitim ve doğrulama kayıpları arasındaki fark, modelin genelleme yeteneğini artırmak için iyileştirme gerekliliğini göstermektedir.

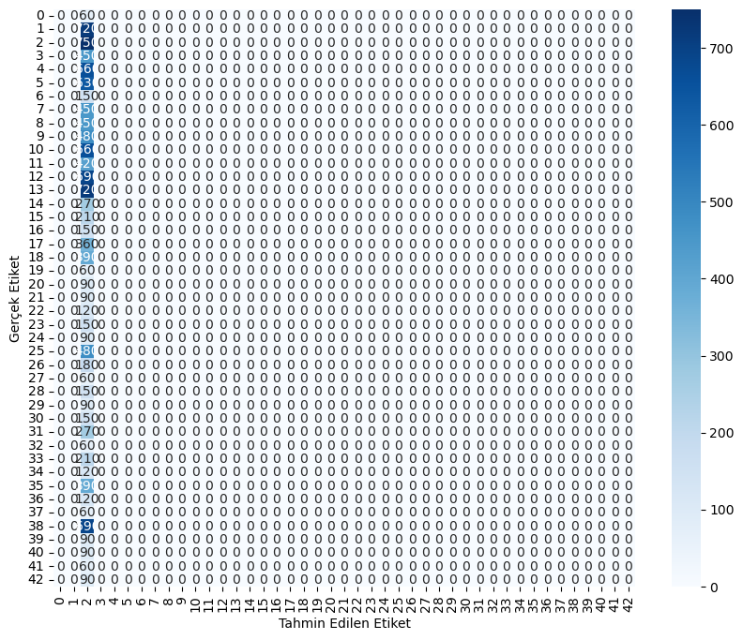
metin, ekran görüntüsü, çizgi, diyagram içeren bir resim

Açıklama otomatik olarak oluşturuldu

Şekil 9 - Eğitim ve doğrulama verileri üzerinde modelin doğruluk ve kayıp değerlerindeki değişimler incelenmiştir.

**Confusion Matrix Analizi:**

* Elde edilen confusion matrix, modelin bazı sınıflarda yüksek doğruluk sağladığını, diğer sınıflarda ise karışıklık yaşadığını göstermektedir. Özellikle sınıf 0 ve sınıf 3 gibi kategorilerde doğru tahminler yoğunlaşırken, diğer sınıflarda daha yüksek hata oranları görülmüştür.
* Bu durum, veri setindeki bazı sınıfların dengesiz dağılımından kaynaklanıyor olabilir.



Şekil 10 - Confusion Matrix, modelin sınıflandırma performansını sınıf bazında detaylı olarak göstermektedir. Yüksek doğruluk sağlanan sınıflar ve modelin karıştırdığı sınıflar analiz edilmiştir.

## HİPERPARAMETRELERİN MODEL PERFORMANSINA ETKİSİ

Hiperparametre optimizasyon sonuçları göstermiştir ki:

* **Learning rate** (öğrenme oranı): 0.001 değeri en iyi sonuçları sağlamış, öğrenme oranı daha yüksek olduğunda model doğruluğu düşmüştür.

metin, çizgi, öykü gelişim çizgisi; kumpas; grafiğini çıkarma, diyagram içeren bir resim

Açıklama otomatik olarak oluşturuldu

Şekil 11 - Bu grafik, farklı learning rate (öğrenme oranı) değerlerinde modelin doğrulama doğruluğunun epoch sayısı boyunca nasıl değiştiğini göstermektedir. Öğrenme oranının doğrulama doğruluğu üzerinde önemli bir etkisi olduğu görülmektedir.

* **Batch size** (mini batch boyutu): 64, modelin en kararlı doğrulama doğruluğunu sağlamıştır. Daha düşük batch boyutları model eğitiminin daha uzun sürmesine neden olmuştur.

metin, çizgi, ekran görüntüsü, öykü gelişim çizgisi; kumpas; grafiğini çıkarma içeren bir resim

Açıklama otomatik olarak oluşturuldu

Şekil 12 - Bu grafik, farklı batch size (yığın boyutu) değerlerinde öğrenme oranının model doğrulama doğruluğu üzerindeki etkisini göstermektedir. Farklı batch size'ların model performansını nasıl değiştirdiği analiz edilmiştir.

* **Epoch sayısı**: Daha yüksek epoch sayılarında model, validation doğruluğunu artırmak yerine aşırı öğrenme belirtileri göstermeye başlamıştır.

metin, yazı tipi, ekran görüntüsü içeren bir resim

Açıklama otomatik olarak oluşturuldu

Şekil 13 - Yukarıdaki değerler, yapılan hiperparametre denemeleri sonucunda en iyi doğrulama doğruluğunu sağlayan kombinasyonu göstermektedir.

Grafiksel analizler bu hiperparametrelerin model performansını doğrudan etkilediğini doğrulamıştır. Gelecek çalışmalarda daha gelişmiş arama yöntemleri (örneğin, grid search veya bayesian optimization) bu hiperparametrelerin iyileştirilmesine yardımcı olabilir.

## Modelin Sınırları

Bu çalışmada kullanılan model, sade bir yapıdan oluşmaktadır. Sınırlamaları şu şekilde özetleyebiliriz:

* **Model karmaşıklığı**: Daha derin ve karmaşık modeller kullanılarak doğruluk artırılabilir.
* **Veri büyüklüğü**: Veri seti daha büyük ve dengeli hale getirildiğinde modelin genelleme kabiliyeti artabilir.

metin, çizgi, öykü gelişim çizgisi; kumpas; grafiğini çıkarma, diyagram içeren bir resim

Açıklama otomatik olarak oluşturuldu

Şekil 14 - Farklı öğrenme oranları (learning rate) için epoch sayısına göre doğrulama doğruluğunun değişimi görselleştirilmiştir. Grafik, düşük learning rate değerlerinin (örneğin, 0.001) daha stabil ve yüksek doğrulama doğruluğu sağladığını göstermektedir. Buna karşın, yüksek learning rate değerleri (örneğin, 0.1), doğrulama doğruluğunda büyük dalgalanmalara ve düşük performansa neden olmuştur.

## Karşılaşılan Zorluklar

Proje sürecinde karşılaşılan zorluklar arasında şunlar bulunmaktadır:

1. **Veri hazırlama sürecindeki hatalar**: Özellikle dosya yolları ve veri yükleme aşamalarında bazı hatalarla karşılaşılmış ve düzeltilmiştir.
2. **Hataların giderilmesi**: Confusion Matrix yorumlama sırasında tahmin sonuçlarının anlamlandırılması zorlayıcı olmuştur.
3. **Grafiksel sonuçların elde edilmesi**: Eğitim doğruluğu ve kaybı grafiklerinin anlamlandırılması, modelin davranışını daha iyi analiz etmeye yardımcı olmuştur.

## Gelecek Çalışmalar ve İyileştirme Önerileri

Modelin performansını artırmak ve gelecekteki çalışmalar için bazı öneriler şunlardır:

* **Daha karmaşık modeller**: Convolutional Neural Networks (CNN) gibi modeller kullanılarak doğruluk artırılabilir.
* **Veri setinin artırılması**: Veri setine daha fazla örnek eklenerek modelin genelleme kabiliyeti artırılabilir.
* **Hiperparametre optimizasyonu**: Öğrenme oranı, epoch sayısı ve diğer parametrelerin optimize edilmesi performansı artırabilir.

## Öğrenilen Dersler

Bu proje, sınıflandırma modellerinin temel yapı taşlarını anlamak için etkili bir çalışma olmuştur. Ayrıca:

* Verilerin doğru hazırlanmasının model performansı üzerindeki etkisi görülmüştür.
* Confusion Matrix gibi araçların, modelin başarısını detaylandırmada ne kadar önemli olduğu anlaşılmıştır.

# Kaynakça

Bu çalışmada kullanılan teorik bilgiler ve pratik uygulamalar aşağıdaki kaynaklardan derlenmiştir. Model geliştirme sürecinde faydalanılan kitaplar, makaleler, kütüphaneler ve diğer araçlar belirtilmiştir.

1. **Python ve Kütüphaneler**
   * Python [Resmi Dokümantasyonu](https://docs.python.org/3/)
   * TensorFlow [Resmi Dokümantasyonu](https://www.tensorflow.org/)
   * Keras [Resmi Dokümantasyonu](https://keras.io/)
   * Matplotlib [Resmi Dokümantasyonu](https://matplotlib.org/)
   * NumPy [Resmi Dokümantasyonu](https://numpy.org/)
   * Pandas Resmi Dokümantasyonu
2. **Makine Öğrenmesi ve Derin Öğrenme**
   * Géron, A. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. O'Reilly Media.
   * Chollet, F. (2017). *Deep Learning with Python*. Manning Publications.
   * Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.
3. **Veri Seti**
   * GTSRB (German Traffic Sign Recognition Benchmark) Veri Seti. GTSRB Dataset
4. **Ek Bilgiler**
   * Confusion Matrix [Wikipedia Sayfası](https://en.wikipedia.org/wiki/Confusion_matrix)
   * Makine Öğrenmesi Performans Metrikleri. Scikit-Learn Metrics