



**MÜHENDİSLİK VE DOĞA BİLİMLERİ FAKÜLTESİ**  
**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**  
**VERİ BİLİMİNE GİRİŞ DERSİ FİNAL PROJESİ**

**Derin Öğrenme ile Görüntü Sınıflandırma**

23120205032, Sabahattin TAN

23120205010, Mehmet Emin DURMUŞ

23120205067, Taha YÜCEL

23120205061, Mehmet Efe KAYACIK

23120205012, Selim ÖZTÜRK

**Ders Sorumlusu**

Dr. Öğr. Üyesi Muhammet Sinan BAŞARSLAN

Ocak, 2026

İstanbul Medeniyet Üniversitesi, İstanbul

# İÇİNDEKİLER

	Sayfa No
İÇİNDEKİLER	i
ŞEKİL LİSTESİ	ii
KISALTMALAR	iii
ÖZET	iv
SUMMARY	v
1. 1	
2. 1	
3. 1	
3.1 Kullanılan Araçlar ve Kütüphaneler	1
3.2 Veri Seti ve Ön İşleme	2
3.3 Model Mimarisi	2
3.4 Eğitim Süreci	2
4. <b>Hata! Yer işareti tanımlanmamış.</b>	
4.1 Eğitim Süreci ve Öğrenme Eğrileri	2
4.2 Test Verisi Üzerindeki Performans	4
5. <b>Hata! Yer işareti tanımlanmamış.</b>	
6. KAYNAKÇA	6
7. EKLER	6

# ŞEKİL LİSTESİ

	<b><u>Sayfa No</u></b>
Şekil 1 Bazı Geniş Çaplı Görsel Sınıflandırma Çalışmaları	1
Şekil 2 <i>Epoch</i> Başına Eğitim Kayıpları	3
Şekil 3 <i>Epoch</i> Başına Doğruluk Oranı	4
Şekil 4 Hata Matrisi	5

## KISALTMALAR

- **CNN** = Convolutional Neural Networks  
Evrişimsel Sinir Ağları
- **ResNet** = Residual Networks
- **VGG** = Visual Geometry Group

## ÖZET

Bu projede, evrişimsel sinir ağları (*Convolutional Neural Networks* – CNN) kullanılarak görüntü sınıflandırma problemi ele alınmıştır. Çalışma kapsamında, etiketlenmiş bir görüntü veri seti üzerinde derin öğrenme tabanlı bir model eğitilmiş ve modelin başarımı ayrı bir test veri seti üzerinde değerlendirilmiştir. Eğitim sürecinde aşırı öğrenme (*overfitting*) problemi gözlemlenmiş, bu problemi azaltmak amacıyla veri artırma (*data augmentation*), *dropout* ve L2 regularizasyon teknikleri uygulanmıştır. Modelin eğitim sürecine ait kayıp (*loss*) ve doğruluk (*accuracy*) değerleri kaydedilmiş ve sonrasında grafikler yardımıyla analiz edilmiştir. Elde edilen sonuçlar, uygulanan iyileştirmelerin modelin genelleme yeteneğini artırdığını göstermektedir.

**Anahtar Kelimeler:** Derin Öğrenme, CNN, Görüntü Sınıflandırma, PyTorch, Veri Artırma.

## SUMMARY

In this project, an image classification problem was addressed using Convolutional Neural Networks (CNN). A deep learning model was trained on a labeled image dataset and evaluated on a separate test dataset. During training, overfitting was observed; therefore, data augmentation, dropout, and L2 regularization techniques were applied to mitigate this issue. Training loss and accuracy values were saved and later visualized for analysis. The results indicate that the applied improvements enhanced the model's generalization performance.

**Keywords:** Deep Learning, CNN, Image Classification, PyTorch, Data Augmentation.

# 1. GİRİŞ

Görüntü sınıflandırma, bilgisayarla görme ve veri bilimi alanlarında yaygın olarak karşılaşılan temel problemlerden biridir. Derin öğrenme yöntemlerinin gelişmesiyle birlikte, özellikle Evrişimsel Sinir Ağları bu alanda yüksek başarılar elde etmiştir. Bu projede amaç, gerçek bir veri seti üzerinde CNN tabanlı bir model geliştirerek modelin eğitim sürecini, karşılaşılan problemleri ve elde edilen sonuçları analiz etmektir.

## 2. LİTERATÜR TARAMA/İLGİLİ ÇALIŞMALAR

Görüntü sınıflandırma alanı, üzerine uzun süredir çalışmalar yapılan bir alandır fakat bu çalışmada bahsi geçen alan ve veri seti üzerine yapılan çalışmalar sınırlıdır. Bununla beraber, bazı geniş çaplı ve genel bazlı veri setleri ve görüntü sınıflandırma çalışmalarına verilebilecek örneklerin bazıları Şekil 1’deki gibidir.

Şekil 1 Bazı Geniş Çaplı Görsel Sınıflandırma Çalışmaları

ResNet ( <i>Residual Networks</i> )	<i>Microsoft Research</i> tarafından eğitilen, verimlilik ve isabet oranını yüksek tutmaya odaklanan çeşitli boyutlarda versiyonlara sahip bir model.
VGG ( <i>Visual Geometry Group</i> )	16 – 19 katmanlı derin modellere sahip, <i>University of Oxford</i> tarafından eğitilen bir model.
Inception	<i>Google</i> tarafından eğitilen isabet ve verimlilik dengesini yüksek tutmaya odaklanan bir başka model.

## 3. MATERYAL METOD

Bu bölümde çalışmada kullanılan yazılım araçları ve eğitim süreci açıklanmaktadır.

### 3.1 Kullanılan Araçlar ve Kütüphaneler

Proje kapsamında *Python* programlama dili tercih edilmiştir. Derin öğrenme tabanlı modelin geliştirilmesi için *PyTorch* kütüphanesi kullanılmıştır. *PyTorch*, esnek yapısı ve dinamik hesaplama grafiği sayesinde eğitim sürecinin kolayca kontrol edilmesine olanak sağlamaktadır. Görüntü veri setlerinin yüklenmesi ve ön işlenmesi için *torchvision* modülü kullanılmıştır. Elde edilen eğitim sonuçlarının görselleştirilmesi amacıyla *matplotlib* kütüphanesinden yararlanılmıştır. Model değerlendirme aşamasında ise doğruluk hesaplamaları ve karışıklık matrisi (*confusion matrix*) oluşturmak için *scikit-learn* kütüphanesinin ilgili modülleri kullanılmıştır.

### 3.2 Veri Seti ve Ön İşleme

Bu çalışmada kullanılan veri seti, sınıflara ayrılmış etiketli görüntülerden oluşmaktadır. Veri setindeki görüntüler farklı çözünürlük ve aydınlatma koşullarına sahip olduğundan, model eğitiminde önce standart bir ön işleme adımı uygulanmıştır. Tüm görüntüler 64×64 piksel boyutuna ölçeklendirilmiş ve piksel değerleri normalize edilmiştir.

Modelin genelleme yeteneğini artırmak ve aşırı öğrenme problemini azaltmak amacıyla eğitim sırasında veri artırma (*data augmentation*) teknikleri uygulanmıştır. Bu teknikler arasında rastgele yatay çevirme, küçük açılarda döndürme ve parlaklık/kontrast değişimleri yer almaktadır. Test aşamasında ise veri artırma uygulanmamış, yalnızca temel ön işleme adımları kullanılmıştır. Kaynak veri setinde bir miktar veri artırma tekniği kullanımı gözlenirse de yeterli bulunmamış ve çalışma sürecinde bu metotlar tekrar işlenmiştir.

### 3.3 Model Mimarisi

Geliştirilen CNN modeli, temel bir mimari üzerine kurulmuştur. Model iki adet evrişim (*convolution*) katmanı içermektedir. Bu katmanlar, giriş görüntülerinden kenar, köşe ve doku gibi temel özelliklerin çıkarılmasını sağlamaktadır. Her evrişim katmanından sonra ReLU aktivasyon fonksiyonu ve maksimum havuzlama (*max pooling*) katmanı kullanılmıştır. Evrişim katmanlarından elde edilen özellik haritaları düzleştirilerek tam bağlantılı (*fully connected*) katmanlara aktarılmıştır. Çalışma sırasında modelde aşırı öğrenme problemi gözlenmiş, önlemek amacıyla *dropout* tekniği uygulanmış ve bazı nöronlar eğitim sırasında rastgele devre dışı bırakılmıştır. Ayrıca optimizasyon sürecinde L2 regularizasyon (*weight decay*) kullanılarak modelin daha dengeli öğrenmesi sağlanmıştır.

### 3.4 Eğitim Süreci

Model, eğitim veri setinin tamamı kullanılarak belirlenen *epoch* (grup) sayısı boyunca eğitilmiştir. Kayıp fonksiyonu olarak çok sınıflı sınıflandırma problemleri için yaygın olarak kullanılan çapraz entropi kaybı (*CrossEntropyLoss*) tercih edilmiştir. Optimizasyon algoritması olarak *Adam* optimizasyon yöntemi kullanılmıştır. Her epoch sonunda ortalama eğitim kaybı ve doğruluk oranı daha sonra incelenmek üzere hesaplanıp kaydı tutulmuştur.

## 4. UYGULAMA

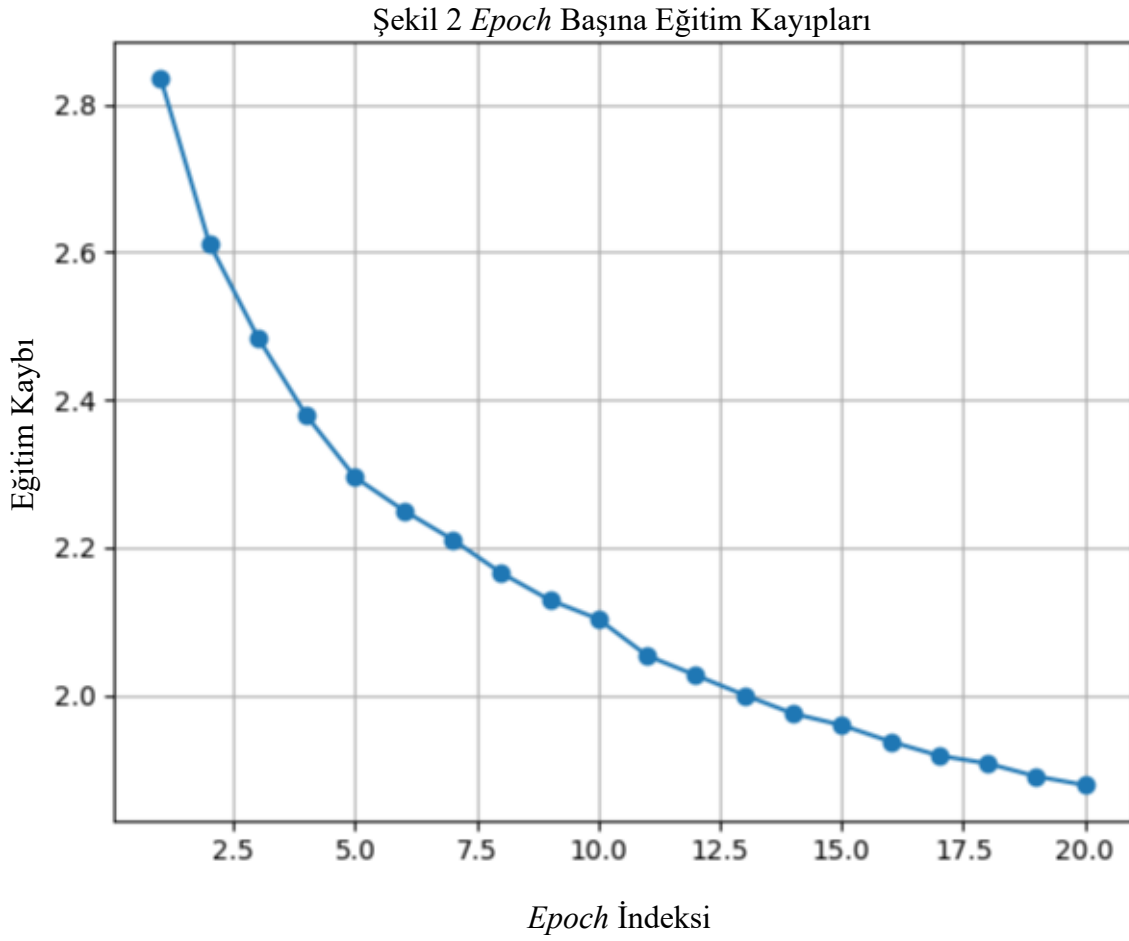
Bu bölümde geliştirilen modelin eğitim ve test aşamalarına ait çıktılar sunulmakta ve yorumlanmaktadır. Alınan veri setinin rastgele seçilen 20%'ye yakın miktarı, test verilerinde güvenilirlik açısından test etmek için kullanılmadan önce eğitim setinden çıkartılmıştır.

### 4.1 Eğitim Süreci ve Öğrenme Eğrileri

Model eğitimi sırasında her *epoch* sonunda eğitim kaybı (*training loss*) ve eğitim doğruluğu (*training accuracy*) hesaplanmış ve kaydedilmiştir. Şekil 2'de eğitim kaybının *epoch* 'lara bağlı olarak değişimi gösterilmektedir. Grafikten de görüldüğü üzere, eğitim sürecinin başında yüksek olan kayıp değeri *epoch* sayısı arttıkça düzenli bir şekilde azalmaktadır. Bu durum,

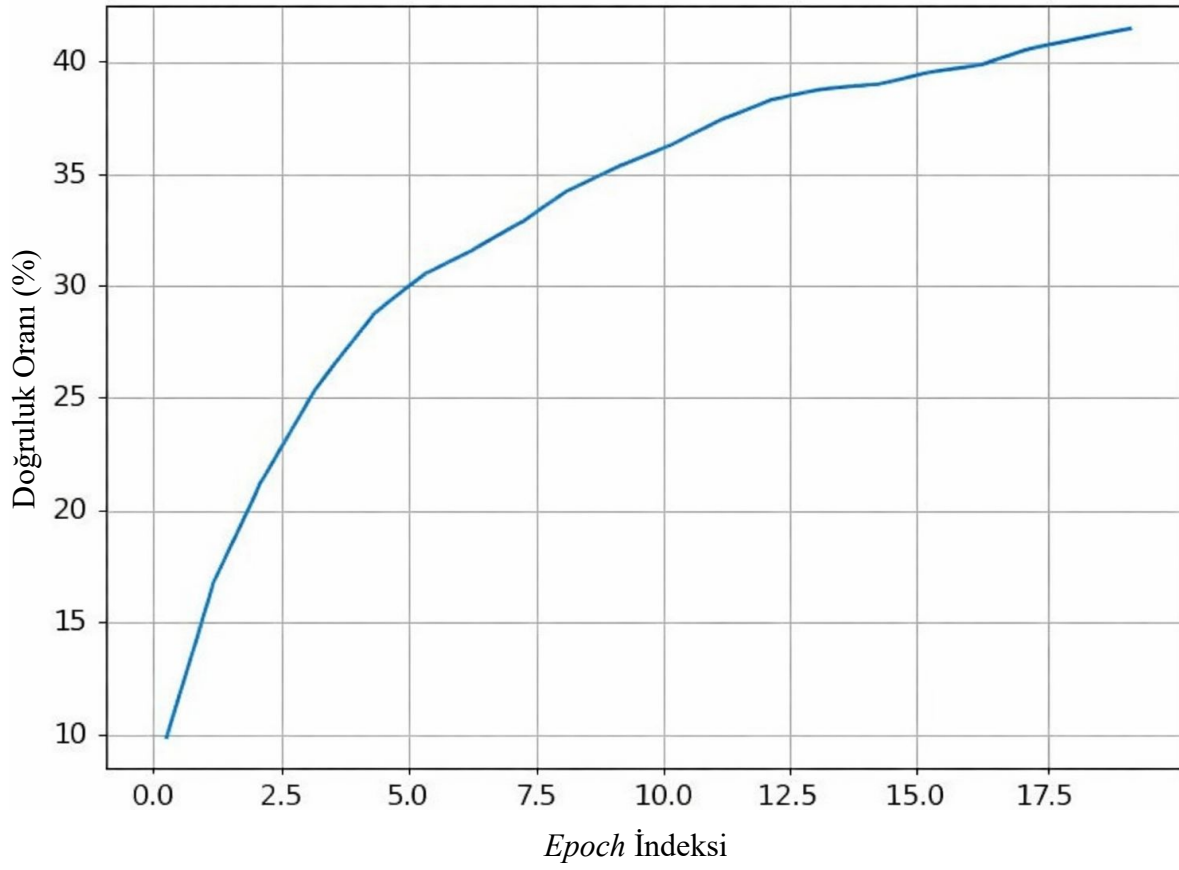


modelin eğitim verisi üzerindeki hatasını zamanla düşürdüğünü ve öğrenme sürecinin istikrarlı olduğunu göstermektedir



Şekil 3'te ise eğitim doğruluğunun *epoch*'lara göre değişimi verilmiştir. Eğitim doğruluğunun başlangıçta düşük olduğu, ancak *epoch* ilerledikçe sürekli artış gösterdiği gözlemlenmiştir. Bu artış, modelin veri setindeki sınıfları giderek daha iyi ayırt etmeye başladığını göstermektedir. Eğitim kaybı ve doğruluk eğrilerinin birlikte değerlendirilmesi, modelin aşırı dalgalanma göstermeden öğrenme sürecini sürdürdüğünü ortaya koymaktadır.

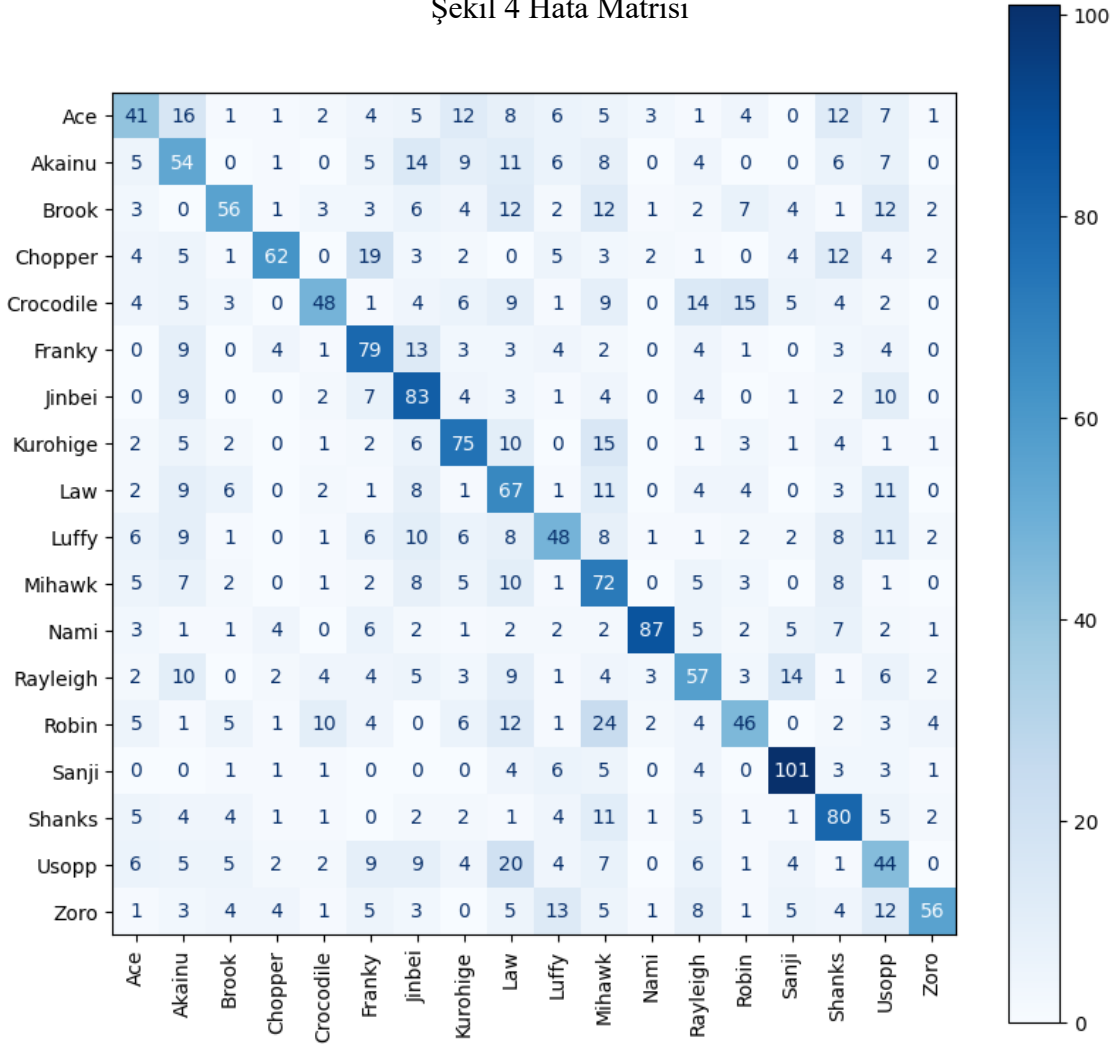
Şekil 3 *Epoch* Başına Doğruluk Oranı



#### 4.2 Test Verisi Üzerindeki Performans

Modelin genelleme yeteneğini değerlendirmek amacıyla, model orijinal veri setinin, eğitim sırasında kullanılmayan ve rastgele seçilen yaklaşık 20%'sine tekabül eden bir miktarı ile performanst testi yapılmıştır. Test sonuçları ile Şekil 4'te görülebilen hata matrisi (*confusion matrix*) oluşturulmuştur.

Şekil 4 Hata Matrisi



Şekil 4’te verilen karışıklık matrisi, her bir sınıf için doğru ve yanlış sınıflandırma sayılarını göstermektedir. Diyagonal üzerinde yer alan değerler, modelin ilgili sınıfı doğru tahmin ettiği örnekleri temsil etmektedir. Diyagonal dışındaki değerler ise modelin hangi sınıfları birbirleriyle karıştırdığını ortaya koymaktadır. Karışıklık matrisi incelendiğinde, bazı sınıfların yüksek doğrulukla tahmin edildiği, bazı sınıflarda ise benzer görsel özellikler nedeniyle karışıklık yaşandığı görülmektedir. Bu durum, veri setindeki sınıflar arasındaki görsel benzerliklerin model performansı üzerinde etkili olduğunu göstermektedir.

## 5. SONUÇ VE TARTIŞMA

Bu çalışmada, evrimsel sinir ağları kullanılarak bir görüntü sınıflandırma modeli geliştirilmiştir. İlk denemelerde modelin eğitim verisine aşırı uyum sağladığı ve test performansının düşük kaldığı gözlemlenmiştir. Bu durum, aşırı öğrenme probleminin varlığını ortaya koymuştur. Aşırı öğrenme problemini azaltmak amacıyla veri artırma, *dropout* ve L2 regularizasyon teknikleri uygulanmıştır. Yapılan bu iyileştirmeler sonucunda modelin genelleme yeteneği artmış ve test veri seti üzerindeki başarımlar belirgin şekilde iyileşmiştir.

Eğitim ve test sonuçları birlikte değerlendirildiğinde, uygulanan yöntemlerin etkili olduğu görülmüştür.

Bu proje kapsamında elde edilen deneyimler, derin öğrenme modellerinde yalnızca yüksek doğruluk elde etmenin yeterli olmadığını, aynı zamanda modelin genelleme kabiliyetinin de dikkate alınması gerektiğini göstermektedir.

## 6. KAYNAKÇA

- [1] İ. Serouis, One Piece Image Classifier, Kaggle Veri Seti, 2025. Çevrimiçi: <https://www.kaggle.com/datasets/ibrahimserouis99/one-piece-image-classifier>
- [2] PyTorch Geliştirici Ekibi, *PyTorch Documentation*. Çevrimiçi: <https://docs.pytorch.org/docs/stable/index.html>
- [3] PyTorch Geliştirici Ekibi, *torchvision Documentation*. Çevrimiçi: <https://docs.pytorch.org/vision/main/>
- [4] Scikit-learn Geliştirici Ekibi, *scikit-learn: Machine Learning in Python*. Çevrimiçi: <https://scikit-learn.org/stable/>.
- [5] Matplotlib Geliştirici Ekibi, *Matplotlib Documentation*. Çevrimiçi: <https://matplotlib.org/stable/>
- [6] I. Goodfellow, Y. Bengio, A. Courville, *Deep Learning*, MIT Press, Cambridge, MA, 2016.

## 7. EKLER

Çalışma dahilindeki *Python* programı, beraberinde eğitilen model, ve model hakkındaki veriler bu bağlantıda yer almaktadır:

<https://github.com/HsHappy/pytorch-one-piece-image-classifier>

Çalışma için kullanılan veri seti bu bağlantıda yer almaktadır:

<https://www.kaggle.com/datasets/ibrahimserouis99/one-piece-image-classifier>