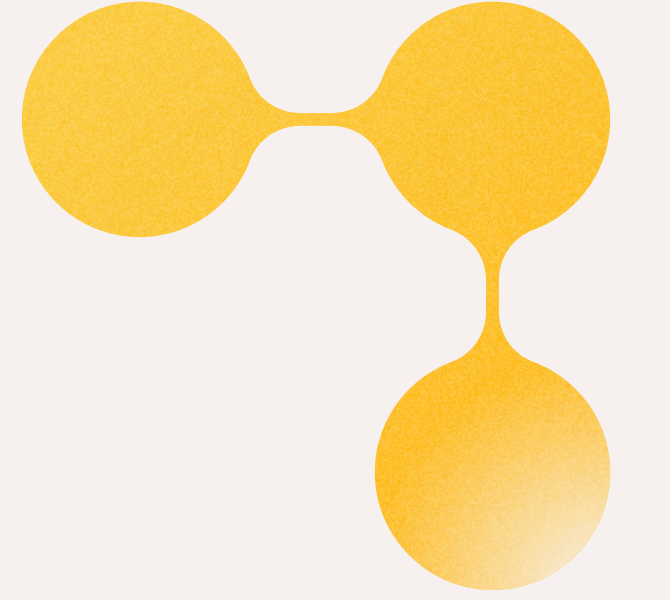


GÜLBELAZ BAYRAM ÖZER

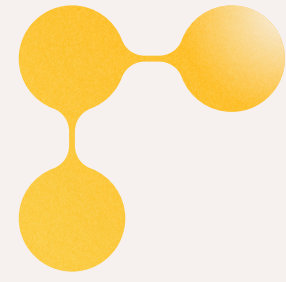
GİTHUB REPO: [HTTPS://GİTHUB.COM/GULBEYAZZB/WTECH-YAPAYZEKA](https://github.com/gulbeyazzb/wtech-yapayzeke)

WTECH YAPAY ZEKA EĞİTİMİ BİTİRME PROJESİ

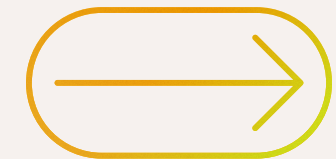


03.05.2024

İÇİNDEKİLER



- Veri Seti Seçimi
- Model Oluşturma
- Compile İşlemi
- Veri Ön İşleme ve Veri Arttırma
- Modelin Eğitilmesi
- FastApi Kodlaması



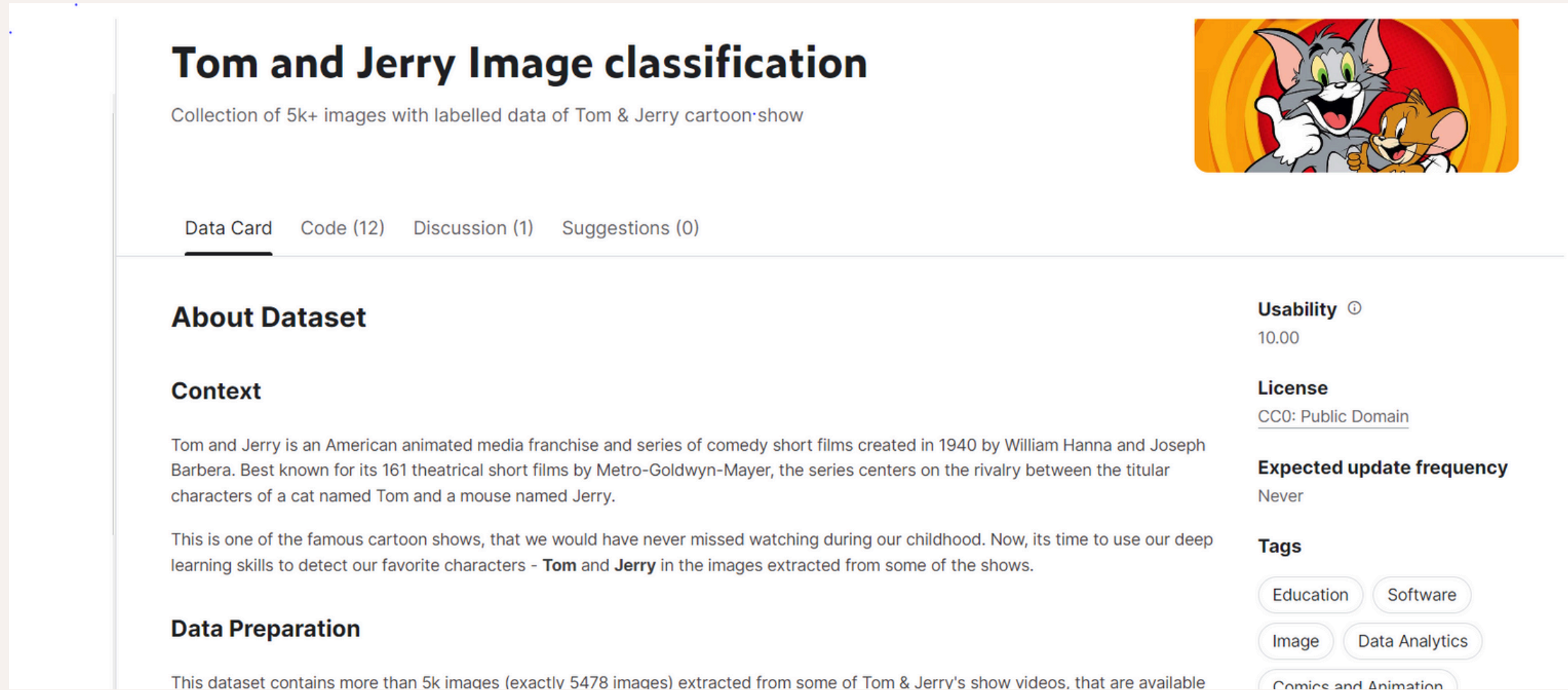
1. VERİ SETİ SEÇİMİ

Veri seti Kaggle'dan indirildi (Resim 1.1).

Kaggle link: <https://www.kaggle.com/datasets/balabaskar/tom-and-jerry-image-classification>

Veri seti tom ve jerry olmak üzere **iki kategoride** resimler içeriyor; **Tom** kategorisine ait **1,930**, **Jerry** kategorisine ait **1,240** adet resim içeriyor.

Görseller **RGB** formatında.



Resim 1.1. Kaggle'da Veri Seti

2.MODEL OLUŞTURMA

İlgili dataset görsel içerdiği için, görüntü sınıflandırmada tercih edilen Convolutional Neural Network yapısı kullanılmıştır.



Modelin özetini görseldeki gibidir.

```
model2.summary()
```

✓ 0.2s

Model: "sequential"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 238, 238, 64)	1792
max_pooling2d (MaxPooling2D)	(None, 119, 119, 64)	0
conv2d_1 (Conv2D)	(None, 117, 117, 128)	73856
max_pooling2d_1 (MaxPooling2D)	(None, 58, 58, 128)	0
conv2d_2 (Conv2D)	(None, 56, 56, 256)	295168
max_pooling2d_2 (MaxPooling2D)	(None, 28, 28, 256)	0
flatten (Flatten)	(None, 200704)	0
dense (Dense)	(None, 256)	51380480
dropout (Dropout)	(None, 256)	0
...		
Total params:	51792513 (197.57 MB)	
Trainable params:	51792513 (197.57 MB)	
Non-trainable params:	0 (0.00 Byte)	

Sınıflandırma problemleri için **genellikle son katmanda sigmoid** kullanılırken, **gizli katmanlarda** ReLU gibi aktivasyonlar daha yaygın olarak tercih edilir. Modelin eğitiminde yine bu aktivasyon yapısı tercih edildi.

Dataset büyük ve karmaşık olduğu için filtre sayısı 64'den başlatılarak her layerda 2 kat arttırıldı. Böylece detaylı görseller daha başarılı eğitildi.

Daha büyük bir pooling boyutu, daha az özetleme ve daha fazla bilgi kaybı anlamına gelir ve Küçük pooling boyutları, aşırı uyum riskini azaltabilir. Ayrıca görsel veriler için, genellikle 2x2 veya 3x3 boyutunda pooling katmanları tercih edilir. 2x2 boyutunda pooling katmanları oldukça yaygındır çünkü veriyi yarıya indirir ve hesaplama maliyeti düşüktür. Tüm bu nedenlerden dolayı 2x2 boyutu tercih edildi.

3

Resim 2.1. Model Özeti

2.MODEL OLUŞTURMA

```
> model2.summary()  
[7] ✓ 0.2s  
.. Model: "sequential"  


| Layer (type)                   | Output Shape          | Param #  |
|--------------------------------|-----------------------|----------|
| conv2d (Conv2D)                | (None, 238, 238, 64)  | 1792     |
| max_pooling2d (MaxPooling2D)   | (None, 119, 119, 64)  | 0        |
| conv2d_1 (Conv2D)              | (None, 117, 117, 128) | 73856    |
| max_pooling2d_1 (MaxPooling2D) | (None, 58, 58, 128)   | 0        |
| conv2d_2 (Conv2D)              | (None, 56, 56, 256)   | 295168   |
| max_pooling2d_2 (MaxPooling2D) | (None, 28, 28, 256)   | 0        |
| flatten (Flatten)              | (None, 200704)        | 0        |
| dense (Dense)                  | (None, 256)           | 51380480 |
| dropout (Dropout)              | (None, 256)           | 0        |

  
...  
Total params: 51792513 (197.57 MB)  
Trainable params: 51792513 (197.57 MB)  
Non-trainable params: 0 (0.00 Byte)
```

Dense katmanlarındaki nöron sayısı(units) modelin her bir katmanda daha fazla özellik ve karmaşıklığı öğrenmesini sağlar. önceki Conv2D ve MaxPooling2D katmanlarından elde edilen özelliklerin bir temsilini alabilmesi için başlangıç nöron sayısı.256 seçildi.

Dense katmanı aynı zamanda aşırı uyum riskini de artırabilir. Bu nedenle, Dropout katmanlarını kullanarak aşırı uyumu kontrol altında tutmaya çalışıyoruz.

Genellikle 0.2 veya 0.5 olarak tercih edilir. İlgili modelin eğitiminde de yine 0.5 tercih edildi.

Son katmanın sınıf sayısı 2 olduğu için ve çoğu ikili sınıflandırma problemi için standart bir yaklaşım olduğu için son katman dense değeri 1 olarak belirlendi



3.COMPILE İŞLEMİ

```
model.compile(optimizer="adam",loss="binary_crossentropy",metrics=["accuracy", Precision(), Recall()])
```

✓ 0.0s

Resim 3.1. Compile İşlemi Kod Satırı

optimizer="adam" yaygın olarak tercih edilen optimizasyon algoritmasıdır.

"binary_crossentropy", ikili sınıflandırma problemleri için yaygın olarak kullanılan bir kayıp fonksiyonudur.

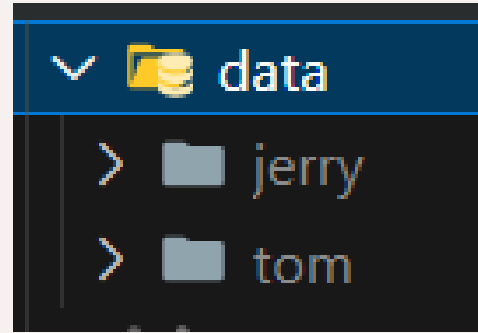
"accuracy" modelin doğruluğunu, **"Precision()"** hassasiyetini ve **"Recall()"** duyarlılığını hesaplar. modelin performansını ölçmek için kullanılan metriklerdir.



4. VERİ SETİ ÖN İŞLEME VE VERİ ARTTIRMA

Eğitim veri setinin çeşitliliğini artırmak ve modelin genelleme yeteneğini artırmak için **ImageDataGenerator** ile veri artırma işlemleri gerçekleştirildi.

Piksel değerlerini normalize ederek modelin eğitimini daha iyi yapmasını sağlamak için resim piksel değerlerini 0 ile 1 arasına ölçekleriz. Bu işlem için **rescale=1./255** yöntemi kullanıldı.



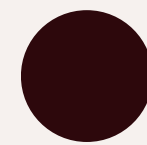
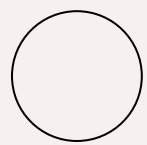
Resim 4.1. Veri Seti Klasörü

shear_range=0.2(kırpma), **zoom_range=0.2**(yakınlaştırma), **horizontal_flip=True**(çevirme) gibi özelliklerle resim arttırıldı.

validation_split=0.2 ile Veri setinin %20'sini doğrulama verisi olarak ayarlamak için kullanıldı. %20'lik bir doğrulama verisi oranı, genellikle modelin iyi performans göstermesini sağlayacak yeterli sayıda doğrulama örneğini sağlarken, hesaplama maliyetini de kontrol altında tutar.

flow_from_directory, veri setini yükler. Veri setinin her sınıfı bir klasör içinde bulunmalıdır, ve her klasörün adı sınıf etiketi olmalıdır (Resim 4.1).





4. VERİ SETİ ÖN İŞLEME VE VERİ ARTTIRMA

target_size=(240, 240): Resimlerin boyutu 240x240'a ölçeklenir. **input_shape (240,240)** ile genellikle aynı olmalı.

batch_size=32: Her bir eğitim ve doğrulama adımında kaç resmin işleneceğini belirtir. batch_size büyük seçilirse maliyet, aşırı uyumluluk, bellek tüketimi ve performans açısından olumsuz sonuçlara sebep olabileceğinden küçük denebilecek 32 değeri seçilmiştir.

subset='training' veya 'validation': Veri setinin hangi alt kümesinin kullanılacağını belirtir. 'training', eğitim verisi için kullanılırken, 'validation' doğrulama verisi için kullanılır.

class_mode='binary': Sınıflandırma türünü belirtir. "binary" olduğunda, modelin ikili sınıflandırma yapması beklenir (örneğin, 0 veya 1).



5.MODELİN EĞİTİMİ

```
model.fit(train_generator, epochs=30, validation_data=validation_generator)
```

Resim 5.1. Modelin Eğitimi

fit yöntemi, modelin veriye uyum sağlaması için kullanılır ve modelin kayıp fonksiyonunu minimize etmek, belirtilen metrikleri maksimize etmek için ağırlıkları günceller.

epochs 30 tercih edildi çünkü 20 ile yeterli başarı elde edilmedi; 70 ile de çok uyumlu bir başarı elde edildi. 30'nda yeterli olabileceği sonucuna varıldı ve model başarıyla eğitildi.

validation_data, modelin performansını izlemek için kullanılır. Her bir epoch sonunda, model doğrulama veri setindeki performansını ölçer ve eğitim süreci sırasında aşırı uyumu (overfitting) kontrol etmeye yardımcı olur.



5.MODELİN EĞİTİMİ

```
- val_loss: 0.6461 - val_accuracy: 0.6120 - val_precision: 0.6108 - val_recall: 1.0000
```

Resim 2: Modelin Eğitim Başarısı İlk Epoch

Resim 5.2. İlk Epoch Sonucu

```
val_loss: 0.0183 - val_accuracy: 0.9968 - val_precision: 1.0000 - val_recall: 0.9948
```

Resim 2: Modelin Eğitim Başarısı Son Epoch

Resim 5.3. Son Epoch Sonucu

Başlangıçta 0.65 olan loss değeri eğitim sonunda 0.02'ye düşürüldü; accuracy değeri 0.61'den eğitim sonunda 0.99'a yükseltildi.



6. FASTAPI

```
@app.post("/predict_image/")
async def predict_image(file: UploadFile = File(...)):
    load_cnn_model = load_model(r'C:\Users\Hp\Documents\GitHub\wtech-yapayzeka\fastApi\models\model_cnn.h5')
    with open("frame12.jpg", "wb") as f:
        f.write(await file.read())

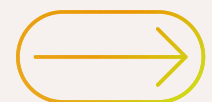
    test_image = Image.open("frame12.jpg")
    test_image = test_image.resize((240, 240))

    test_image = np.array(test_image)
    test_image = np.expand_dims(test_image, axis=0) / 255.0

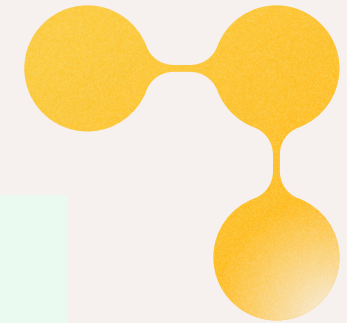
    prediction = load_cnn_model.predict(test_image)
    # predicted_class_index = np.argmax(prediction[0][0])

    if prediction[0][0] <= 0.5:
        return 'Jerry'
    else:
        return 'Tom'
```

Resim 6.1. FastApi Kodlama



6. FASTAPI



POST `/predict_image/` Predict Image

Parameters

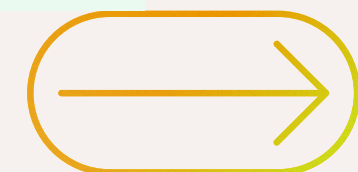
No parameters

Request body required

file * required
`string($binary)`

frame12.jpg

Resim 6.2. FastApi Resim Dosyası Seçme



6. FASTAPI

Code	Details
200	<div><div>Response body</div><div>"Jerry"</div><div><div>Download</div></div></div> <div><div>Response headers</div><div><div>content-length: 7</div><div>content-type: application/json</div><div>date: Thu,02 May 2024 16:27:56 GMT</div><div>server: uvicorn</div></div></div>
Responses	

Resim 6.3 FastApi Jerry’e ait Resim Sonucu

Code	Details
200	<div><div>Response body</div><div>"Tom"</div><div><div>Download</div></div></div> <div><div>Response headers</div><div><div>content-length: 5</div><div>content-type: application/json</div><div>date: Thu,02 May 2024 16:26:42 GMT</div><div>server: uvicorn</div></div></div>
Responses	

Resim 6.4. FastApi Tom’a ait Resim Sonucu



TEŞEKKÜR EDERİM

GÜLBAYAZ BAYRAM ÖZER

