

# Modelin Özellikleri ve Eğitim Süreci

Bu projede, bilgisayarın kedi ile köpeği ayırt etmesini sağlamak için iki farklı derin öğrenme modeli oluşturdum. Bu modeller, 128x128 boyutlarına küçültülmüş renkli resimler üzerinde çalışıyor. Amaç, bir resmin **kedi mi yoksa köpek mi** olduğunu otomatik olarak tahmin etmek.

## Modeller

### 1. Model – Temel Model

- İki tane “görüntü tarayıcı” katmanı var (Conv2D).
- Her taramadan sonra, resmi küçültüp önemli kısımları seçen katmanlar var (MaxPooling).
- Son olarak veriyi düzleştirip, sonucu tek bir sayıya çeviriyoruz: kedi mi köpek mi diye karar veriyor.
- Bu modelde dropout yok. Yani kendini çok ezberleyebilir, bu da gerçek hayatı işe yaramayabilir.

### 2. Model – Dropout’lu Model

- Birinci modelle çok benzer ama bu sefer araya dropout katmanları ekledik.
  - Dropout, bazı bağlantıları rastgele kapatıyor. Bu da modelin ezberlemesini zorlaştırmıyor.
  - Böylece daha genelleyici ve sağlam bir model elde etmeyi hedefliyoruz.
- 

## Eğitim

- Resimler ikiye ayrılıyor: %80’i modelin öğrenmesi için (%20’si de test etmek için).
  - Resimler 0 ile 1 arasında bir değere dönüştürülüyor ki model daha rahat çalışsin.
  - Modelin öğrenirken yaptığı hataları görmek için binary\_crossentropy adlı bir ölçüm kullanılıyor.
  - Öğrenme sırasında Adam adlı bir algoritma kullanılıyor. Öğrenme hızı 0.0005 olarak ayarlandı.
  - Takip edilen başarı ölçümleri:
    - accuracy: Tahminlerin ne kadar doğru olduğu
    - mse: Tahmin ile gerçek değer arasındaki farkın ortalaması
    - loss: Genel hata puanı
-

## Modellerin Karşılaştırılması

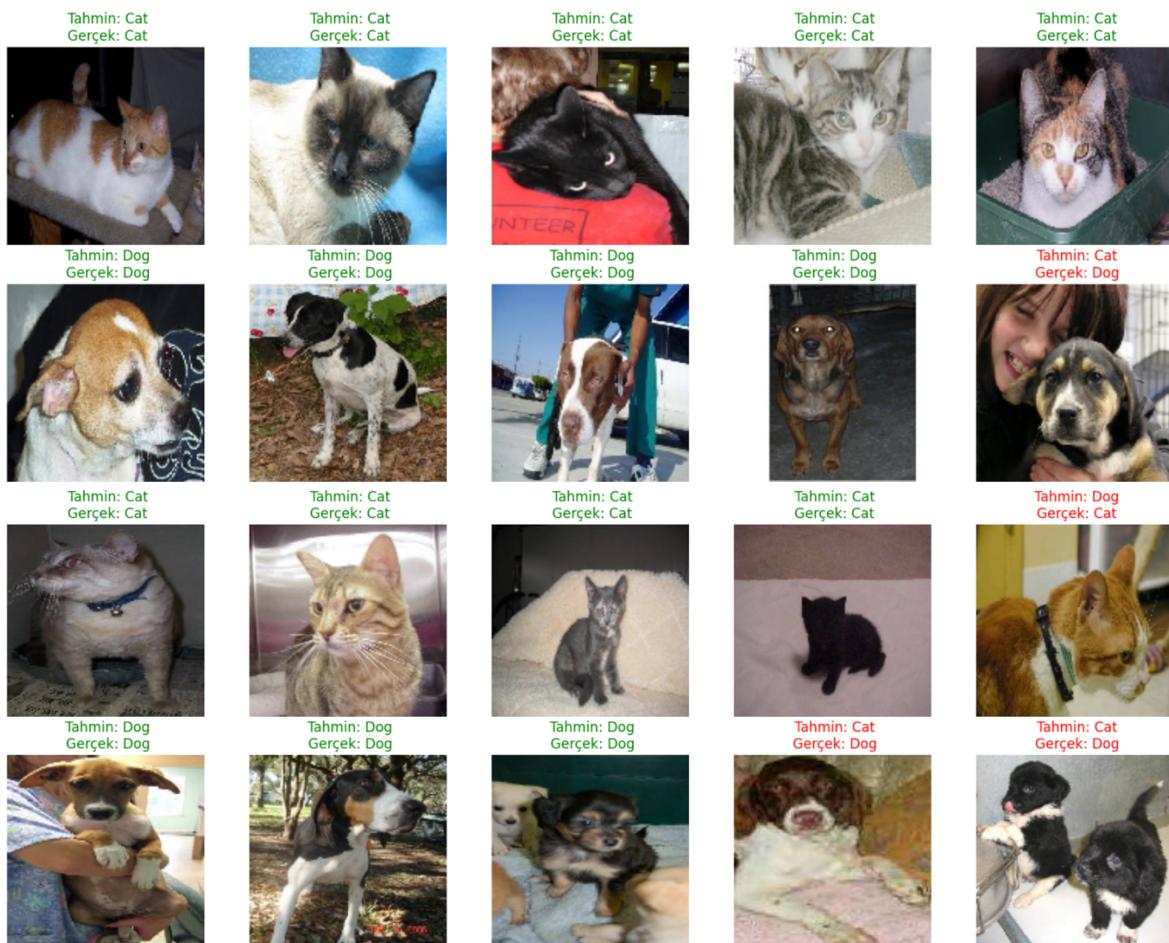
- İki model de aynı şartlarda çalıştırılıyor.
  - Hangisi doğrulama verisinde daha iyi sonuç verirse o model kaydediliyor (".h5" uzantılı dosya).
  - Her modelin eğitim süreci sırasında başarı, hata gibi veriler grafik olarak çiziliyor.
  - Ayrıca tüm sonuçlar bir .json dosyasına da yazılıyor: models/histories.json
- 

## Çıktılar (Modelin Ürettiği Sonuçlar)

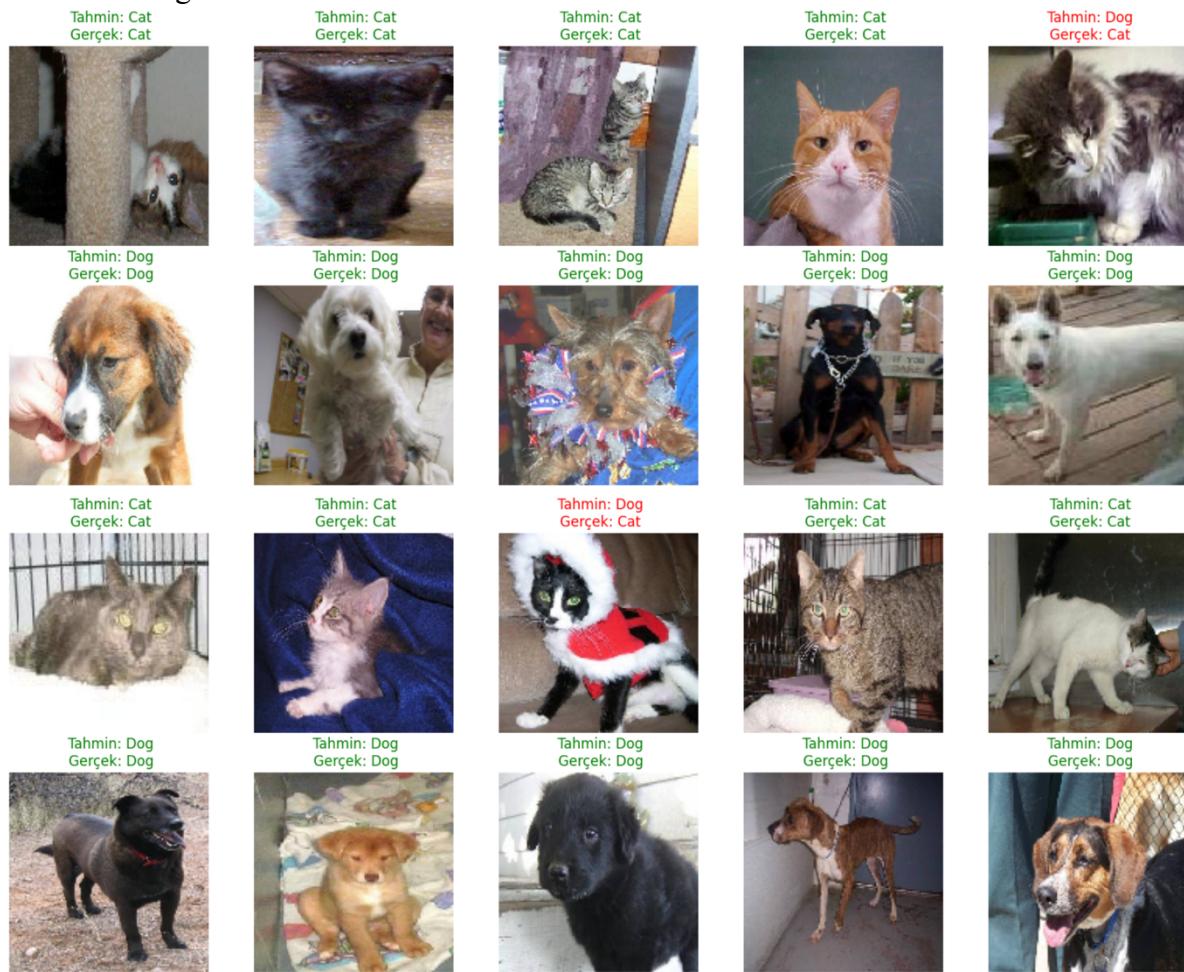
Eğitim tamamlandığında elimizde şunlar oluyor:

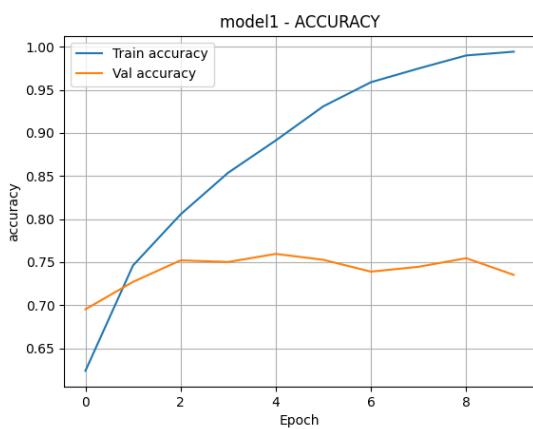
- En iyi modelin kayıt dosyası: model1.h5 veya model2.h5
- Eğitim sırasında başarı/hata grafikleri: örneğin model1\_accuracy.png
- Eğitim sürecinin detayları: histories.json

Model 1 Test görüntüleri:

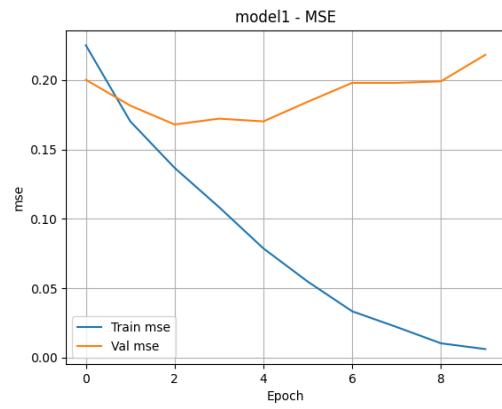


## Model 2 Test görüntüleri:

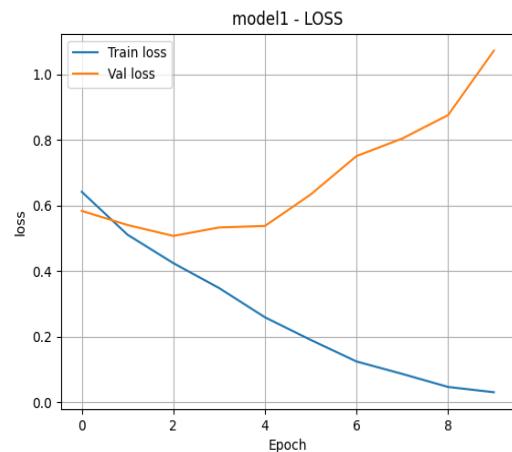




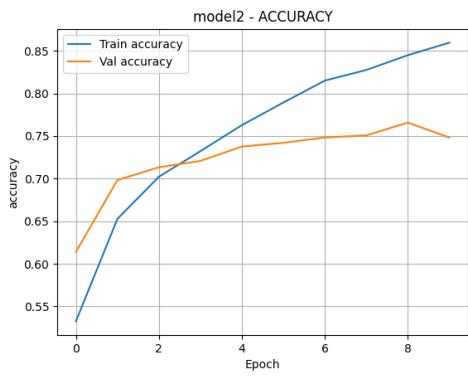
kesinliğimiz en başta 0.65 lerle yanlışlarla başlaması sonrasında doğruluk oranları artmaya başladı ve 0.95-0.99 arasında bir sayıya tekabül etti



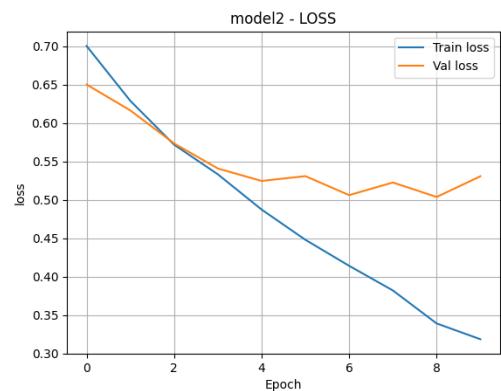
Hatalarımız gittikçe azalmaya başlamış  
0 a çok yakın olması doğru bir yolda  
olduğumuzu belirtir



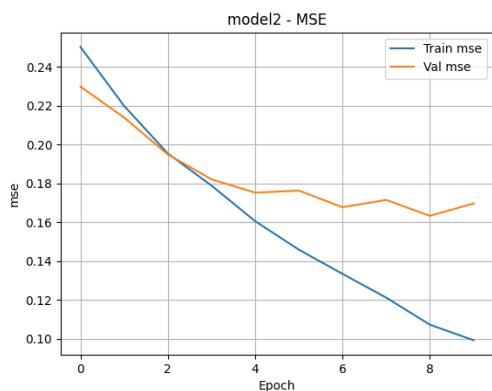
Gerçek ile tahmin arasındaki farkın 0 a yakın olması ve gittikçe azalması  
modelin doğru bir öğrenme de olduğunu  
belirtir



Model 1 ile kıyasla kötü bir sonuç olmuş  
0.85 doğruluk oranı düşük bir oran



kayıp fonksiyonumuzun 0.35 olması  
gerçekler ile tahminleri arasında farkın  
biraz falza olduğunu belirtir



Hatalarımız 0.10 luk bir kısım halen hataların  
olduğu düzenlemeler yapmamız gerektiğini belirtir

## SONUÇ:

Model1 Sonuçlarına göre daha doğru bir model ve daha sade bir yaklaşımla doğru sonuçlar almamıza neden olmuştur.