Görüntü işleme alanında makine öğrenmesi ve derin öğrenme yöntemlerinin kullanımı, özellikle sınıflandırma problemlerinde yaygınlaşmıştır. Bu çalışmada, kediler ile köpeklerin görüntülerinden yola çıkarak, hangi görüntünün kediye, hangisinin köpeğe ait olduğunu tahmin eden bir model geliştirilmiştir. Proje kapsamında Kaggle'dan temin edilen "Dogs vs. Cats" veri seti kullanılmıştır. Amacımız, derin öğrenme teknikleri ile bu görüntülerin doğru şekilde sınıflandırılmasını sağlamaktır.

Veri seti, Kaggle platformunda bulunan "Dogs vs. Cats" yarışmasından alınmıştır. Bu veri seti, kedi ve köpeklerin çeşitli açılardan çekilmiş resimlerini içerir. Veri setini indirirken gerekli izinler alınmış ve veri işleme sürecinde etik kurallar dikkate alınmıştır.

Veri seti, eğitim ve test olmak üzere iki ana kısımdan oluşmaktadır:

* Eğitim veri seti: 25.000 etiketlenmiş kedi ve köpek görüntüsü
* Test veri seti: Etiketi bulunmayan görüntüler.

Verileri işlemek için ilk olarak, Kaggle'dan indirilen .zip dosyaları açıldı ve ham görüntüler uygun dizinlere çıkarıldı. Bu işlem sonucunda, ../data/raw/train dizininde eğitilmiş veri setinin, ../data/raw/test1 dizininde ise test verilerinin kaydedilmesi sağlanmıştır.

Veri işleme sürecinde, görsellerin kategorilere ayrılması ve etiketlenmesi kritik bir adımdır. Bu projede, kedi ve köpek resimlerini sınıflandırmak için dosya isimleri üzerinden bir etiketleme işlemi gerçekleştirilmiştir.

Görüntüler, dosya isimlerinde hangi kategoriye ait olduklarını belirtir bir yapıya sahiptir. Dosya isimleri, cat.0.jpg, dog.0.jpg gibi yapıdadır. Bu yapı sayesinde, dosya isminin başındaki "cat" veya "dog" kelimeleri, görüntünün bir kedi mi yoksa köpek mi olduğunu belirtmektedir.

İlk olarak, os.listdir() fonksiyonu kullanılarak ../data/raw/train dizininde bulunan dosya isimleri listelenir. Bu dizin, ham eğitim verilerinin bulunduğu yerdir.

Her bir dosya ismi, "cat" ya da "dog" ifadesini içerir. Dosya isimleri, nokta (.) karakteri ile bölünerek ilk kısım alınır (örneğin, cat.0.jpg dosya isminde "cat" kısmı alınır). Bu işleme göre:

* + Eğer dosya ismi "dog" ile başlıyorsa, kategori olarak 1 atanır (köpek).
  + Eğer dosya ismi "cat" ile başlıyorsa, kategori olarak 0 atanır (kedi).

Tüm dosya isimleri ve kategoriler bir pandas veri çerçevesine (DataFrame) dönüştürülür. Bu yapı, ileride modelin eğitimi için kullanılacak tabloyu oluşturur. Bu tablo şu iki sütundan oluşur:

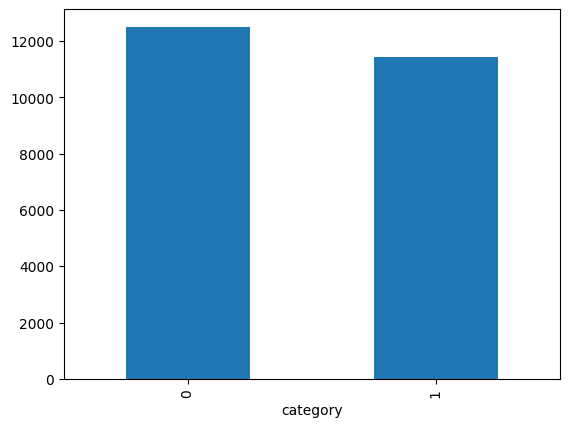
* + filename: Görüntü dosyalarının isimleri.
  + category: Görüntünün kedi (0) veya köpek (1) olup olmadığını belirten etiket.

Oluşturulan veri çerçevesi, modelin eğitimi sırasında kullanılacak etiketli veri setini tanımlar. Görseller ve onların ait oldukları sınıflar (kedi ya da köpek) bu çerçevede tutulur. Bu, modelin hangi resmin hangi kategoriye ait olduğunu öğrenmesi ve sınıflandırmayı doğru yapabilmesi için gerekli bir adımdır.

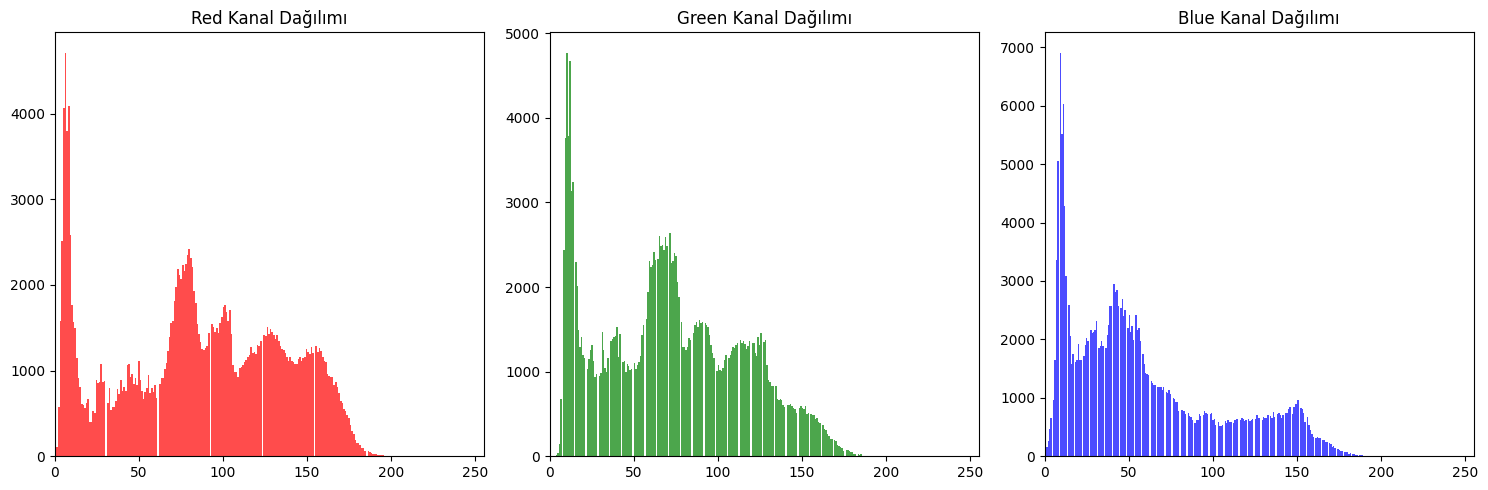
Veri seti üzerinde temel keşifsel analizlerin yapılması ve görüntülerin görselleştirilmesi, projeyi daha iyi anlamak açısından önemlidir. Bu aşamada, kategori dağılımlarının incelenmesi ve rastgele seçilen bir görüntünün görselleştirilmesi yapılmıştır. Veri setindeki kedi ve köpek görüntülerinin dağılımını incelemek için category sütununda yer alan verilerin frekansları analiz edilmiştir. Bu dağılımın bar grafiği ile görselleştirilmesi, sınıflar arasındaki dengeli veya dengesiz durumu incelemeye olanak tanır.

**value\_counts() Fonksiyonu:** Bu fonksiyon, veri çerçevesindeki category sütunundaki her bir sınıfın (0: kedi, 1: köpek) kaç kez tekrarlandığını sayar.

**Bar Grafiği:** plot.bar() komutu ile bu sayılar bir çubuk grafiği şeklinde görselleştirilir. Bu sayede, veri setindeki kedi ve köpek görüntülerinin dengeli olup olmadığı incelenebilir.



**Görüntü Yükleme ve Gösterme:** Seçilen görüntü dosyası load\_img() fonksiyonu ile yüklenir ve plt.imshow() fonksiyonu ile ekranda gösterilir. Bu işlem, görüntünün ham haliyle projede kullanıldığı şekliyle incelenmesine olanak sağlar.



Görüntülerin çözünürlüğü (boyutları) ve renk kanallarının dağılımları analiz edilmiştir. Görüntülerin RGB renk uzayındaki her bir kanalının (kırmızı, yeşil, mavi) histogramları çıkarılarak, her kanaldaki piksel yoğunluklarının dağılımı görselleştirilmiştir. Bu adım, modelin eğitim verilerinin niteliklerini anlamak için önemli bir adımdır.

Proje kapsamında, veri setindeki görüntüler rastgele seçilmiş ve seçilen görüntünün boyutları ile renk kanalı bilgileri (RGB) çıkarılmıştır.

**Görüntü Boyutu (Resolution):** image.size ifadesi, seçilen görüntünün genişlik ve yükseklik değerlerini verir. Bu, görüntünün çözünürlüğü hakkında bilgi sağlar.

**Renk Kanalları:** image\_array.shape[2] ifadesi, görüntüde kaç adet renk kanalı olduğunu gösterir. Renkli görüntüler için bu değer 3’tür (Kırmızı, Yeşil, Mavi kanalları).

Seçilen görüntünün her bir renk kanalındaki (RGB) piksel dağılımları analiz edilmiştir. Bu dağılımlar, her bir kanalda bulunan piksel yoğunluklarının 0 (siyah) ile 255 (beyaz) arasında nasıl dağıldığını göstermektedir.

**Histogram Analizi:** Renk kanallarının (kırmızı, yeşil ve mavi) her biri için, piksel değerlerinin histogramları oluşturulmuştur. ravel() fonksiyonu, çok boyutlu diziyi tek boyutlu hale getirir ve hist() fonksiyonu, bu değerlerin dağılımını gösterir.

**Dağılım Aralığı:** Piksel değerleri 0 ile 255 arasında olup, 0 siyahı, 255 ise tam beyazı ifade eder. Her bir renk kanalının dağılımı, o kanalın görüntüdeki baskınlığını ve yoğunluğunu anlamamıza yardımcı olur.

Kedi ve köpek sınıflandırma problemi için bir Convolutional Neural Network (CNN) modeli tanımlanmıştır. Modelin eğitimi için gerekli veri hazırlama işlemleri gerçekleştirilmiş ve eğitim sırasında kullanılacak optimizasyon ve erken durdurma stratejileri belirlenmiştir.

Görüntü sınıflandırma problemlerinde yaygın olarak kullanılan Convolutional Neural Networks (CNN) mimarisi bu projede tercih edilmiştir. CNN, görüntü verilerindeki mekansal hiyerarşileri öğrenerek başarılı sınıflandırmalar yapabilen bir yapay sinir ağı mimarisidir.

**Katmanlar:** CNN modeli üç adet evrişim (convolutional) katmanı, her birinin ardından gelen max-pooling katmanları ve dropout katmanlarından oluşur. Bu katmanlar, görüntüdeki mekansal hiyerarşiyi öğrenir ve özellikleri çıkartır.

**Dropout:** Modeldeki Dropout katmanları aşırı öğrenmeyi (overfitting) engellemek için kullanılır. Dropout, eğitim sırasında bazı nöronların rastgele devre dışı bırakılmasını sağlar.

**Aktivasyon Fonksiyonları:** Evrişim katmanlarında ReLU (Rectified Linear Unit) aktivasyon fonksiyonu kullanılırken, çıkış katmanında iki sınıfı (kedi ve köpek) tahmin etmek için softmax aktivasyonu kullanılmıştır.

**Modelin Derlenmesi:** Model categorical\_crossentropy kaybı ile derlenmiş ve optimizer olarak rmsprop kullanılmıştır. Modelin performansını değerlendirmek için accuracy metriği seçilmiştir.

**EarlyStopping:** Eğitim sırasında, belirli bir süre boyunca modelin doğruluğu iyileşmediğinde eğitim durdurulur. Bu, modelin gereksiz yere eğitim görmesini ve aşırı öğrenmeyi önler.

**ReduceLROnPlateau:** Eğer validation doğruluğu iyileşmezse, öğrenme hızı yarıya düşürülür. Bu, modelin daha düşük bir öğrenme hızı ile daha iyi sonuçlar elde etmesine yardımcı olabilir.

**Veri Arttırma:** Görüntülere çeşitli dönüşümler (dönme, yatay çevirme, yakınlaştırma, kaydırma vb.) uygulanarak veri seti çeşitlendirilmiştir. Bu, modelin farklı görüntüleme koşullarına karşı daha sağlam olmasını sağlar.

**Rescale:** Görüntülerdeki pikseller 0 ile 255 arasında değerler içerir. rescale=1./255 ile bu değerler normalize edilerek 0 ile 1 arasına çekilmiştir.

Modelin eğitim süreci tamamlanmış ve sonuçlar analiz edilmiştir. Eğitim sırasında modelin başarımı, doğrulama verisi üzerinde gözlemlenmiş ve modelin sınıflandırma yetenekleri sınıflandırma raporu ve karışıklık matrisi ile değerlendirilmiştir.

Model, aşağıdaki parametreler ile eğitilmiştir:

**Epochs:** Model belirli bir sayıdaki epoch boyunca eğitim almıştır. Epoch sayısı modelin ne kadar süreyle veriler üzerinden öğrenim gerçekleştirdiğini gösterir.

**Validation Data:** Eğitim sırasında her epoch sonunda model doğrulama verisi üzerinden de test edilmiş ve doğrulama kaybı (validation loss) ile doğrulama doğruluğu (validation accuracy) ölçülmüştür.

**Callbacks:** Erken durdurma ve öğrenme hızının azaltılması için tanımlanan callback fonksiyonları eğitim sürecinde etkin bir şekilde kullanılmıştır.

Eğitim süreci sonunda, modelin eğitim ve doğrulama aşamalarındaki kayıp ve doğruluk değerleri çizilerek görselleştirilmiştir. Bu grafikler, modelin öğrenme sürecini ve overfitting olup olmadığını anlamaya yardımcı olur.

**Eğitim ve Doğrulama Kayıp Grafiği:** Eğitim ve doğrulama kayıpları, modelin overfitting yapıp yapmadığını anlamak için incelenmiştir. Doğrulama kaybı eğitim kaybına göre yükselmeye başlarsa modelin aşırı öğrenme yapmaya başladığı anlaşılabilir.

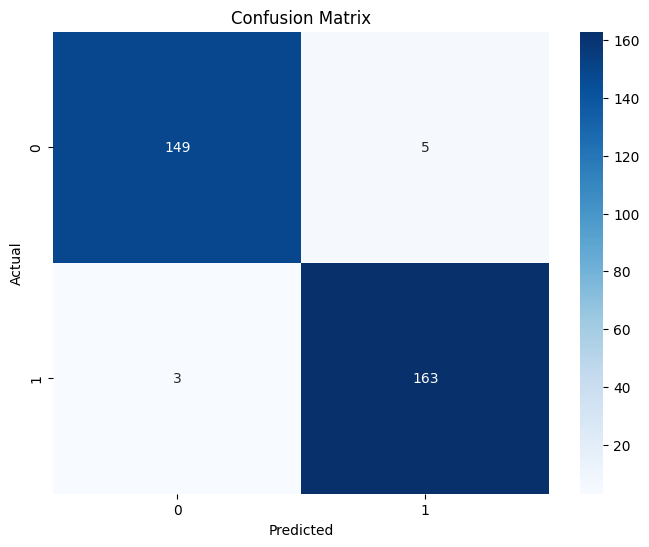
**Eğitim ve Doğrulama Doğruluk Grafiği:** Modelin eğitim ve doğrulama doğrulukları da eğitim boyunca izlenmiştir. Modelin doğrulama doğruluğunun artışı, modelin genel olarak başarılı bir şekilde öğrenim gerçekleştirdiğini gösterir.

Model eğitildikten sonra, doğrulama seti üzerindeki performansı aşağıdaki metrikler ile değerlendirilmiştir:

**Sınıflandırma Raporu (Classification Report):** Sınıflandırma raporu, doğruluk (precision), hatırlama (recall) ve F1 skoru gibi metrikleri içerir.

**Sınıflandırma Raporu:** Bu rapor, modelin her bir sınıf (kedi ve köpek) için doğruluk, hatırlama ve F1 skorlarını sunar. Bu metrikler, modelin sınıflandırma başarımını farklı açılardan değerlendirir.

**Karışıklık Matrisi:** Modelin tahmin performansını daha görsel bir biçimde anlamak için karışıklık matrisi kullanılmıştır. Bu matriste, satırlar gerçek sınıfları, sütunlar ise tahmin edilen sınıfları temsil eder. Diyagonal üzerindeki değerler doğru tahminleri, diyagonal dışındaki değerler ise yanlış tahminleri gösterir.



Modelin performansını anlamak ve iyileştirmek adına, yanlış sınıflandırılan görüntülerin analizi önemli bir adımdır. Yanlış sınıflandırmalar modelin sınıflandırma sınırlarını, zayıflıklarını ve olası iyileştirme alanlarını gösterir.

Modelin doğrulama setinde yaptığı hataları analiz etmek için, yanlış sınıflandırılan görüntüler tespit edilmiştir. Bu amaçla, tahmin edilen etiketler ile gerçek etiketler karşılaştırılarak hatalı sınıflandırılan örnekler seçilmiştir.

Yanlış sınıflandırılan görüntüler, modelin sınıflandırmada zorluk yaşadığı durumları daha iyi anlamak için görselleştirilmiştir. Görüntüler, yanlış sınıflandırmanın nedenlerini anlamak açısından önemli ipuçları sunacaktır.