

NESNE TANIMA

Muhammet Enes İnal

Bilgisayar Mühendisliği

İstanbul Topkapı Üniversitesi

22040101023

muhammetenesinal@stu.topkapi.edu.tr

EEE- Özcan Çomak

Elektrik-Elektronik Mühendisliği

İstanbul Topkapı Üniversitesi

22040201026

ozcancomak@stu.topkapi.edu.tr

u.tr

Muhammed Mert Oruç

Bilgisayar Mühendisliği

İstanbul Topkapı Üniversitesi

22040101035

muhammedmertoruc@stu.topkapi.edu.tr

kapi.edu.tr

Zekeriya Deniz Uğurlu

Bilgisayar Mühendisliği

İstanbul Topkapı Üniversitesi

22040101034

zekeriyadenizugurlu@stu.topkapi.edu.tr

I. ÖZET

Bu projede, videolardan veya anlık görüntülerden nesneleri tespit etmek amacıyla derin öğrenme teknikleri kullanılmaktadır. Nesne tanıma projesi, kameralardan alınan görüntüler üzerinde nesneleri tanımak ve sınıflandırmak amacıyla geliştirilmiştir. Bu tespitler, güvenlik sistemlerinden otonom araçlara kadar birçok alanda kullanılabilecek şekilde tasarlanmıştır. Projede, ResNet gibi gerçek zamanlı nesne tanıma algoritmalarından yararlanılmıştır. Görüntülerdeki nesneler yüksek doğruluk oranıyla tespit edilmekte ve kullanıcıya görsel ya da sesli bildirimlerle sunulmaktadır. Ayrıca, sistemin farklı ortamlarda ve ışık koşullarında da başarılı sonuçlar vermesi hedeflenmiştir. Gerçek ortamda yapılan testlerde, sistem bir aracın üzerine yerleştirilen kamera ile çalıştırılmış ve hareket hâlindeyken de nesneleri başarıyla tespit edebilmiştir. Bu uygulama, sistemin dinamik ve değişken çevre koşullarında da etkili sonuçlar verebildiğini göstermektedir.

II. GİRİŞ

Son yıllarda derin öğrenme tekniklerinin gelişmesiyle birlikte, bilgisayarla görme alanında önemli ilerlemeler kaydedilmiştir. Bu gelişmeler sayesinde, nesne tanıma sistemleri farklı sektörlerde yaygın şekilde kullanılmaya başlanmıştır. Bu projede, kamera görüntülerinden nesnelerin tespiti ve sınıflandırılması hedeflenmiştir. ResNet mimarisi kullanılarak geliştirilen model, görüntülerdeki nesneleri yüksek doğrulukla tanımlayabilmektedir. Sistem, bir aracın üzerine yerleştirilen kamera ile gerçek zamanlı olarak test edilmiş ve hareketli ortamlarda da başarılı sonuçlar vermiştir. Bu çalışma, otonom araçlar ve akıllı gözetim sistemleri gibi uygulamalara katkı sağlamayı amaçlamaktadır.

III. YÖNTEMLER

A) Veri Toplama ve Hazırlık

Bu projede kullanılan veri kümesi, beş farklı nesne kategorisini içermektedir:

Araba , Bardak,Gözlük,Top,Parfüm .Görseller, her bir sınıf için ayrı klasörlerde olacak şekilde düzenlenmiştir. Veriler "`/home/eliff/Downloads/proje/dataset 2`" dizini altında yer almakta ve iki alt klasöre ayrılmıştır:

Training: Modelin eğitimi için kullanılır.

Testing: Modelin doğrulanması ve test edilmesi için kullanılır.

Veriler Raspberry Pi cihazıyla birlikte çalışan PiCamera modülü kullanılarak toplanmıştır Görseller, daha sonra eğitim ve test klasörlerine yerleştirilerek modelin eğitimi için uygun hale getirilmiştir.

B) Veri Ön İşleme

Veri ön işleme aşamasında, Keras'ın ImageDataGenerator sınıfı kullanılarak veri artırma (data augmentation) işlemleri uygulanmıştır. Bu işlemler sayesinde modelin genelleme yeteneği artırılmış ve aşırı öğrenme (overfitting) riski azaltılmıştır.

Uygulanan işlemler:

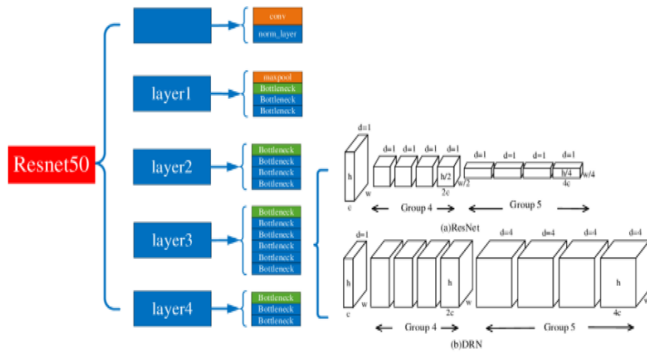
- Döndürme (rotation_range=30)
- Genişlik ve yükseklik kaydırma (width/height_shift_range=0.2)
- Kesme (shear_range=0.2)

- Yakınlaştırma (zoom_range=0.2)
- Yatay çevirme (horizontal_flip=True)
- ResNet50 modeline uygun ön işleme fonksiyonu (preprocess_input) uygulanmıştır.

Görseller tüm sınıflar için 224x224 piksel boyutunda yeniden boyutlandırılmıştır.

C) Modelin Tanımlanması

Model yapısı olarak evrişimli sinir ağı (Convolutional Neural Network - CNN) mimarisine dayalı, önceden eğitilmiş bir derin öğrenme modeli olan ResNet50 kullanılmıştır. Transfer öğrenme yöntemiyle, ImageNet veri kümesi üzerinde eğitilmiş ResNet50 modeli temel alınmış ve kendi problemimize özel katmanlar eklenerek yeniden yapılandırılmıştır.



Fotograf 1

Modelin include_top=False parametresiyle en üstteki sınıflandırma katmanları kaldırılmış ve yerine sırasıyla GlobalAveragePooling2D(), 512 nöronlu Dense katmanı (ReLU aktivasyon fonksiyonlu), Dropout(0.5) katmanı (aşırı öğrenmeyi engellemek için), ve son olarak 5 sınıflı Dense(5, activation='softmax') katmanı eklenmiştir. Modelin giriş boyutu input_shape=(224, 224, 3) olarak belirlenmiştir. Böylece yalnızca son katmanlar eğitime açık bırakılmış, ResNet50'nin önceden öğrenilmiş özellik çıkarıcı katmanları sabitlenmiştir. Bu yaklaşım sayesinde hem eğitim süresi kısaltılmış hem de küçük boyutlu veri setiyle yüksek başarı elde edilmesi sağlanmıştır.

D) Modelin Eğitilmesi

Modelin eğitimi, belirlenen eğitim veri kümesi üzerinde gerçekleştirilmiştir. Eğitim sürecinde kayıp fonksiyonu olarak kategorik çapraz entropi (categorical_crossentropy), optimizasyon algoritması olarak ise Adam optimizasyon yöntemi tercih edilmiştir. Eğitim işlemi, mini-batch yöntemiyle gerçekleştirilmiş ve batch boyutu 16 olarak

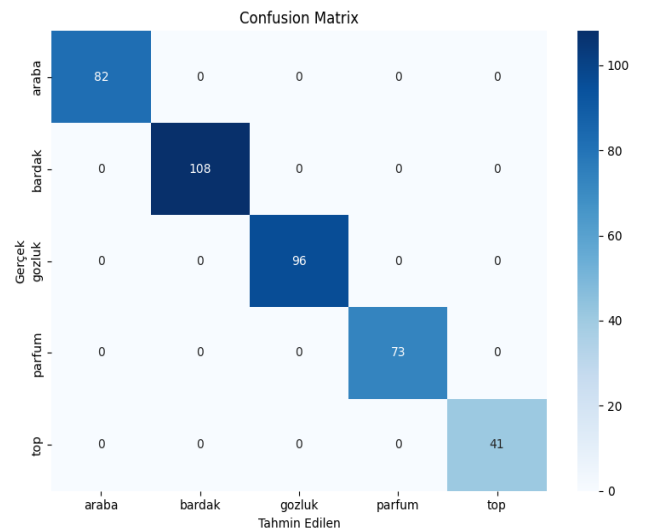
belirlenmiştir. Eğitim sırasında her bir epoch sonunda modelin doğrulama veri kümesindeki performansı da değerlendirilmiş ve bu sayede modelin aşırı öğrenme gösterip göstermediği kontrol edilmiştir. Eğitim sürecinde EarlyStopping gibi tekniklerin kullanılması, eğitim tamamlanmasa bile modelin doğrulama başarısı düşmeye başladığında işlemi durdurmayı mümkün kılmıştır (bu özellik doğrudan kodda geçmiyor olabilir, eklenebilir). Eğitim süresi boyunca doğruluk ve kayıp değerleri izlenmiş ve eğitim tamamlandıktan sonra bu metrikler grafiksel olarak görselleştirilmiştir.

E) Modelin Kaydedilmesi

Model eğitildikten sonra, daha sonra tekrar kullanılmak üzere kalıcı olarak diske kaydedilmiştir. Keras'ın model.save() fonksiyonu kullanılarak model.h5 formatında kaydedilmiştir. Bu işlem sayesinde modelin yeniden eğitilmesine gerek kalmadan farklı zamanlarda, farklı uygulamalarda ya da donanımlarda kolayca yüklenmesi mümkün hale gelmiştir. Özellikle gerçek zamanlı uygulamalarda (örneğin Raspberry Pi üzerinde canlı kamera verisiyle çalışırken) bu kaydedilen model doğrudan yüklenerek kullanılmaktadır.

F) Modelin Değerlendirilmesi

Model eğitimi tamamlandıktan sonra, test veri seti kullanılarak modelin doğruluğu ve genelleme kabiliyeti değerlendirilmiştir. Modelin çıktıları ile gerçek etiketler karşılaştırılarak doğruluk oranı hesaplanmış ve sınıflandırma başarısı analiz edilmiştir. Her sınıfa ait başarı oranları ve karışıklık matrisi (confusion matrix) gibi istatistiksel ölçütler değerlendirme kapsamında kullanılabilir. Ayrıca modelin hangi sınıflarda daha başarılı olduğu, hangi sınıflarda hata yaptığı gözlemlenmiştir. Bu aşama, modelin eksiklerini tespit etmek ve gerekirse yeni veri ile yeniden eğitmek için önemli bir referans olmuştur.



Fotograf 2

G) Modelin kullanılması

Modelin Yüklmesi ve Hazır Hâle Getirilmesi

Eğitim süreci tamamlanan derin öğrenme modeli, .h5 uzantılı bir dosya olarak kaydedilmiştir. Bu dosya, TensorFlow ve Keras kütüphaneleri aracılığıyla sistem çalıştırıldığında belleğe yüklenmektedir. Yüklenen model, canlı görüntü verisi üzerinde tahmin yapmaya hazır hale getirilmiştir. Modelin yüklmesi sırasında, ağırlıklar ve mimari yapılar doğrudan geri çağrılarak çalışma esnasında herhangi bir yeniden eğitim gerektirmeden tahmin işlemleri gerçekleştirilmiştir.

Görüntü Verisinin Alınması (Kamera ile)

Modelin kullanılabilmesi için canlı görüntü gereklidir. Bu amaçla Raspberry Pi cihazına entegre edilen PiCamera modülü kullanılmaktadır. Kamera ile sürekli olarak görüntü alınır ve bu görüntüler gerçek zamanlı olarak işlenir. Alınan görüntüler, modelin beklentilerine uygun şekilde 224x224 piksel boyutuna yeniden boyutlandırılır. Bu adım, modelin doğru ve anlamlı çıktılar üretebilmesi için son derece kritiktir.

Görüntünün Modelle Uyumlu Hâle Getirilmesi (Önişleme)

Kameradan alınan görüntüler, modelin giriş formatına uygun olacak şekilde yeniden boyutlandırılır, numpy dizisine çevrilir ve pikseller normalize edilir. Genellikle 0-255 arası RGB değerleri 0-1 arasına çekilerek modelin daha dengeli ve hızlı öğrenme yapması sağlanır. Ardından bu dizi dört boyutlu hale getirilerek (örneğin (1, 224, 224, 3)), doğrudan modelin predict() fonksiyonuna verilebilecek formata dönüştürülür.

Sınıflandırma Sonucunun Elde Edilmesi

Model tahmin işlemini gerçekleştirdiğinde, her sınıfa ait olasılık değerlerini içeren bir çıktı üretir. Bu çıktılarından en yüksek değere sahip olan sınıf, görüntüdeki nesnenin hangi kategoriye ait olduğunu gösterir. Örneğin çıktı [0.05, 0.02, 0.91, 0.01, 0.01] ise bu, nesnenin %91 ihtimalle “gözlük” olduğunu ifade

eder. Bu çıktı doğrudan kullanıcıya sunulmak üzere ekrana yazdırılır veya sesli olarak okunur.

Sesli Bildirim Mekanizması (Text-to Speech)

Kullanıcının sınıflandırma sonucunu hızlı ve anlaşılır bir şekilde alabilmesi için sesli bildirim sistemi entegre edilmiştir. pyttsx3 kütüphanesi kullanılarak sınıflandırma sonucu Türkçe olarak sesli biçimde ifade edilir. Bu sistem özellikle görme engelli bireyler için erişilebilirliği artırmakta ve kullanıcıların etkileşim kurmadan bilgi edinmesini sağlamaktadır. Sistem, sınıfın Türkçeleştirilmiş ismini otomatik olarak okuyarak kullanıcıya anında sesli geri bildirim verir.

Uzaktan Takip için Web Sunucusu (Flask ile)

Modelin sadece yerel cihazda değil, aynı zamanda ağ üzerinden de takip edilebilmesi amacıyla bir Flask tabanlı web servisi geliştirilmiştir. Raspberry Pi cihazı, bu sunucu aracılığıyla sınıflandırma sonucunu ağdaki başka bir cihazın (örneğin bir telefon veya dizüstü bilgisayarın) görebileceği şekilde yayınlamaktadır. Böylece sistemin kontrolü ve takibi, yalnızca fiziksel olarak cihazın başında olmadan da gerçekleştirilebilir.

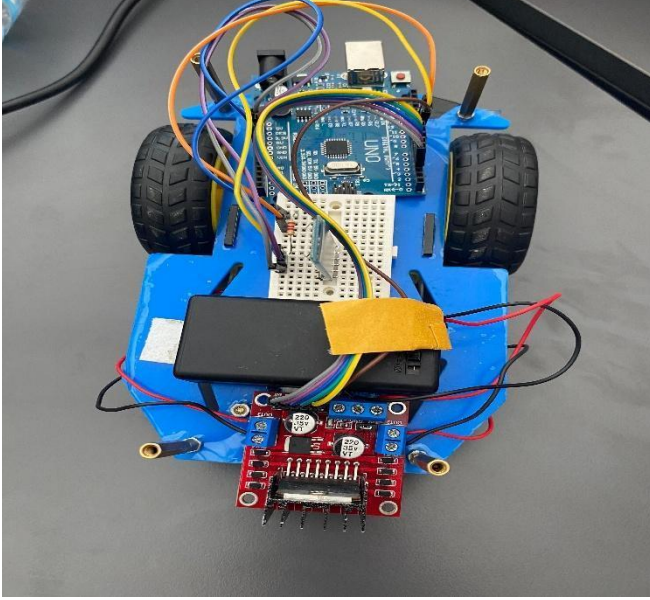
Gerçek Zamanlı Kullanılabilirlik ve Otomasyon Potansiyeli

Tüm bu adımların birleşimiyle sistem, gerçek zamanlı çalışan, sesli geri bildirim sunan ve uzaktan izlenebilen bir yapıya kavuşmuştur. Bu durum, sistemin yalnızca bir akademik çalışma olarak kalmadığını; aynı zamanda gerçek dünyada, özellikle akıllı ev sistemleri, engelli bireyler için yardımcı teknolojiler ve eğitim uygulamaları gibi alanlarda kullanılabileceğini göstermektedir.

H) Arduino

Arduino Uno mikrodenetleyicisi kullanılarak Bluetooth kontrollü basit bir araç tasarlanmıştır. Araç, iki adet DC motorla hareket ederken, yön dengesini sağlamak için bir adet sarhoş (kastor) tekerlek kullanılmıştır. HC-05 Bluetooth modülü sayesinde araç, mobil cihazlardan gelen komutlarla uzaktan kontrol edilebilmektedir. Motorlara enerji sağlamak amacıyla 9V pil kullanılmıştır. Arduino'ya bağlı motor sürücü (L298N

ya da L293D) üzerinden motorlara ileri, geri, sağ ve sol yönde hareket komutları gönderilmiştir. Mobil uygulama ile gönderilen komutlar, HC-05 üzerinden Arduino'ya iletilmiş ve işlenmiştir. Sistem, düşük maliyetli ve taşınabilir bir robotik çözüm sunmaktadır. Bu proje ile temel seviye mekatronik uygulamalarına giriş yapılmış ve Bluetooth kontrollü sistemler hakkında pratik deneyim kazanılmıştır.



Fotograf 3

IV. SONUÇ

Bu proje kapsamında, Python dili ve TensorFlow kütüphanesi kullanılarak, farklı nesne türlerini yüksek doğrulukla ayırt edebilen bir görüntü sınıflandırma sistemi başarıyla geliştirilmiştir. Eğitim verileri dikkatli şekilde toplanmış, ön işleme adımları uygulanmış ve derin öğrenme modeli gerçek zamanlı tanıma yapabilecek düzeyde optimize edilmiştir. Model, test verileri üzerinde yüksek başarı oranları sergileyerek hedeflenen amaca ulaşmıştır.

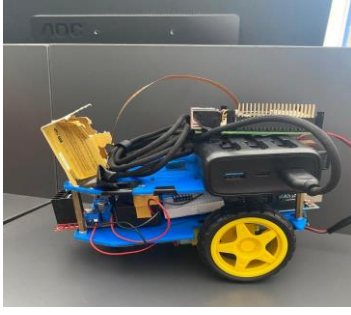
ResNet50 tabanlı önceden eğitilmiş bir model kullanılarak transfer öğrenme yöntemiyle sınırlı veriyle etkili bir öğrenme süreci gerçekleştirilmiştir. Bu sayede, sıfırdan bir model eğitmek için gereken büyük veri setlerine ve uzun eğitim sürelerine ihtiyaç duyulmadan yüksek performanslı bir sınıflandırıcı elde edilmiştir.

Eğitim tamamlandıktan sonra model Raspberry Pi platformuna entegre edilmiş, canlı görüntü akışı üzerinden nesne tanıma yapılabilmektedir. Modelin bu platformda sorunsuz çalışması, sistemin taşınabilirliği, maliyet etkinliği ve enerji verimliliği açısından önemli bir başarı olarak değerlendirilmiştir.

Geliştirilen sistem, tanıdığı nesneleri sadece ekranda yazı ile göstermekle kalmamış, aynı zamanda sınıflandırma sonuçlarını sesli olarak kullanıcıya iletmıştır. Bu özellik, özellikle görme engelli bireylerin sistemden doğrudan fayda sağlayabilmesini sağlamış ve sistemin erişilebilirliğini artırmıştır.

Flask kullanılarak oluşturulan basit web sunucusu sayesinde, Raspberry Pi tarafından sınıflandırılan nesne bilgileri uzaktaki cihazlardan takip edilebilir hale getirilmiştir. Bu durum, sistemin sadece yerel cihazda değil, aynı zamanda internet ya da lokal ağ üzerinden farklı cihazlarla uyumlu çalışabileceğini göstermiştir.

Modelin genel başarımı, gerçek zamanlı çalışabilirliği, sesli bildirim sistemi ve taşınabilirlik gibi faktörler bir araya getirildiğinde, sistemin yalnızca teorik bir uygulama olmadığı, pratikte de kullanılabilir bir çözüm sunduğu sonucuna varılmıştır. Bu yönüyle proje, sosyal fayda sağlayabilecek, erişilebilir teknolojilere örnek oluşturabilecek bir yapıya sahiptir.



Fotograf 4

V.KAYNAKLAR

- [1] Chollet, F. (2015). *Keras: The Python deep learning library*. <https://keras.io>
- [2] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778. <https://doi.org/10.1109/CVPR.2016.90>
- [3] Abadi, M., et al. (2016). TensorFlow: Large-scale machine learning on heterogeneous systems. *Software available from tensorflow.org*. <https://www.tensorflow.org>
- [4] Simonyan, K., & Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. *International Conference on Learning Representations (ICLR)*. <https://arxiv.org/abs/1409.1556>
- [5] Brownlee, J. (2019). *Deep Learning for Computer Vision: Image Classification, Object Detection, and Face Recognition in Python*. Machine Learning Mastery.
- [6] Raspberry Pi Foundation. (2022). *Raspberry Pi Documentation*. <https://www.raspberrypi.com/documentation>
- [7] Zhang, Y., & Wallace, B. (2015). A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. *Proceedings of the 8th International Conference on Learning Representations (ICLR)*. <https://arxiv.org/abs/1510.03820>
- [8] "CNN (Convolutional Neural Networks) Nedir?" *Medium*, [Online]. Available: <https://bartubozkurt35.medium.com/cnn-convolutional-neural-networks-nedir-a5bafc4a82a1>. [Accessed: Jan. 16, 2025]. “

Video Link: <https://youtu.be/d3Ls1BCTmAE>

PROJEDE EMEĞİ GEÇENLER



Muhammet Enes İnal, Flutter, Dart, C, C#, Java, Javascript, Python, PHP dilleri üstünde çalışmıştır. Önemli projeleri arasında C# ile hastane uygulaması, Html-CSS-PHP ile otel web sitesi yer alır.



Zekeriya Deniz Uğurlu; Flutter, Dart, C, C++, C# Html-CSS-PHP , Java, Python dilleri üzerinde yoğunlaşmış ve kendini geliştirmiştir. Önemli projeleri arasında C# ile sürücü kursu kayıt uygulaması, Flutter ile Spor takvim oluşturma mobil uygulaması ve HTML-CSS ile sürücü kursu web sitesi projeleri yer alır.



Muhammed Mert Oruç, C, Flutter, Dart, C++, C#,Java, HTML, CSS, PHP ve Python dillerinde kendini geliştirmiştir.C# ile otopark uygulaması, Html-CSS-PHP ile bankacılık web sitesi yer alır.

EEE-



**Özcan
Çomak,**
Proteus,
SolidWorks ve
AutoCAD gibi
mühendislik ve
tasarımda
çalışmalar

yapmıştır. Projeleri arasında, “piezoelektrik
malzeme ile yenilenebilir enerji üretimi” üzerine
çalışmıştır.