

Object Recognition and Guessing Game Using YOLO Model

YOLO Modeli Kullanılarak Nesne Tanıma ve Etkileşimli Tahmin Oyunu

Tunahan Çolak
Information Systems Engineering
Kocaeli University
Kocaeli, Türkiye
211307053@kocaeli.edu.tr

Yunus Emre Akyay
Information Systems Engineering
Kocaeli University
Kocaeli, Türkiye
211307046@kocaeli.edu.tr

Özet— Bu proje, YOLOv8 modeli kullanılarak geliştirilen bir nesne algılama sistemini kullanarak, bir tahmin oyunu sunmaktadır. Projede bileşenler: nesne algılama, bulanıklaştırılmış görüntüler ve tahmin mekanizmasıdır. Uygulama, oyun şeklinde kullanıcıya ipuçları sunarak tanınan nesnelerin doğru bir şekilde tahmin edilmesidir.

Anahtar Kelimeler— YOLOv8, Nesne tanıma, Tkinter

I. GİRİŞ

Bu proje, nesne tanıma teknolojisinin eğlenceli bir uygulama alanını keşfetmeyi amaçlamaktadır. Proje kapsamında, kamera ile gerçek zamanlı olarak algılanan nesneler, kullanıcıya bir oyun arayüzü üzerinden sunulmuş ve kullanıcıdan nesneleri ipuçlarıyla eşleştirmesi istenmiştir. Böylelikle, nesne tanıma teknolojisi hem eğlenceli bir deneyim sunmuş hem de kullanıcıların bu teknolojiyi öğrenmesi ve anlaması sağlanmıştır.

Literatüre bakıldığında, nesne algılama teknolojisinin özellikle eğitim, güvenlik ve sağlık alanlarında yaygın olarak kullanıldığı görülmektedir. Örneğin, eğitimde görsel içerik analizi, güvenlikte yüz tanıma ve izinsiz giriş tespiti, sağlıkta ise medikal görüntü analizi gibi uygulamalar bu teknolojinin önemini ortaya koymaktadır. Bununla birlikte, nesne tanıma teknolojisinin eğlence ve oyun sektöründeki potansiyeli henüz tam anlamıyla araştırılmamıştır.

Bu projenin katkısı, nesne tanıma teknolojisinin eğlence ve eğitim alanındaki kullanımına yönelik yeni bir yön sunması

olabilir. Geliştirilen oyun, kullanıcıların nesne algılama teknolojisini deneyimlemesine olanak tanıırken aynı zamanda bu alanda yapılan çalışmalara farklı bir bakış açısı kazandırmaktadır. Çalışma, gerçek zamanlı nesne algılama ve uygulama geliştirme konularında önemli bir örnek olabilir. Ayrıca, kullanılan yöntemlerin ve geliştirilen yazılımın detaylı bir şekilde açıklanması, bu teknolojiyi kendi projelerinde kullanmak isteyen araştırmacılar ve geliştiriciler için bir rehber niteliği taşımaktadır.

II. KULLANILAN METODOLOJİ VE TEKNİKLER

A. Roboflow

Veri setinin oluşturulması ve nesne tanıma modelinin eğitimi için Roboflow platformu kullanılmıştır. Roboflow, görüntülerin etiketlenmesi, veri artırma ve model eğitimi için uygun formatlara dönüştürülmesi gibi işlemler için etkili bir araçtır. Bu projede, Roboflow üzerinden farklı kategorilerdeki nesneler için etiketleme yapılmış ve veri seti YOLO formatına dönüştürülerek model eğitimi için hazırlanmıştır.

B. YOLOv8

Veri setinin model eğitimi sürecinde, Roboflow API kullanılarak veri setine Google Colab ortamında erişim sağlanmıştır. Eğitim süreci Google Colab üzerinde

gerçekleştirilmiş ve YOLOv8 modeli burada eğitilmiştir. Colab, GPU hızlandırmalı bir platform sunarak model eğitimi sırasında işlem sürelerini önemli ölçüde azaltmıştır. Roboflow API'nin sağladığı kolaylık sayesinde veri seti doğrudan Colab'a entegre edilmiş ve eğitim süreci sorunsuz bir şekilde ilerlemiştir.

Model eğitimi ve nesne algılama süreci için YOLOv8 tercih edilmiştir. YOLOv8, hızlı ve hassas sonuçlar sunan bir nesne algılama modelidir. Model, açık kaynaklı bir platform olan Ultralytics tarafından sağlanmıştır ve Python programlama dili kullanılarak entegre edilmiştir. YOLOv8 modeli, algılama sırasında görüntüleri gerçek zamanlı olarak analiz ederek belirli nesneleri tanımlamaktadır. Bu çalışmada kullanılan YOLOv8 modeli, "best.pt" isimli özel bir ağırlık dosyası ile çalıştırılmıştır.

C. PyCharm

Geliştirme ortamı olarak PyCharm kullanılmıştır. PyCharm, Python projeleri için etkili bir geliştirme ortamı sunan bir IDE'dir. Projenin kodlama, hata ayıklama ve entegrasyon süreçleri PyCharm üzerinden gerçekleştirilmiştir. Bu sayede, geliştirme süreci daha düzenli ve hızlı bir şekilde ilerlemiştir.

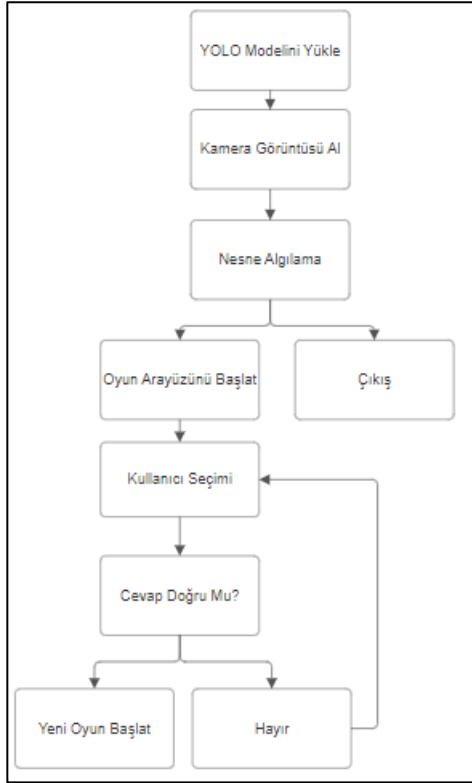


Figure 1 Blok Şeması

D. Literatür Çalışması

Dijital oyunlar, son yıllarda eğitim, eğlence ve strateji geliştirme gibi birçok alanda önemli bir yer edinmiştir. Bununla birlikte, bu oyunlar aynı zamanda çocuklar ve gençler üzerinde olumsuz etkiler yaratabilen şiddet, cinsellik, küfür gibi zararlı içerikler de barındırabilmektedir. Bu tür zararlı içeriklerin tespiti ve önlenmesi, özellikle genç yaşta bireylerin gelişimi için büyük bir önem taşımaktadır. Dijital oyunlarda şiddet unsurlarının tespiti üzerine yapılan çalışmalar, bu unsurların erken dönemde tespit edilmesi ve müdahale edilmesi gerektiğini vurgulamaktadır. (YALÇIN, N., & ÇAPANOGĞLU, A. E)

III. GELİŞTİRİLEN YAZILIMIN TANITIMI

Geliştirilen yazılım, kullanıcı dostu bir oyun arayüzü ile gerçek zamanlı nesne algılama teknolojisini birleştiren bir yapıya sahiptir. Oyun, kullanıcının kameradan alınan görüntüler aracılığıyla dört farklı nesneyi tanımasını gerektiren bir mekanizmaya sahiptir.

Kullanıcı, oyunu başlattığında kamera ile tespit edilen nesneler ekranda görüntülenir ve yazılım, her bir nesneyi algıladığında bunun kaydedilmesini sağlar. Ancak, bir nesnenin başarılı bir şekilde kaydedilebilmesi için algılama güven puanının (detection confidence) en az **0.5** olması gerekmektedir.

Oyunun kuralları şu şekilde tanımlanmıştır:

1. Kullanıcı, kamerayı kullanarak yazılımın dört farklı nesneyi algılamasını sağlamalıdır.
2. Algılanan her nesne, sistem tarafından kayıt altına alınır
3. Algılanan nesnenin kayıt altına alınabilmesi için tespit doğruluğunun 0.5 veya üzerinde olması zorunludur. Daha düşük bir doğrulukla algılanan nesneler kayıt altına alınmaz.
4. Kullanıcı, tüm nesneleri başarıyla tanıttıktan sonra oyunun bir sonraki aşamasına geçebilir.
5. Kullanıcının seçim sırasında şıklar otomatik olarak karıştırılır ve bulanık fotoğraf gösterilir.
6. Fotoğrafın altında ipucu bulunur, kullanıcı doğru tahmin etmesi halinde fotoğrafın bulanıksız hali gösterilir ve diğer fotoğrafa geçilir aksi takdirde bilene kadar tekrarlanır.



Figure 2 Bulanık hale getirilen örnek fotoğraf



Figure 3 Doğru bilindiğinde oluşan arayüz

IV. DENEYSEL SONUÇLAR

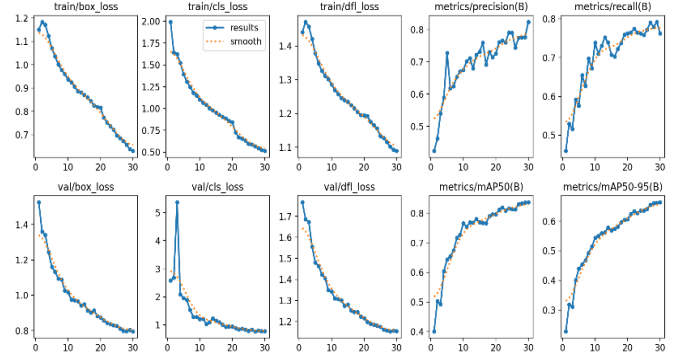


Figure 4 Epoch Loss Verileri (YOLOv8)

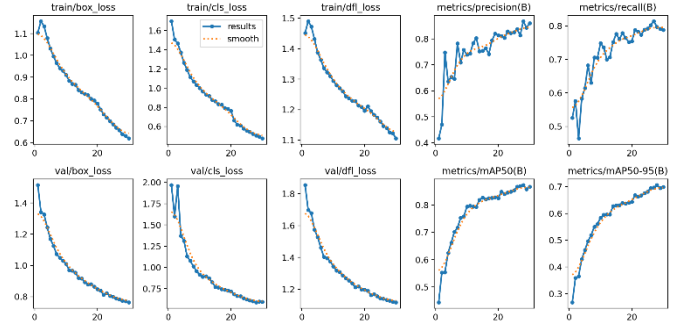


Figure 5 Epoch Loss Verileri (YOLOv11)

Model eğitim aşamasında YOLOv8 'de sınıflandırma eğitimi YOLOv11'e göre daha iyi yapmış fakat test kısmında YOLOv11 daha iyi olduğunu görüyoruz.

Model Performansı genel olarak, modelin eğitim ve doğrulama verisi üzerindeki performansı iyidir. Özellikle precision ve recall metriklerinin yüksek olması, modelin sınıflandırma görevini başarılı bir şekilde yerine getirdiğini gösterir. mAP metriklerinin de yüksek olması, modelin nesne tespiti görevini de başarılı bir şekilde yerine getirdiğini gösterir.

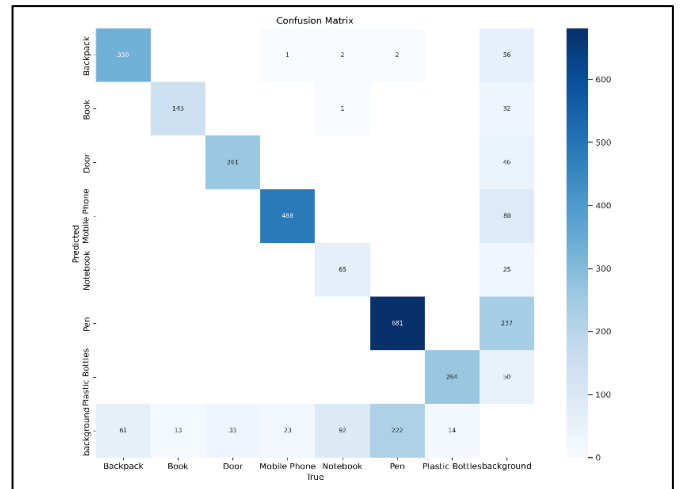


Figure 6 Karmaşıklık matrisi

Matrise göre, modelin genel olarak iyi bir performans gösterdiği söylenebilir. Diagonal üzerindeki yüksek değerler, modelin çoğu örneği doğru sınıflandırdığını gösterir.

- "pen" sınıfı: Model, "pen" sınıfını diğer sınıflara göre daha iyi bir şekilde sınıflandırmıştır. Bu, modelin "pen" nesnelerini tanımada oldukça başarılı olduğu anlamına gelir.
- "background" sınıfı: "background" sınıfında daha fazla yanlış sınıflandırma olduğu gözlemlenmektedir. Bu, modelin arka planı diğer nesnelerden ayırt etmekte zorlandığını gösterir.
- Diğer Sınıflar: Diğer sınıflar arasında da bazı yanlış sınıflandırmalar bulunmaktadır. Özellikle benzer özelliklere sahip sınıflar arasında karışıklıklar olabilir (örneğin, "book" ve "notebook").

KAYNAKÇA

- 1) <https://docs.ultralytics.com/tr/models/yolov8/>
- 2) *OpenCV Documentation*. (n.d.). *Open Source Computer Vision Library*
- 3) YALÇIN, N., & ÇAPANOĞLU, A. E. (2023). *Detection of Violent Elements in Digital Games Using YOLO V7 Algorithm*.
- 4) Agarwal, V., Singh, M., & Prathap, B. R. (2024, August). *Enhanced Multi-Model Approach for Motion and Violence Detection using Deep Learning Methods Using Open World Video Game Dataset*. In *2024 First International Conference on Pioneering Developments in Computer Science & Digital Technologies (IC2SDT)* (pp. 1-6). IEEE.
- 5) <https://docs.ultralytics.com/tr/models/yolo11/>