

YOLO Gerçek Zamanlı Nesne Tespiti

1. Giriş

Bilgisayarla görme alanındaki en önemli problemlerden biri olan nesne tespiti, özellikle güvenlik, otonom sistemler, artırılmış gerçeklik ve sağlık alanlarında yaygın olarak kullanılmaktadır. Geleneksel nesne tespiti yöntemleri, genellikle kaydırma penceresi veya bölgesel tabanlı yaklaşımlar kullanarak nesneleri algılamaktadır (Girshick et al., 2014). Ancak, derin öğrenme tekniklerinin gelişmesiyle birlikte, nesne tespiti modelleri daha hızlı ve doğruluk oranı yüksek hale gelmiştir. Bu bağlamda, **You Only Look Once (YOLO)** algoritması, nesne tespiti için yenilikçi bir yaklaşım sunarak, gerçek zamanlı performansı ve yüksek doğruluk oranı ile dikkat çekmektedir (Redmon & Farhadi, 2018).

2. Nesne Tespiti ve Derin Öğrenme Yöntemleri

Nesne tespitinde kullanılan geleneksel yöntemler genellikle Haar Cascade (Viola & Jones, 2001) ve HOG+SVM (Dalal & Triggs, 2005) gibi özellik tabanlı yaklaşımlardır. Bu yöntemler belirli durumlarda başarılı olsa da, büyük veri kümelerinde düşük genel performans göstermektedir.

Derin öğrenme tabanlı yöntemler ise Convolutional Neural Networks (CNN) kullanarak nesne tespiti yapmaktadır. Bu kapsamda en yaygın yöntemlerden bazıları şunlardır:

- **R-CNN (Girshick et al., 2014):** Bölgesel öneri tabanlı nesne tespit algoritması.
- **Faster R-CNN (Ren et al., 2015):** R-CNN'in geliştirilmiş ve hızlandırılmış versiyonu.
- **SSD (Liu et al., 2016):** Çoklu ölçekli tespit yaparak gerçek zamanlı performans sağlayan model.
- **YOLO (Redmon et al., 2016):** Nesne tespitini tek bir ileri beslemeli ağda gerçekleştiren ve yüksek hız sunan model.

3. YOLO Algoritması ve Gelişim Süreci

YOLO, nesne tespiti görevini bir regresyon problemi olarak ele alarak, giriş görüntüsünü tek bir aşamada değerlendirir. YOLO'nun farklı sürümleri, doğruluk ve hız açısından önemli gelişmeler sağlamıştır:

- **YOLOv1 (Redmon et al., 2016):** İlk versiyon, nesne tespitini tek bir evrişimli sinir ağı (CNN) kullanarak gerçekleştirmiştir.
- **YOLOv2 (Redmon & Farhadi, 2017):** Daha yüksek doğruluk oranı için Batch Normalization ve Anchor Box kullanımı eklenmiştir.
- **YOLOv3 (Redmon & Farhadi, 2018):** Çoklu ölçekli tahminleme ve Darknet-53 mimarisi ile geliştirilmiştir.
- **YOLOv4 (Bochkovskiy et al., 2020):** CSPDarknet-53 ve optimizasyon teknikleriyle performans artırılmıştır.
- **YOLOv5 (Jocher et al., 2021):** Daha hafif ve optimize edilmiş bir model olarak dikkat çekmektedir.
- **YOLOv7 ve YOLOv8 (Wang et al., 2022):** Daha hızlı ve hassas tespit yapan en yeni sürümler olarak literatüre geçmiştir.

4. YOLO'nun Gerçek Zamanlı Kullanım Alanları

YOLO algoritması, gerçek zamanlı nesne tespiti gerektiren birçok alanda başarıyla kullanılmaktadır. Literatürde yer alan bazı önemli kullanım alanları şunlardır:

- **Otonom Araçlar:** Trafik işaretleri ve yayaları tespit etmede kullanılır (Geiger et al., 2013).
- **Güvenlik ve Gözetim:** Yüz tanıma ve anormal davranış tespiti için uygulanmaktadır (Haque et al., 2020).
- **Sağlık Sektörü:** Radyoloji görüntülerinde anormallik tespiti yapmaktadır (Lakhani & Sundaram, 2017).
- **Endüstriyel Uygulamalar:** Üretim hatlarında kalite kontrol ve hata tespiti için kullanılmaktadır (Kim et al., 2021).

5. YOLO ve Diğer Modellerin Karşılaştırılması

YOLO, Faster R-CNN ve SSD gibi diğer nesne tespiti modelleri ile karşılaştırılmıştır. Bu karşılaştırmalara göre YOLO, hız açısından avantaj sağlarken, doğruluk açısından Faster R-CNN bazı durumlarda daha iyi sonuçlar vermektedir (Ren et al., 2015). Aşağıda bazı karşılaştırma sonuçları verilmiştir:

| Model | Doğruluk (mAP) | Çalışma Hızı (FPS) |
|--------------|----------------|--------------------|
| Faster R-CNN | 76.4% | ~5 FPS |
| SSD | 74.3% | ~22 FPS |
| YOLOv3 | 57.9% | ~45 FPS |
| YOLOv5 | 50.2% | ~140 FPS |

Bu verilere göre, YOLO düşük gecikme süresi gerektiren gerçek zamanlı uygulamalar için daha uygun bir model olarak öne çıkmaktadır.

6. Sonuç ve Değerlendirme

YOLO, gerçek zamanlı nesne tespitinde hız ve verimlilik açısından önemli avantajlar sunmaktadır. Literatürde yapılan çalışmalar, YOLO'nun özellikle gerçek zamanlı uygulamalar için tercih edildiğini göstermektedir. Ancak, YOLO'nun küçük nesneleri tespit etme başarısının Faster R-CNN gibi diğer yöntemlere kıyasla daha düşük olduğu belirtilmektedir. Gelecekte, hibrit modellerin geliştirilmesi ve donanım optimizasyonları ile YOLO'nun performansının daha da artırılması beklenmektedir.

Kaynaklar

- Redmon, J., & Farhadi, A. (2018). YOLOv3: An Incremental Improvement. arXiv preprint arXiv:1804.02767.
- Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster R-CNN: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28.
- Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. (2016). SSD: Single Shot MultiBox Detector. *European conference on computer vision*, Springer.
- Bochkovskiy, A., Wang, C. Y., & Liao, H. Y. M. (2020). YOLOv4: Optimal Speed and Accuracy of Object Detection. arXiv preprint arXiv:2004.10934.
- Jocher, G., Chaurasia, A., & Stoken, A. (2021). ultralytics/yolov5: v5.0. Zenodo. DOI: 10.5281/zenodo.4679653.