



Demiryolu Ray ve Çevresinin Anlamlandırılması için Derin Öğrenme Tabanlı Yöntemlerin Karşılaştırmalı Analizi

İlhan AYDIN¹, Taha Kubilay ŞENER¹, Mehmet SEVİ^{*2}

¹ Fırat Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, Elâzığ, Türkiye

² Muş Alparslan Üniversitesi, Bilgi İşlem Daire Başkanı, Muş, Türkiye

m.sevi@alparslan.edu.tr

(Alınış/Received: 02.08.2023, Kabul/Accepted: 14.09.2023, Yayınlanma/Published: 31.01.2024)

Öz: Demiryollarında ray güvenliği tren kazalarının önlenmesi için oldukça önemlidir. Ray çevresinde ve üzerinde bulunan nesnelere tren için tehlike arz etmektedir. Dolayısıyla demiryoluna izinsiz girişlerin tespit edilerek trenlerin güvenli çalışması akıllı ulaşım sistemleri için oldukça önemlidir. Bu çalışmada raylı sistemlerde ray çevresinin anlamlandırılması amacıyla görüntü bölütleme tabanlı yaklaşımlar karşılaştırılmış ve ray çevresindeki nesnelere tespiti sağlanmıştır. Görüntü bölütleme tabanlı ray ve çevresinin anlamlandırılması için UNet, BiSeNetV2, DeepLabV3 ve PP-LiteSeg modelleri karşılaştırmalı olarak analiz edilmiştir. Ayrıca ray çevresindeki nesnelere tespitinde YOLOv7 uygulanmıştır. Böylece, modellerin gerçek dünya senaryolarında ne kadar başarılı olduğu değerlendirilmiştir. Deneyler sonucunda, hafif yapısıyla dikkat çeken PP-LiteSeg modelinin yüksek segmentasyon performansı gösterdiği tespit edilmiştir. Eğitim aşamasının nesne tespitinde önemli olduğu görülmüş ve PP-LiteSeg'in Jetson Nano gibi tek devre kartlarda başarılı bir şekilde uygulanabildiği sonucuna ulaşılmıştır. Çalışmadaki bir diğer model YOLOv7, TensorRT kütüphanesi kullanılarak paralel çalışacak şekilde optimize edilmiş ve hafıza alanlarının bağımsız olarak kullanılabilmesi için özel bir kontrol mekanizması geliştirilmiştir. Elde edilen sonuçlara göre, PP-LiteSeg modelinin diğer modellere göre daha yüksek doğruluk ve mIoU değerleri elde ettiği görülmüştür. Yapılan çalışma raylı sistemlerde hızlı ve doğru nesne tespiti için segmentasyon modellerinin seçimine yönelik önemli sonuçlar içermektedir. Çalışma PP-LiteSeg modelinin kullanımıyla birlikte sınırlı kaynağa sahip ortamlarda bile yüksek kalitede nesne tespiti yapılabileceğini kanıtlamıştır.

Anahtar kelimeler: Derin öğrenme, Demiryolu, Nesne tespiti, Akıllı ulaşım, YOLO, Semantik segmentasyon

Comparative Analysis of Deep Learning-Based Methods for Making Sense of Railway and Its Environment

Abstract: Rail safety in railways is very important for the prevention of train accidents. Objects around and on the rails pose a danger to the train. Therefore, the safe operation of trains by detecting unauthorized access to the railway is very important for smart transportation systems. In this study, image segmentation-based approaches are compared in order to make sense of the rail environment in railway systems, and the objects around the rail are detected. UNet, BiSeNetV2, DeepLabV3, and PP-LiteSeg models were analyzed comparatively to segment of the rail and its environment based on image segmentation. In addition, YOLOv7 has been applied to detect objects around the rail. Thus, it was evaluated how successful the models were in real-world scenarios. As a result of the experiments, it was determined that the PP-LiteSeg model, which stands out with its lightweight structure, showed high segmentation performance. It has been seen that the training phase is important in object detection, and it has been concluded that PP-LiteSeg can be successfully applied on single circuit boards such as Jetson Nano. Another model in the study, YOLOv7, has been optimized to run in parallel using the TensorRT library. A special control mechanism has been developed to use memory areas independently. According to the results obtained, it was seen that the PP-LiteSeg model achieved higher accuracy and mIoU values than other models. The study includes important results for the selection of segmentation models for fast and accurate object detection in rail systems. The study proved that with the use of the PP-LiteSeg model, high-quality object detection can be achieved even in environments with limited resources.

Atıf için/Cite as: İ. Aydın, T. K. Şener, M. Sevi, "Demiryolu ray ve çevresinin anlamlandırılması için derin öğrenme tabanlı yöntemlerin karşılaştırmalı analizi," *Demiryolu Mühendisliği*, no. 19, pp. 1-16, Jan. 2024. doi: 10.47072/demiryolu.1336812

Keywords: Deep learning, Railway, Object detection, Intelligent transportation, YOLO, Semantic segmentation

1. Giriş

Demiryolu ve trenlerdeki hızlı gelişim ile demiryolu ulaşımı yük taşımacılığının yanı sıra yolcu taşımacılığında da önemli bir alan haline gelmiştir. Demiryolu hatlarında sıklıkla kazalar meydana gelmekte olup ciddi yaralanma ve ölümler ile sonuçlanabilmektedir [1]. Tren güvenliği mevcut sistemlerde makinistin görüş alanı ve dikkatine bağlıdır. İstasyona giriş çıkışlarda tren düşük hız ile ilerlemekte olup makinistin görüşü yeterli olmaktadır. Fakat hava koşulları değişimi veya yorgunluk gibi etkenlerden dolayı makinist bazı izinsiz görüşleri kaçırabilmektedir [2]. Bu yüzden ray ve çevresinde tehdit oluşturabilecek nesnelerin gerçek zamanlı ve otomatik tespiti önem arz etmektedir. Raylı sistemlerin bir bileşeni olan raylar atmosferik koşullar, coğrafik farklılıklar, doğal felaketler (heyelan, deprem vb.) ve kullanım süresine bağlı olarak deformasyona uğrayabilir. Dolayısıyla rayların zaman zaman kontrol edilmesi ve denetimlerin yapılması oldukça önemlidir. Ray üzerinde çoklu ortam koşullarında nesne tespitinin yapılması, olası hataların tespit edilmesi, ray kavisleri, yabancı cisimler ve benzeri gibi birçok unsur nesne tespitinde güçlüğüne sebep olabilmektedir. Farklı durumlar için, nesne tespit yöntemleri çeşitlilik gösterir ve bu yöntemler, modele, niteliğe, olasılığa ve görünüme bağlı olarak değişmektedir [3-4]. Nesne tespiti, bir görüntü veya video içerisinde belirli nesnelerin konumlarını ve sınıflarını tespit etmeyi hedefler. Ancak görüntülerde olası ışık değişimleri nesne tespiti işlemlerini olumsuz etkileyebilir.

Demiryollarında izinsiz giriş tespiti için literatürde bazı çalışmalar yapılmıştır. Yapılan çalışmalar daha çok görüntü işleme ve makine öğrenmesi yöntemlerinden faydalanmaktadır [5]. Özellikle bilgisayarlı görme ve nesne tespit sistemlerindeki gelişmeler derin öğrenme yöntemlerinin kullanımı ile daha doğru sonuçlar vermektedir. Derin öğrenme teknikleri, geleneksel tekniklere kıyasla probleme özgü farklı özelliklerin otomatik olarak öğrenilmesine yardımcı olmaktadır [6-8]. Belirli noktalara yerleştirilen video gözetim sistemleri sayesinde ray ve çevresindeki insanlar, hayvanlar ve araçlar otomatik olarak tespit edilebilmektedir [3]. Ye ve ark. [2] demiryolundaki izinsiz girişleri tespit etmek amacıyla az veri kullanan düşük ağırlıklı bir derin öğrenme modeli sunmuşlardır. Az veri kullanan sistemin performansını geliştirmek için ince ayar özelliği ve sınıflandırma modeli önermişlerdir. Geliştirilen sistemin tek kamera ve gömülü bir bilgisayar üzerinde prototipi oluşturulmuştur. Zheng ve ark. [4] metro istasyonlarında ray ve çevresindeki anormallikleri tespit etmek amacıyla sıralı güncellenebilir bir anormallik tespit yöntemi önermiştir. Önerilen sistem ile girişlerinde yasak bölgede bulunan kişilerin tespit edilmesi amaçlanmıştır. Gong ve ark. [9] demiryolu hatlarında izinsiz giriş tespiti için bölütleme ve birkaç adımlı öğrenme tabanlı çözüm önermiştir. Önerilen yaklaşım farklı hava koşullarında test edilmiş ve doğru sonuçlar elde edilmiştir. Demiryolu sistemlerinde video gözetimi ve yapay görme ile çevre ihlallerini tespit etmek amacıyla çoklu görev tabanlı bir sinir ağı modeli önerilmiştir [10]. Önerilen yaklaşım ilk olarak ufuk noktası tespiti için bir regresyon ağını eğitmektedir. Daha sonra kaybolan ufuk noktasının tespiti için kodlayıcı-kod çözücü bir bölütleme yaklaşımı kullanılmaktadır. Son aşama ise kodlayıcı-kod çözücü ile bölütleme sonucu ufuk noktasının işaretlenmesidir. Ding ve ark. [11] ray üzerindeki yabancı nesnelere tespit etmek amacıyla YOLOv5 tabanlı bir yaklaşım önermiştir. Önerilen yaklaşım diğer nesne tespit yöntemleri ile karşılaştırılmış ve daha doğru sonuçların elde edildiği deneysel olarak belirlenmiştir. Oluşturulan veri seti gerçek ray verileri yerine el ile oluşturulmuş verilerden oluştuğundan gerçek zamanlı uyarlanabilirliği düşüktür. Demiryolu hattına izinsiz girişlerin tespiti için EfficientNet tabanlı tek atışlı çoklu kutu tespit edicisi (Single Shot MultiBox Detector-SSD) önerilmiştir [3]. Önerilen yaklaşım SSD'deki VGG16 omurgası yerine EfficientNet omurgası kullanılmıştır. Böylece CPU üzerinde de saniyede 11 frame işlenecek şekilde gerçek zamanlı bir yaklaşım önerilmiştir. Fakat

eğitimde kullanılan veriler Pascal VOC veri setinden olup ray ve çevresinin anlamlandırılması ve ray üzerinde bir nesnenin belirlenmesini içermemektedir. Ray üzerinde bulunan nesnelerin insansız hava aracından alınan görüntülerden tespiti için evrişimsel uzun kısa süreli hafıza ağı tabanlı bir hibrit yöntem önerilmiştir [12]. Önerilen yaklaşım ile rayın güvenli olup olmadığı şeklinde iki sınıflı bir sınıflandırma yapılmaktadır. Cao ve ark. [13] ray ve çevresindeki izinsiz girişleri belirlemek amacıyla düşük ağırlıklı bir derin sinir ağı mimarisi önermişlerdir. Önerilen yaklaşım geleneksel görüntü işleme ile derin öğrenme yöntemlerini birleştirmektedir. Ayrıca giriş görüntüleri sıkıştırılmadığı için küçük boyutlu nesnelerin tespit performansı da artırılmıştır. Ray ve çevresinin anlamlandırılması için UNet tabanlı bir performans artırma yöntemi önerilmiştir [14]. Önerilen yöntem UNet'in omurga kısmında VGG16, Resnet34 ve MobileNetV2'yi kullanan modellerin birleştirilmesine dayalıdır.

Literatürde ray ve çevresinin anlamlandırılması için yapılan çalışmaların büyük bir kısmı daha çok ray ve çevresine izinsiz girişlerin tespit edilmesi üzerinedir. Literatürde bölütleme tabanlı ray ve çevresinin anlamlandırılması, ray tespiti ve tespit edilen ray bölgesine göre izinsiz giriş tespiti ile ilgili çalışma bulunmamaktadır. Nesne tespiti olarak ele alınan izinsiz giriş tespitinde nesnelerin rayın üzerinde olup olmadığına bakılmamaktadır. Bu çalışmada ray ve çevresinin anlamlandırılması için dört farklı derin öğrenme modeli kullanılmıştır. Bu modellerden UNet ve DeepLabV3 gibi klasik modellerin yanı sıra BiSeNetV2 ve PP-LiteSeg gibi gerçek zamanlı bölütleme yaklaşımları test edilmiştir. Bölütleme için kullanılan yöntemler bir kıyaslama veri seti üzerinde test edilmiş ve tespit performansı ile çalışma süresine göre en iyi performansı veren model belirlenmiştir. Ayrıca görüş açısına göre trene yakın olan iki ray bileşeni bölütlenerek ray üzerinde olan izinsiz girişler YOLOv7 nesne tespit algoritması ile tespit edilmiştir. Çalışmanın özgün katkıları aşağıda sıralanmıştır.

- Ray ve çevresinin gerçek zamanlı olarak bölütlenmesi için derin öğrenme tabanlı yöntemlerin karşılaştırılması,
- Ray ve çevresindeki izinsiz girişlerin tespit edilmesi,
- Ray bölgesinin çıkarılarak görüntü içerisinde belirlenmesi ve takibi için yeni bir yaklaşım sunulması
- Piksel tabanlı görüntü segmentasyonunda kullanılan hafif ve etkili bir derin öğrenme modeli olan PP-LiteSeg modeli ile demiryolu sistemleri ve hareket analizinin gerçek zamanlı olarak hassas bir şekilde belirlenmesi,
- Kameralı bir İHA ile pilotaj olmadan demiryolu rayı hattını takibini sağlamak,
- Bilgisayarlı görme sistemi geliştirilerek belli boyutlardaki farklı nesnelerin tanınması, otomatik olarak hareketli nesnenin belirlenmesi ve takibi.

2. Ray ve Çevresinin Anlamlandırılması için Derin Öğrenme Tabanlı Yaklaşım

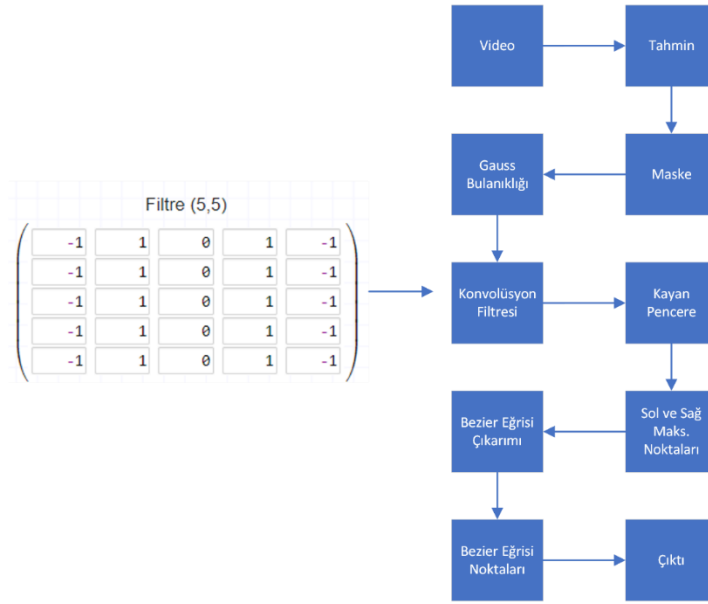
Bu çalışmada ray ve çevresinin anlamlandırılması için farklı bölütleme yöntemleri test edilmektedir. Eğitim aşamasında ray ve çevresindeki nesnelerin bölütlenmesi için derin öğrenme tabanlı yaklaşımlar kullanılmaktadır. Bu amaçla UNet [15], DeepLabV3 [16], BiSeNetV2 [17] ve PP-LiteSeg [18] yöntemleri kullanılmaktadır. Daha test aşamasında bölütlenen görüntüden sadece ray bölgesi belirlenmekte ve ray takibi için rayların işaretlenmesi sağlanmaktadır. Aynı zamanda ray çevresindeki tehdit oluşturan nesnelerin tespiti YOLOv7 [19] nesne tespit yöntemi ile tespit edilmektedir. Şekil 1'de önerilen yaklaşımın akış diyagramı verilmiştir.



Şekil 1. Ray ve çevresinin anlamlandırılması için önerilen yaklaşım

Şekil 1’de demiryolu sistemleri üzerinde ray ve çevresinin anlamlandırılması için Railsem19 veri seti üzerinde UNet, BiSeNetV2, DeepLabV3 ve PP-LiteSeg modelleri ayrı ayrı test edilmiştir. En iyi başarımları ve hız performansını veren bölütleme yaklaşımını daha sonra seçilmektedir. Bölütlenen görüntüden ray bölgesini içeren alan çıkarılmakta ve ray takibi ve izinsiz giriş için ilgili bölge belirlenmektedir. Daha sonra ilgili bölgedeki izinsiz girişler YOLOv7 nesne tespit modeli ile tespit edilmektedir. Bu amaçla, COCO veri seti kullanılmaktadır. Hibrit modelde, iki modelin harmonik bir şekilde bir arada çalışması için bazı değişiklikler yapılmalıdır. Çünkü her iki model de sınırlı bir kaynağa sahip olan ekran kartı hafızasını ve CUDA çekirdeklerini kullanarak çıkarım yapmaktadır. Dolayısıyla modellerin ayrı hafıza alanlarında bağımsız olarak çıkarım yapabilmeleri için özel bir kontrol mekanizması gerekmektedir. Bu kontrol mekanizması sayesinde, her iki modelin hatalarının birbirlerinin hafıza alanlarını işgal etmeden, çıkarım yapması kritik öneme sahiptir. Bu gereksinimi sağlamak için TensorRT kütüphanesi kullanılmıştır. TensorRT, segmentasyon ağına öncelik vererek modelin gereksinim duyduğu hafızayı almasını sağlar ve geri kalan hafıza alanını en iyi şekilde kullanarak iki modelin paralel olarak çalışmasını sağlar. Bu şekilde, ek bir bekleme süreci olmaksızın ortak sonuçlar elde edilmesi mümkün olur.

İki modelin çıkarımına ek olarak, bir konvolüsyon filtresi yardımı ile demiryolu rayları üzerinde yapılan çıkarımın etraftaki gürültü olarak oluşmuş veya trenin üzerinde olmadığı rayların çıkarım yapılması engellenmiştir. Şekil 2’de görüldüğü gibi kullanılan bu yöntem sayesinde sol ve sağ rayların birbirinden ayrı biçimde oluşturulabilmesi mümkün olmuştur. Demiryolu raylarının iç kısmında kalan bölümün rahatlıkla çıkarımının yapılabilmesiyle, hem raylar üzerinde yapılacak incelemelerin daha kolay olması hem de ray üzerinde kalan alan içerisindeki nesnelerin diğer objelerden ayırt edilmesi sağlanmıştır. Çünkü raylı sistemler çoklu ortam koşullarına sahip olup, bir görüntüde birden çok nesne bulunabilmektedir.



Şekil 2. Sol ve sağ rayların bulunması

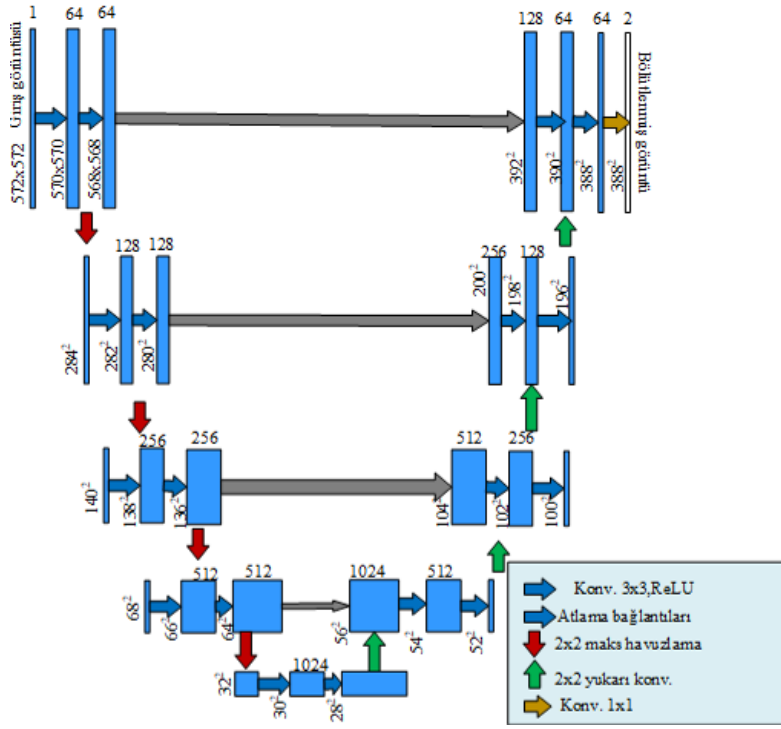
Çalışmada demiryolu rayı, elektrik direkleri, yerleşim alanları, hareketli nesne (kuş, araç vs.) ayrı ayrı sınıflara bölünmüştür. Kullanılan Railsem19 [20] veri seti 8500 adet görüntüden oluşmaktadır. Görüntüler 21 alt sınıfa ayrılmıştır. Her bir sınıf için görüntü artırımı teknikleriyle 8500'in üzerinde etiketlenmiş görüntü verisi temin edilmiştir. Görüntü artırımı teknikleri olarak parlaklık, görüntü çevirme ve gürültü ekleme kullanılmıştır.

2.1. YOLOv7

YOLO (You Only Look Once-Sadece bir kez bak) nesne tespit algoritması, Redmon ve ark. [21] tarafından gerçek zamanlı nesne tespiti yapılabilmesi için geliştirilen bir algoritmadır. Araştırmacılar, bu yaklaşımla doğrudan piksellerden nesnelerin yerini hızlıca belirleyebildiler. Ayrıca, Titan X GPU ile SSD de 45 resim saniyeler içinde belirlenirken, YOLO'da saniyede 150 resim belirlenebiliyor olduğu görüldü. YOLO algoritması, R-CNN mimarisinde belirlenen alanların çıkartılması, alanların sınıflandırılması problemini bir ağ ile çözebilme özelliğine sahiptir. Bu algoritma geleneksel nesne tespit yöntemlerinin aksine sınıf olasılıklarının hesaplanması, sınırlayıcı kutuların bulunması ve hemen hemen tüm işlemleri tek bir regresyon problemi olarak değerlendirilmesini sağlamaktadır. Yani, YOLO ile görüntü üzerinde hangi nesnelerin nerede olduklarını tespit etmek için görüntüye yalnızca bir kez bakılması yeterlidir. YOLO tespit modeli, eğitim seti kapsamındaki her görüntüyü NxN kafeslere böler. Bu kafeslerin her biri bir hücreyi temsil eder. Modelin sorumluluğu, belirlenen nesnelerin merkez konumunu tespit etmek ve merkezin bulunduğu hücreyi o nesneyi tespit etmekten sorumlu tutmaktır. Bu çalışmada son popüler YOLO modellerinden YOLOv7 nesne tespiti için kullanılmaktadır.

2.2. UNet

UNet 2015 yılında Ronneberger ve ark. [15] tarafından sunulmuş, bir kodlayıcı (encoder) ve çözücü (decoder) yapısından oluşan bir mimardır. Kodlayıcı yolu, tipik bir evrişimli sinir ağı gibi giriş görüntüsünü aşağı örnekleme yaparak yüksek seviyeli özellikleri çıkarırken mekânsal bağlamı yakalar. Her kodlayıcı blok sırasıyla konvolüsyonel katmanlardan, ReLU aktivasyon fonksiyonundan ve maksimum havuzlama işlemlerinden oluşmaktadır. Aşağı örnekleme işlemi mekânsal boyutları azaltırken özellik kanallarının sayısını artırmaktadır. Çözücü yolu ise kodlayıcı yolundan gelen düşük çözünürlüklü özellik haritalarını orijinal görüntü boyutuna yükseltme işleminden sorumludur. Şekil 3'te UNet mimarisi verilmiştir.



Şekil 3. UNet model mimarisini

Şekil 3'te her çözücü blok, kodlayıcı yolundaki karşılık gelen özellik haritalarıyla yukarı örnekleme edilen özellik haritalarını birleştirir. Bu atlama bağlantıları, ağırlıklı ölçeklerde hem yerel hem de global bağlamsal bilgiyi yakalamasını sağlar. Atlama bağlantılarının birleştirilmesi, kodlayıcı yolundan gelen ince ayrıntıların yükseltme işlemi sırasında korunmasını sağlar. Bu bağlantılar, düşük seviyeli özellikleri ve mekânsal bilgiyi çözücüye aktararak segmentasyon sınırlarını iyileştirir ve daha doğru segmentasyon sonuçları üretmeye yardımcı olur. UNet, tıbbi görüntü segmentasyonu, hücre çekirdeği segmentasyonu ve otonom sürüşte yol segmentasyonu gibi çeşitli uygulamalarda yaygın olarak kullanılmaktadır. Başarısı, sınırlı eğitim verisiyle başa çıkma yeteneği ve yerel ve global bağlamsal bilgiyi yakalama etkinliği ile ilişkilendirilebilir.

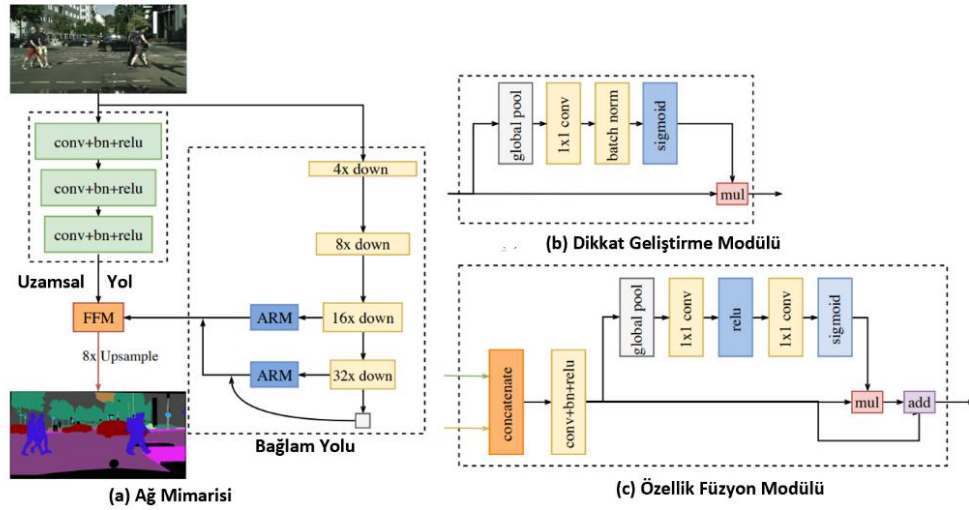
2.3. DeepLabV3

Google tarafından geliştirilen bu model bilgisayarlı görüde yaygın olarak kullanılan bir derin öğrenme modelidir. Özellikle piksel tabanlı görüntü segmentasyonunda etkili sonuçlar vermektedir [16]. Evrişimsel sinir ağı tabanlı mimariye sahip olup, daha önceki versiyonlarına kıyasla oldukça yüksek bir segmentasyon performansı sunmaktadır. Bu modelde, dilatasyon konvolüsyon ve atrous konvolüsyon gibi yenilikçi teknikler sayesinde daha büyük bir reseptif alan elde edilir ve ayrıntılar daha iyi bir şekilde yakalanabilmektedir [22]. Mimari olarak DeepLabV3, derinlik bilgisi eklemek için önceden eğitilmiş bir evrişimli sinir ağını omurga olarak kullanır. Bu omurga modeli, görüntüyü önce birkaç evrişimli ve havuzlama katmanından geçirerek özellik haritalarını çıkarır. Tipik olarak, Xception, ResNet ve MobileNet gibi güncel omurga modelleri kullanılır. Aşırı Uzaysal Piramit Havuzu (ASPP) modülü kullanılarak çıkarılan özellik haritaları işlenir. Bu modül, farklı dilatasyon oranlarına sahip atrous konvolüsyon katmanları içermesi sebebiyle, farklı ölçeklerdeki nesnelere ve nesnelere detaylarını doğru bir şekilde yakalayabilmesine olanak sağlar. Özellik haritaları piksel tabanlı sınıflandırma yapmak için bir konvolüsyonel ağına verilir. Bu aşamada, çıktı haritası elde edilir ve her pikselin sınıf etiketi belirlenir. Genellikle, çıktı haritasını daha pürüzsüz hale getirmek için bir rötüş işlemi olan eşikleme veya Gauss filtresi uygulanır. Görüntü segmentasyonunda bu model üstün bir performansa sahiptir.

2.4. BiSeNetV2

BiSeNetV2 (Bilateral Segmentation Network- İkili Segmentasyon Ağı V2), görüntü segmentasyonunda kullanılan bir derin öğrenme modelidir. BiSeNetV2, tamamen konvolüsyonel bir ağıdır. Bu model, gerçek zamanlı uygulamalarda hızlı ve yüksek doğruluk sağlayabilen bir segmentasyon yöntemi sunar [17]. BiSeNetV2, biri yol-odaklı bir diğeri detay-odaklı olarak iki farklı özellik çıkarımını sunar. Yol-odaklı özellik uzun menzilli bağlamsal bilgiyi yakalamak için geniş bir görüntü alanına odaklanırken, detay-odaklı özellik kısa menzilli detay bilgisini korumak için daha yakından odaklanır. Bu iki yol birleştirilerek hem genel bağlamı hem de detayları yakalayan bir segmentasyon ağı oluşturulur.

BiSeNetV2 modeli sınıf-etiketi ve kenar pikselleri tahmin edilebilmektedir. Böylece nesnelerin keskin kenarları korunurken, aynı zamanda doğru bir şekilde segmentasyon yapılabilmesi sağlanmaktadır. Ayrıca, yanıtlama haritalarını kullanarak segmentasyon performansını geliştirmiştir. BiSeNetV2, hızlı ve doğru segmentasyon sonuçları sağlayan, ölçeklenebilir bir segmentasyon ağıdır. Bu model, sürücüsüz araçlar, robotik, video analizi ve diğer birçok uygulama alanında kullanılmaktadır. BiSeNetV2 modeli Şekil 4'te verilmiştir.

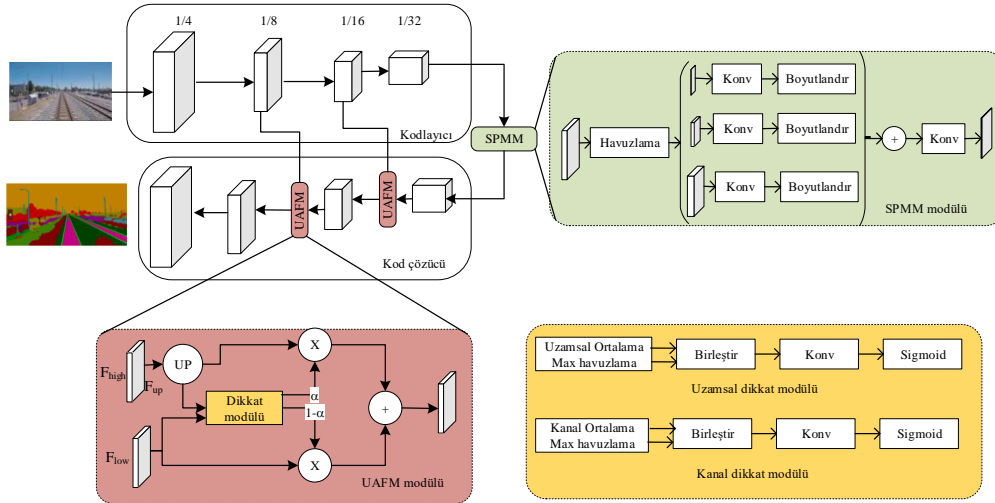


Şekil 4. BiSeNetV2 model mimarisi [17]

Şekil 4'te görüldüğü gibi iki farklı ölçekli gözetim yöntemini ile hem yüksek hem de düşük ölçekli özellik haritalarının segmentasyon sonuçlarıyla eğitimini sağlayarak daha iyi segmentasyon performansı elde eder. Bu, hem genel nesne yapısını doğru bir şekilde tahmin etmeyi hem de daha küçük nesnelerin detaylarını yakalamayı sağlar [17].

2.5. PP-LiteSeg

PP-LiteSeg (Progressive Perception Lite Segmentation- Aşamalı Algılayıcı Basit Segmentasyon) piksel tabanlı görüntü segmentasyonunda kullanılan hafif ve etkili bir derin öğrenme modelidir [18]. Bu model, yüksek performans sergilerken daha az hesaplama gücü ve bellek kullanımı gerektiren uygulamalara odaklanmıştır. PP-LiteSeg, önceden eğitilmiş bir evrişimli sinir ağı ve birden fazla aşamalı algılayıcı modülünden oluşan bir ilerleyici öğrenme stratejisini birleştirir. Bu strateji, segmentasyon performansını aşamalı olarak artırarak daha hafif bir modele olanak tanır. Şekil 5'te PP-LiteSeg modelinin şematik bir görünümü verilmiştir.



Şekil 5. PP-LiteSeg bölütleme mimarisi

Şekil 5'te verilen PP-LiteSeg mimarisi kodlayıcı-kod çözücü ve gerçek zamanlı çalışan bir bölütleme yöntemidir. Bu model kodlayıcı-kod çözücü tabanlı bir bölütleme yöntemi olup birleşik dikkat füzyon modülü (Unified Attention Fusion Model-UAFM) ve basit piramit havuzlama modülüne (Simple Pyramid Pooling Module-SPPM) sahiptir. UAFM çerçevesi bir α ağırlığı üretmek amacıyla dikkat modülü kullanılmaktadır. F_{high} ve F_{low} olarak verilen giriş özelliklerini α ve $1-\alpha$ ile çarpılarak çıkış olan F_{out} 'u elde eder. Dikkat modülü olarak kanal ve uzamsal modüller kullanılmaktadır. Uzamsal dikkat modülü giriş özelliklerinde her bir pikselin önemini gösterirken kanal dikkat modülü ise giriş özelliklerinde her bir kanalın önemini vurgular. SPPM modülü ise giriş özelliklerini kaynaştırmak için piramit havuzlama modülü kullanır. Bu havuzlama modülü boyutları 1×1 , 2×2 ve 4×4 olan üç havuzlama modülüne sahiptir. Boyutlandırma aşamasından sonra ise toplama ve evrişim modülleri bulunmaktadır. SPPM modülü birleştirme işlemi yerine toplama işlemini kullanması orta ve çıkış katmanlarını azaltmasından dolayı gerçek zamanlı modeller için uygundur. Modelin temelinde, evrişimli sinir ağı tabanlı omurga modeli bulunur. Bu omurga modeli, giriş görüntüsünden özellik haritalarını çıkarmak için kullanılır. PP-LiteSeg'de, genellikle hafif ve hızlı çalışan omurga modeli olarak MobileNetV2 tercih edilir. Birden çok aşamalı algılayıcı modülü kullanır. PP-LiteSeg'de aşamalı algılayıcı modülleri, modelin sırasıyla geliştirilmesi ve eğitilmesi ile elde edilir. PP-LiteSeg, hafif bir yapıya sahip olmasına rağmen yüksek segmentasyon performansı sunar. Özellikle sınırlı kaynağa sahip cihazlar veya gerçek zamanlı uygulamalar için oldukça uygundur.

2.6. Metrikler

Modelin test edilmesi için model eğitimi tamamlandıktan sonra eğitilen ağırlıklar kullanılarak model değerlendirilir. Segmentasyon modellerinin performansını değerlendirmek için mIoU, tahmin edilen segmentasyon sonuçlarıyla gerçek etiketler arasındaki örtüşmeyi ölçmek içinse IoU kullanılmıştır. Segmentasyon, bir giriş görüntüsünü piksellerine göre farklı nesne veya bölgelere ayırmayı hedefleyen bir görsel işleme görevidir. Bu nedenle, bir segmentasyon modelinin ne kadar doğru ve hassas olduğunu değerlendirmek için etkili bir ölçüt gereklidir. İşte mIoU bu amaçla kullanılan bir ölçüdür. mIoU, her bir sınıf için IoU değerlerinin ortalamasını olarak hesaplanır. IoU'nun hesaplama denklemi Denklem 1'de gösterilmiştir.

$$IoU = \frac{\text{Gerçek} \cap \text{tahmin}}{\text{Gerçek} \cup \text{tahmin}} \quad (1)$$

Bu denklemde tahmin, modelin tahmin ettiği piksellerin kümesini, gerçek ise gerçek etiketlere karşılık gelen piksellerin kümesini temsil etmektedir. mIoU, sınıf başına IoU değerlerinin ortalamasını alarak hesaplandığından, her sınıfın katkısı dengelenir. Dolayısıyla, modellerin çoklu sınıflarla çalıştığı senaryolarda da adil bir performans ölçütü sağlar. mIoU değeri genellikle %0 ile %100 arasındadır. Yüksek bir mIoU değeri, modelin segmentasyon görevini daha iyi yerine getirdiğini gösterirken, düşük bir mIoU değeri doğruluk düşüklüğünü veya yanlış tahminleri gösterebilir. Model geliştirme sürecinde mIoU değeri, modelin performansını izlemek, hiper parametre ayarlamalarını yapmak ve farklı modelleri karşılaştırmak için kullanılabilir. Çıkarım hızı, bir segmentasyon modelinin giriş görüntüsünü işlemek ve tahminleri üretmek için geçen süreyi ifade eder. Bu hız, modelin ne kadar hızlı çalıştığını ve gerçek zamanlı uygulamalarda kullanılabilirliğini belirler. Segmentasyon modelleri, genellikle derin öğrenme ağı tabanlıdır ve genellikle GPU'lar veya özel işlem birimleri gibi yüksek hesaplama gücü gerektiren donanımlar üzerinde çalışır. Bu nedenle, bir segmentasyon modelinin çıkarım hızı, donanımın özelliklerine, modelin karmaşıklığına ve boyutuna bağlı olarak değişebilir [23]. Bu çalışmada alınan bulgular aynı donanım ve tek bir görüntü üzerinden alınması nedeniyle aşağıdaki sonuçlar bölümünde farklı modellerin kıyaslamaları yapılmıştır.

Semantik segmentasyonun doğruluğunu değerlendirmek için kappa katsayısı istatistiksel ölçü olarak kullanılmıştır. Denklem 2'de gösterildiği gibi bu katsayı, sınıflandırıcılar arasındaki uyumu ölçer ve rastgele ne kadar farklılık olduğunu belirler. Kappa katsayısı, -1 ile 1 arasında bir değere sahiptir. Şayet değer -1 ise tamamen zıt sonuçları, değer 0 ise rastgele sonuçları, değer 1 ise mükemmel uyumu gösterir [24].

$$Kappa\ Katsayısı = \frac{Gözlemlenen\ Doğruluk - Beklenen\ Doğruluk}{1 - Beklenen\ Doğruluk} \quad (2)$$

Bu çalışmada semantik segmentasyonun doğruluğunu ölçmek için, referans etiket ve gerçek etiketlerle birlikte görüntü veri setine ihtiyaç vardır. Doğru sınıflandırılan veri noktaları, modelin tahminlerinin gerçek etiketlerle aynı olduğu veri noktalarını ifade eder. Modelin kesinliğini tespit etmek amacıyla, Denklem 3'te gösterildiği gibi AP (ortalama kesinlik) ve mAP (ortalama kesinlik değerlerinin ortalaması) metrikleri kullanılmıştır [25].

$$Doğruluk = \frac{Doğru\ Sınıflandırılan\ Veri\ Noktalarının\ sayısı}{Toplam\ Veri\ Noktalarının\ sayısı} \quad (3)$$

Denklem 3'te doğruluk değeri 0 ile 1 arasında olup, genellikle yüzde olarak ifade edilir. Yüksek bir doğruluk değeri, modelin doğru tahminlerde bulunma yeteneğini gösterirken, düşük bir doğruluk değeri yanlış sınıflandırmaların olduğunu işaret eder [22].

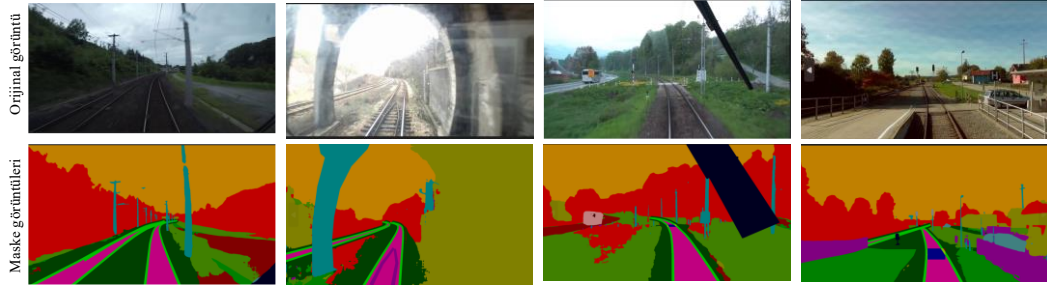
3. Bulgular

Geliştirme ortamı olarak Ubuntu 20.04 LTS işletim sistemi üzerine Python 3 versiyonu kurulmuştur. Grafik kartı olarak Nvidia GeForce RTX 4090, merkezi işlemci olarak Ryzen 9 7900X ve 32 GB RAM'e sahip bir kişisel bilgisayar kullanılmıştır. Önerilen bölütleme yöntemlerinin doğrulanması amacıyla farklı ışık ve gün koşullarında alınan bir kıyaslama veri seti olan Railsem19 veri seti kullanılmıştır. Bu veri seti tren ve şehir içi tramvayın önünden çekilmiş toplam 8500 görüntüden oluşmaktadır. Veri seti semantik bölütleme için etiketlenmiş 21 sınıfa sahip olup ray, kaldırım, trafik levhası, gökyüzü, insan, araba, kamyon gibi nesnelere içermektedir. Tablo 1'de veri setinden her bir etiket için veri setindeki örnek sayısı verilmiştir.

Tablo 1. Railsem19 için sınıf başına etiket sayısı

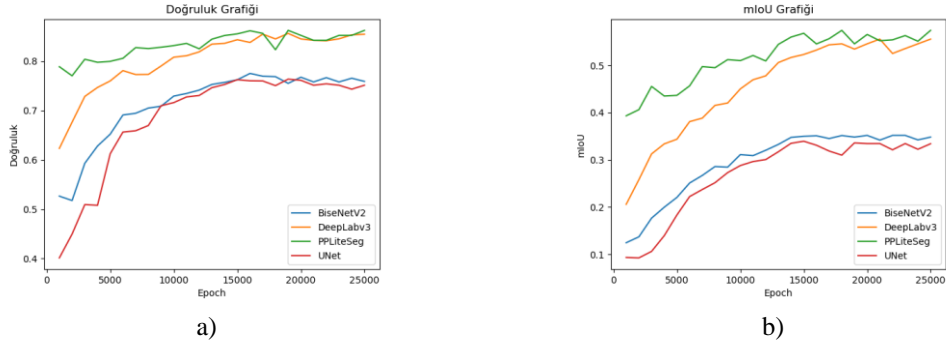
Etiket	Sayı	Etiket	Sayı
Dur Levhası	218	Statik Anahtar	2519
Geçit	2885	Ray İşaret-Ön	7404
Korkuluk	4873	Ray Geri Sinyal	3448
Vagon	1951	Ray İleri Sinyal	5768
Platform	2482	İnsan Grubu	62
Ray	58322	Araba	172
Anahtar Tanımlayıcı	2007	Çit	291
Sol Anahtar	1975	İnsan	234
Sağ Anahtar	2083	Direk	10875
Tanımsız Anahtar	2491	Ray Engeli	3920
Kamyon	11		

Tablo 1’de verilen etiketlere göre özellikle ray çiftlerinin belirlenmesi için yeterli bilginin olduğu görülmektedir. Görüntüler farklı hava koşullarında toplanmıştır. Şekil 6’da veri setinden bazı ray görüntüleri ve maskeleri verilmiştir.

**Şekil 6.** RailSem19 veri seti ve maske görüntüleri

Çalışmada demiryolu raylarının segmentasyonunda birden fazla model incelenmiş ve her bir model üzerinde eğitim gerçekleştirilmiştir. Derin öğrenme yöntemi ile semantik segmentasyon alanında sıkça kullanılan UNet, DeepLabV3 gibi modellerin yanı sıra gerçek zamanlı olarak çalışan BiSeNetV2 ve PP-LiteSeg modelleri test edilmiştir.

Gerçekleştirilen eğitimler belirlenen bulgu parametreleri ile karşılaştırılarak çıktılar elde edilmiştir. Semantik bölütleme modellerinin doğruluk metriği ve mIoU metriklerine göre karşılaştırma sonuçları Şekil 7’de verilmiştir.



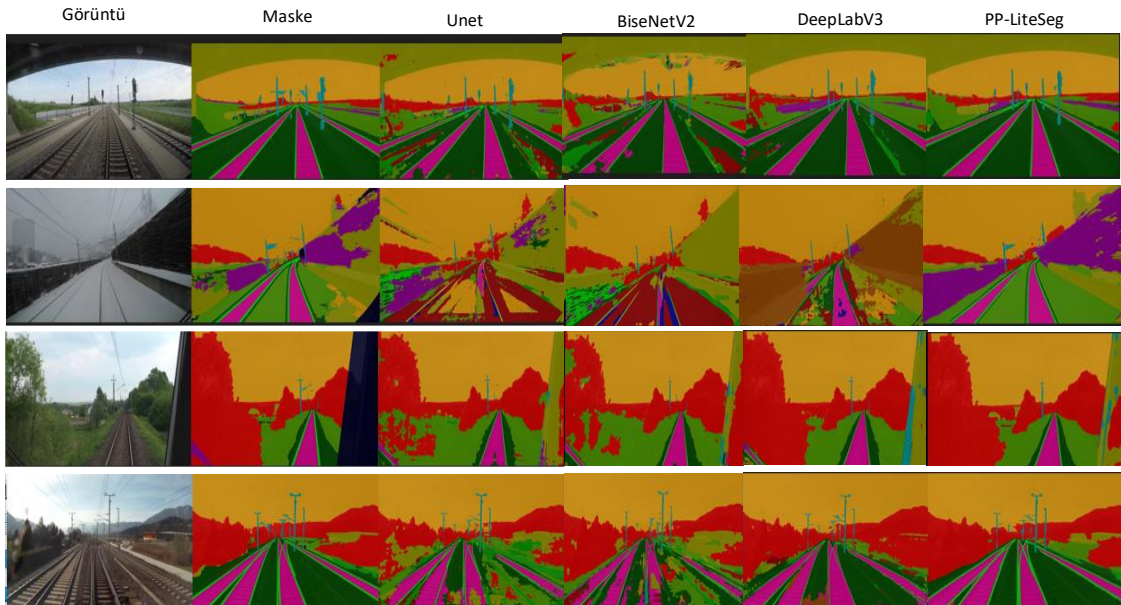
Şekil 7. Derin öğrenme tabanlı bölütleme modellerinin performansı a) Doğruluk grafiği b) mIoU grafiği

Şekil 7’de görüldüğü gibi PP-LiteSeg iki metrikte de diğer modellerden daha iyi sonuçlar vermiştir. Şekil 7’de DeepLabV3, PP-LiteSeg modelinden sonra en iyi sonucu veren model olarak göze çarpmaktadır. PP-LiteSeg’in çıkarım hızının diğerlerinden daha iyi olması, semantik segmentasyon modeli olarak öne çıkmasına sebep olmuştur. Çalışmayı gerçekleştirirken yapılan deneylerin yanında Railsem19 [20] veri seti çalışmasında önerilen yöntemin sonuçları ile karşılaştırılmıştır. Bu metriklerin karşılaştırması Tablo 2’de verilmektedir.

Tablo 2. Semantik bölütleme modellerinin performans karşılaştırılması.

Metrik-Model	PP-LiteSeg	DeepLabV3	UNet	BiSeNetV2	Railsem19 [20]
Çıkarım (ms)	3,82	11,23	6,72	8,76	-
Doğruluk (%)	96,8	96,1	74,38	74,87	91,24
mIoU (%)	58,8	57,9	32,67	31,94	56,9
Kappa (%)	76,7	74,8	55,3	55,2	-

Tablo 2’de görüldüğü gibi UNet, DeepLabV3, BiSeNetV2 yanında PP-LiteSeg modeli veri setini eğitmek için kullanılmıştır. PP-LiteSeg modelinin diğer modellere göre hem çıkarım hızı hem de performans metriklerinde daha iyi sonuç verdiğini göstermektedir. Şekil 8’de dört farklı modelin aynı görüntü üzerindeki bölütleme sonuçları verilmiştir.



Şekil 8. Dört farklı modelin bölütleme sonuçları

Şekil 8’de görüldüğü üzere farklı iklim koşullarında PP-LiteSeg modeli sahneyi daha doğru bölütlemetedir. PP-LiteSeg modeline en yakın sonuçları DeepLabV3 modeli vermiştir. Kullanılan konvolüsyon filtresi ve devamında uygulanan algoritma ile rayların eğrisinin çıkarılması sağlanmış ve sağ-sol ray arasındaki bölge kolaylıkla tespit edilebilmiştir. Şekil 8’de görüldüğü üzere sadece tek rayın olduğu durumlarda değil, rayların birleştiği veya birden çok rayın aynı anda bulunduğu durumlarda dahi rahatlıkla doğru sonuca ulaşılacağı görülmektedir. YOLOv7’nin RTX 4090 ekran kartı kullanılarak yapılan deneylerde sadece 3 milisaniye (ms) çıkarım süresine sahip olması, hızlı ve efektif bir çıkarımın yapılabildiğinin ispatıdır. Modele ait görsel sonuçlar Şekil 9’da görülmektedir.

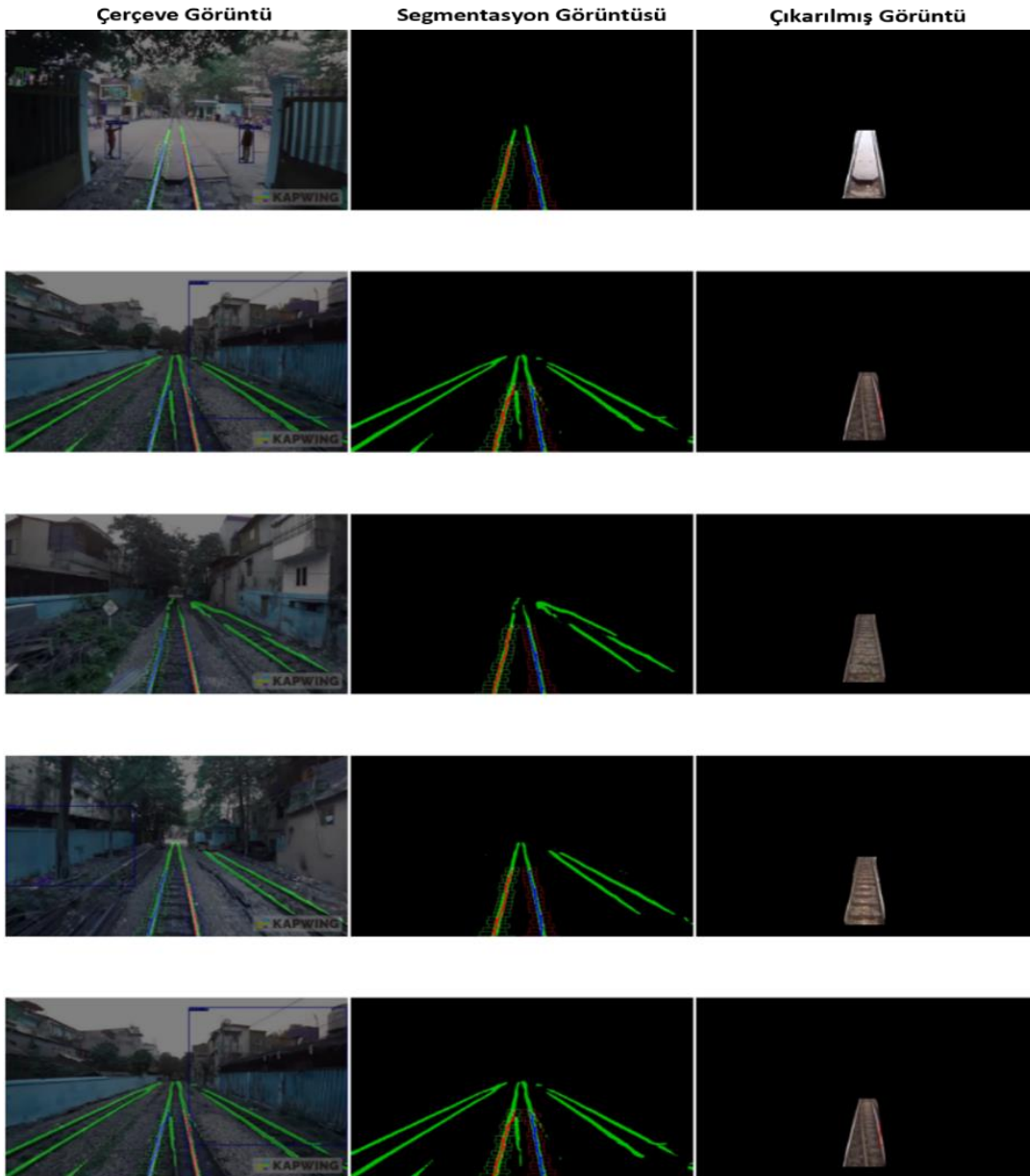


a) Bölütlenmiş ray ve ilgili alanın tespiti

b) Ray ve çevresindeki nesnelerin tespiti

Şekil 9. İlgili ray bölgesinin tespiti ve nesne tespiti

Şekil 9’da ilgili ray bölgesi belirlenmiş ve ray kenarında bulunan nesneler YOLOv7 modeli ile tespit edilmiştir. Şekil 10’da aktif rayın çıkarımı farklı görüntüler üzerinden verilmiştir.



Şekil 10. Aktif ray bölgesinin çıkarımı

4. Sonuç

Bu makalede, atmosferik koşullar, coğrafik farklılıklar, doğal felaketler ve kullanım süresi gibi demiryolu rayları üzerinde meydana gelebilecek olası değişimlerin kontrolünde kolaylık sağlayacak nesne tespitinin yapılabilmesini sağlayacak algoritma modelleri çalışılmıştır. Farklı veri setlerinin hibrit kullanım ile probleme çözüm olabileceği de önerilmiştir. Farklı nesnelere içeren 8500'in üzerindeki görüntü, görüntü artırım işlemine tabi tutulmuştur. Eğitim işleminde 8500 adet ray içeren görüntü kullanarak eğitim tamamlanmıştır. Doğrulama işlemi için 250 adet görüntü, test işlemi için 100 adet görüntü kullanılmıştır. Toplam 21 sınıf üzerinde ortalama %58,8 mIoU oranı elde edilmiş, demiryolu raylarında ise %88,7 IoU oranına ulaşılmıştır. Deneyler sonucunda PP-LiteSeg modelinin yüksek segmentasyon performansı gösterdiği ve sınırlı kaynağa sahip ortamlarda bile yüksek kalitede nesne tespiti yapılabilirliğini kanıtladığı görülmüştür. Ayrıca, YOLOv7 modelinin optimize edilerek paralel çalışabilir hale getirildiği ve özel bir kontrol mekanizması ile hafıza alanlarının bağımsız olarak kullanılabilir hale getirildiği belirtilmiştir.

Kullanılan görüntü işleme ve segmentasyon modellerinin değerlendirilmesi sonucunda, bu çalışmanın güçlü yönleri şunlarla öne çıkmaktadır: yüksek performans, sınırlı kaynaklarla kullanılabilirlik, optimizasyon ve kontrol mekanizması. Bu çalışma, nesne tespitinde topluluk modelleme ve girdi verisinin boyutunu artırmanın başarı oranını etkilediğini göstermiştir. Önerilen model farklı ortam koşullarında test edilmiştir. Ray görüntülerini başarılı bir şekilde tespit eden model bu etkiyi doğrulamıştır. Sistemin demiryolu rayları üzerinde rahatlıkla hata ve anomali tespiti yapabileceği kanıtlanmıştır.

Teşekkür

Bu çalışma, Fırat Üniversitesi Bilimsel Araştırma Projeleri Birimi tarafından ADEP.22.02 nolu proje ile desteklenmiştir.

Kaynakça

- [1] T. Zhu, & JMMS. De Pedro, "Railway traffic conflict detection via a state transition prediction approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 5, pp. 1268-1278.
- [2] T. Ye, Z. Zheng, X. Li, Z. Zhao, & XZ. Gao, "An efficient few-shot object detection method for railway intrusion via fine-tune approach and contrastive learning," *IEEE Transactions on Instrumentation and Measurement*.
- [3] Y. Li, Y. Qin, Z. Xie, Z. Cao, L. Jia, Z. Yu & E. Zhang, "Efficient SSD: a real-time intrusion object detection algorithm for railway surveillance," In 2020 International Conference on Sensing, Diagnostics, Prognostics, and Control (SDPC), pp. 391-395. IEEE
- [4] Z. Zheng, W. Liu, R. Liu, L. Wang, L. Mao, Q. Qiu, & G. Ling, "Anomaly detection of metro station tracks based on sequential updatable anomaly detection framework," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 11, pp. 7677-7691
- [5] X. Ding, X. Cai, Z. Zhang, W. Liu, & W. Song, "Railway foreign object intrusion detection based on deep learning," In 2022 International Conference on Computer Engineering and Artificial Intelligence (ICCEAI), pp. 735-739. IEEE.
- [6] SS. Kırat & İ. Aydın, "Açıklanabilir Yapay Zekâ Tabanlı Denetimsiz Öğrenme ile Ray Kusur Tespiti", *Demiryolu Mühendisliği*, vol. 18, pp. 1-13. doi:10.47072/demiryolu.1231751.
- [7] M. Sevi, İ. Aydın, E. Akın, "Detection of rail surface defects based on ensemble learning of YOLOv5," *Demiryolu Mühendisliği*, vol. 17, pp. 115-132. doi: 10.47072/demiryolu.1205483.
- [8] D. Çetintaş & T. Tuncer, "Determining the type of document read using eye movement properties by hybrid CNN method," *Traitement du Signal*, vol. 39, no. 4, pp. 1099.
- [9] X. Gong, X. Chen, Z. Zhong & W. Chen, "Enhanced few-shot learning for intrusion detection in railway video surveillance," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 8, pp. 11301-11313.
- [10] X. Li, L. Zhu, Z. Yu, B. Guo, & Y. Wan, "Vanishing point detection and rail segmentation based on deep multi-task learning," *IEEE Access*, vol. 8, pp. 163015-163025.
- [11] X. Ding, X. Cai, Z. Zhang, W. Liu, & W. Song, "Railway foreign object intrusion detection based on deep learning," In 2022 International Conference on Computer Engineering and Artificial Intelligence (ICCEAI), pp. 735-739. IEEE.
- [12] H. Huang, G. Zhao, Y. Bo, J. Yu, L. Liang, Y. Yang, & K. Ou, "Railway intrusion detection based on refined spatial and temporal features for UAV surveillance scene," *Measurement*, vol. 211, 112602.
- [13] Z. Cao, Y. Qin, Z. Xie, Q. Liu, E. Zhang, Z. Wu & Z. Yu, "An effective railway intrusion detection method using dynamic intrusion region and lightweight neural network," *Measurement*, vol. 191, 110564.
- [14] M. Sevi, İ. Aydın, "Improving Unet segmentation performance using an ensemble model in images containing railway lines," *Turk J Elec Eng & Comp Sci*, vol. 34, no. 4, pp. 739-750.

- [15] O. Ronneberger, P. Fischer, & T. Brox, “U-net: convolutional networks for biomedical image segmentation,” *In Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference*, Munich, Germany, 2015, Proceedings, Part III 18, pp. 234-241.
- [16] LC. Chen, G. Papandreou, F. Schroff & H. Adam, “Rethinking atrous convolution for semantic image segmentation,” *arXiv preprint arXiv:1706.05587*.
- [17] C. Yu, C. Gao, J. Wang, G. Yu, C. Shen & N. Sang, “Bisenet v2: Bilateral network with guided aggregation for real-time semantic segmentation,” *International Journal of Computer Vision*, vol. 129, pp. 3051-3068.
- [18] J. Peng, Y. Liu, S. Tang, Y. Hao, L. Chu, G. Chen, & Y. Ma, “Pp-liteseg: A superior real-time semantic segmentation model,” *arXiv preprint arXiv:2204.02681*.
- [19] CY. Wang, A. Bochkovskiy & HYM. Liao, “YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors,” *arXiv 2022. arXiv preprint arXiv:2207.02696*.
- [20] O. Zendel, M. Murschitz, M. Zeilinger, D. Steininger, S. Abbasi, S & C. Beleznai, “Railsem19: a dataset for semantic rail scene understanding,” *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 0-0.
- [21] J. Redmon, S. Divvala, R. Girshick & A. Farhadi, “You only look once: Unified, real-time object detection,” *In Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 779-788.
- [22] P. Li, H. Xiong, J. Fan, “Sun dfanet: Deep feature aggregation for real-time semantic segmentation,” *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition 2019*, pp. 9522-9531
- [23] E. Romera, JM. Alvarez, LM. Bergasa, R. Arroyo, “Erfnet: efficient residual factorized convnet for real-time semantic segmentation,” *IEEE Transactions on Intelligent Transportation Systems 2017: 19*, pp. 263-272.
- [24] RP. Poudel, S. Liwicki, R. Cipolla, “Fast-scnn: fast semantic segmentation network,” *arXiv preprint arXiv:1902.04502 2019*.
- [25] H. Zhao, X. Qi, X. Shen, J. Shi & J. Jia, “Icnet for real-time semantic segmentation on high-resolution images,” *In: Proc. European Conference on Computer Vision (ECCV) 2018*, pp. 405– 420.

Özgeçmiş



İlhan AYDIN

1981 yılında Elâzığ’da doğdu. Lisans ve Yüksek Lisans derecelerini Fırat Üniversitesi Bilgisayar Mühendisliğinde 2001 ve 2006 yıllarında aldı. Doktora derecesini 2011 yılında Fırat Üniversitesi, Elâzığ Elektrik ve Elektronik Mühendisliği bölümünden aldı. Halen Fırat Üniversitesi’nde Bilgisayar Mühendisliği bölümünde Profesör Doktor olarak görev yapmaktadır. Araştırma alanları arasında optimizasyon, gerçek zamanlı sistemler, hata teşhisi ve durum izleme, sinyal ve görüntü işleme ve derin öğrenme bulunmaktadır.

E-Posta: iaydin@firat.edu.tr



Taha Kubilay ŞENER

1999 Elâzığ doğumludur. İlk, orta ve lise öğrenimini Elazığ’da tamamladı. 2018 yılında başladığı Fırat Üniversitesi Bilgisayar Mühendisliği halen eğitimine devam etmektedir. Çalışma alanları yapay zekâ, görüntü işleme ve derin öğrenme alanlarıdır.

E-Posta: tksener23@gmail.com

**Mehmet SEVİ**

1988 yılında Denizli’de doğdu. Lisans derecesini Doğu Akdeniz Üniversitesi Bilgisayar Mühendisliğinde (Burslu, İngilizce), Yüksek Lisans derecesini ise Fırat Üniversitesi Bilgisayar Mühendisliği Bölümünde aldı. Şu an Fırat Üniversitesi Bilgisayar Mühendisliği Bölümünde doktora öğrencisidir. Araştırma alanları arasında görüntü işleme ve derin öğrenme bulunmaktadır. Aynı zamanda Muş Alparslan Üniversitesi Bilgi İşlem Daire Başkanlığında mühendis olarak çalışmaktadır.

E-Posta: m.sevi@alparslan.edu.tr

Beyanlar:

Bu makalede bilimsel araştırma ve yayın etiğine uyulmuştur.

Yazarların katkıları: İlhan AYDIN: Kavramsallaştırma, Metodoloji, Yazılım. Taha Kubilay ŞENER: Kaynaklar, Doğrulama, Yazılım. Mehmet SEVİ: Görselleştirme, İnceleme, Yazma-gözden geçirme ve düzenleme.