

Uydu verileriyle nesne tespiti:

Faster R-CNN

Amaç

Denizlerde gemi trafiğini izlemek; güvenlik, çevresel izleme ve arama-kurtarma gibi birçok alanda kritik bir ihtiyaçtır. Geleneksel yöntemler genellikle hava koşullarına bağlı olarak değişiklik gösterirken, SAR görüntüleri gece-gündüz ve kötü hava şartlarında bile güvenilir veri sağlayabilir. Bu nedenle, derin öğrenme tabanlı bir yaklaşım kullanarak bu süreci daha hızlı ve doğru hale getirmek hedeflenmiştir.

Yapılan Çalışmalar

Birçok çalışmada, CNN ve RNN gibi derin öğrenme algoritmaları kullanılarak uydu ve İHA verileriyle nesne tespiti gerçekleştirilmiş ve yüksek doğruluk oranları elde edilmiştir. Zhang (2019), okyanus bölgelerindeki gemileri tespit etmek için uydu verileri ve CNN'lerin bir kombinasyonunu kullanmış ve %87 doğruluk oranına ulaşmıştır. Ancak, uydu verileriyle yapılan eğitimlerde hâlâ karmaşıklıklar bulunmaktadır. Daha fazla çalışma ile bu yöntemler tam potansiyeline ulaşabilir.

Teknolojiler

Faster R-CNN

Faster R-CNN, iki ana bileşenden oluşur: bir **Bölge Öneri Ağı (RPN)** ve bir **Algılama Ağı**.

- **Bölge Öneri Ağı (RPN):** Giriş görüntülerine dayalı olarak nesne önerileri üretir.
- **Algılama Ağı:** RPN tarafından üretilen önerileri kullanarak nesne sınıflandırması ve sınırlayıcı kutu regresyonu gerçekleştirir.

Faster R-CNN, 2015 yılından bu yana sürekli olarak geliştirilmektedir. Çok ölçekli haritaları birleştirme ve öz dikkat mekanizması gibi özellikler eklenmiştir. Gemi tespiti, trafikte yaya tespiti gibi birçok alanda başarıyla kullanılmaktadır.

SAR

Deniz gözetimi ve gemi takibi gibi alanlarda sunduğu potansiyel nedeniyle, SAR (Sentinel-1 Radar Görüntüleme Sistemi) aktif bir araştırma konusu haline gelmiştir.

Deniz yüzeyinin yüksek çözünürlüklü görüntülerini sağlayabildiğinden, optik lenslere bir alternatif sunmaktadır. Gemi tespiti için birçok yaklaşım öne sürülmüştür. Örneğin, istatistiksel modelleme ve morfolojik işlemlere dayalı yöntemler geliştirilmiştir.

SAR'ın sunduğu yüksek potansiyele rağmen, deniz yüzeyindeki karmaşıklık ve gemi özelliklerindeki değişkenlik gibi zorluklar devam etmektedir.

Çalışma

Yapılan çalışma, konumu ve stratejik önemi nedeniyle **Mersin Limanı**'nda gerçekleştirilmiştir.

- **Sentinel-1** uydusundan gelen SAR verileri kullanılmıştır.
- **VH polarizasyonu**, karmaşık şekillere sahip küçük gemileri ve deniz yüzeyi üzerindeki rüzgar yönü gibi unsurları belirlemeye yardımcı olabilir.
- SAR verileri; mahsul izleme, afet müdahalesi gibi birçok farklı alanda kullanılmaktadır. Bu yönü, onu geniş yelpazeli uygulamalar için önemli kılmaktadır.

SARfish

SARfish, **SAR görüntüleri ve Faster R-CNN** kullanarak gemileri tespit eden açık kaynaklı bir sistemdir.

- SARfish, gürültü temizleme ve özellik çıkarma yöntemleri ile **%92 doğruluk oranına** ulaşarak gemi tespiti yapar.
- Gemi hareketlerini izleme, çevresel izleme ve gemi güvenliği gibi alanlarda kullanılır.
- **Faster R-CNN**, Bölge Öneri Ağı (**RPN**) kullanarak gemi içeren bölgeleri belirler ve bu bölgeleri sınıflandırarak daha yüksek doğruluk oranı sunar.
- **Açık kaynaklı** olması sayesinde geliştirilebilir ve özelleştirilebilir bir yapıya sahiptir.

Bu özellikleri sayesinde **deniz güvenliği, gemi trafiği yönetimi ve çevresel izleme** gibi alanlarda büyük potansiyel taşımaktadır.

Bulgular

Faster R-CNN algoritmasına **Sentinel-1**'den elde edilen SAR verileri uygulanarak bölgedeki gemiler tespit edilmiştir.

- SAR görüntülerindeki gemileri tanımlamak için **Konvolüsyonel Sinir Ağları (CNN)** ve **Bölge Öneri Ağları (RPN)** kombinasyonu kullanılmıştır.
- **Farklı türdeki ve farklı boyutlardaki gemiler %86 doğruluk oranı** ile tespit edilmiştir.
- Elde edilen sonuçlar, kullanılan algoritmanın **gerçek zamanlı gemi algılama ve gözetleme sistemlerine uygulanma potansiyeline sahip olduğunu** göstermektedir.

Sonuçlar

Geleneksel yöntemlerin sınırlamalarını aşarak **yüksek doğruluk** ve **düşük hata oranı** sağlamaktadır.

- Algoritma, **farklı gemi türlerini başarıyla tespit edebilir** ve denizcilik uygulamalarına uygundur.
- Çalışma, **SAR görüntüleri ve derin öğrenmenin etkinliğini göstererek** gelecekteki araştırmalara katkı sunmaktadır.
- **Deniz güvenliği ve çevre koruma açısından gemi tespitinin önemini** vurgulamaktadır.

Mask R-CNN

R-CNN yönteminde, görüntüde **nesne adayı bölgeler belirlenmekte** ve **özellik çıkarımı** konvolüsyonel katmanlar aracılığıyla gerçekleştirilmektedir.

Bu tür tespit yöntemlerinden biri olan **Mask R-CNN**'de, nesne konumlarının belirlenmesi **öğrenme yoluyla** gerçekleştirilmektedir.

Veri ve Çalışma Kapsamı

Çalışmada, **Google Earth**'ten alınan **1 m çözünürlüklü RGB uydu görüntüleri** kullanılmıştır.

- **Toplamda 1838 görüntü** üzerinde çalışılmış ve **3279 gemi sayısallaştırılmıştır**.
- Model, **eğitim ve test setleriyle değerlendirilmiştir**.

Verilerin İşlenmesi

Veri işleme süreci şu aşamalardan oluşmaktadır:

1. Girdi verisi
2. Özellik çıkarımı
3. Bölge Öneri Ağı (RPN)
4. Önerilerin işlenmesi
5. R-CNN uygulaması
6. Çıktı üretimi

Mask R-CNN, dört modülden oluşmaktadır:

- Özellik çıkarım ağı
- Bölge öneri ağı (RPN)
- Öneri işleme modülü
- R-CNN

Girdi verileri, **Yükseklik × Genişlik × Derinlik** tensörü formatında olup, bu görüntülerden **konvolüsyonel haritalar** elde edilmektedir.

FPN

FPN, özellik çıkarım ağındaki **yüksek çözünürlüklü ancak sığ semantik değere sahip katman özelliklerini, derin semantik özelliklerle** kaynaştırmak amacıyla geliştirilmiş **yanal bağlantılara** sahip bir mimaridir.

Bu yapı, **bölge tabanlı CNN sistemlerine uyumlu** olarak tasarlanmıştır.

Özellik Piramidi

Özellik piramit ağı, iki aşamadan oluşmaktadır:

1. **Aşağıdan yukarıya yol:** İlk aşamada, **beş bloktan oluşan** bir süreç ile özellik haritaları oluşturulur.
2. **Yukarıdan aşağıya yol:** İkinci aşamada, özellik haritaları **ölçeklenir ve kaynaştırılır**.

Bu işlem, **Bölge Öneri Ağı (RPN)** içinde **nesne adaylarını belirlemek** için kullanılır.

- **Mask R-CNN, P2, P3, P4 ve P5 ölçeklerindeki haritaları RPN modülüne** girdi olarak vererek **nesne tahmini ve konum düzeltme işlemlerini** gerçekleştirir.
- **Rol (İlgi Alanı) seçimi**, nesne boyutuna göre en uygun özellik haritası katmanından yapılmaktadır.

Bu seçim, aşağıdaki denkleme göre belirlenmektedir:

$$k = 4 + \log_2 \left(\frac{\sqrt{wh}}{224} \right)$$

Bu denklemin çözümü sonucunda elde edilen k değerine bağlı olarak hangi katmanın kullanılacağı belirlenmektedir. Bu sayede nesnenin büyüklüğüne göre uygun özellikteki harita kullanılmaktadır. Ayrıca üretilen önerilerin nesne olup olmadığına dair bir güven eşiği tanımlıdır. Öneri için üretilen olasılık bu eşiğin altında olduğu takdirde öneri “arka plan” olarak kabul edilmekte, üstünde olduğundada gemi olarak adlandırılmakta.

Rol Hizalaması, RPN tarafından önerilen nesne bölgelerinin kesilip hizalanarak belirli bir boyuta getirilmesini sağlar. Bilineer enterpolasyon ve max-pooling kullanılarak, bu bölgeler tam bağlantılı katmanlara uygun hale getirilir.

Mask R-CNN’nin son aşamasında, R-CNN modülü nesneleri gemi sınıfına atar, konumlarını düzenler ve maskeler oluşturur. Dört aşamada, (1) özellik haritaları çıkarılır, (2) RPN ile nesne önerileri oluşturulur, (3) Rol hizalaması ile düzenlenir, (4) nesneler konumları ve maskeleriyle tespit edilir.

Model Eğitimi:

TensorFlow ve Keras derin öğrenme çerçevelerinde çalışan GitHub uyarlaması kullanılmıştır. Modelde omurga ağ olarak, Common Objects in Context (COCO) veri setiyle ön eğitim almış ResNet-101 kullanılmıştır. 80 bin iterasyon yapılmış ve Stokastik Gradyan İniş yöntemi ile geriyayılım algoritmaları kullanılmıştır.

Modelin tespit performansını değerlendirmek için yaygın olarak kullanılan kesinlik, geri getirme ve F1 skoru metrikleri kullanılmıştır.

Bulgular:

Güven eşiğinin en doğru oranının 0.5 olduğu tespit edilmiştir. Test aşamasında 604 veri üzerinden %91 doğruluk elde edilmiştir. Birbirine yakın gemilerin yanlış tespit edilmesi, Faster R-CNN ve Mask R-CNN’de dik sınırlayıcı kutuların kullanımından kaynaklanmaktadır. Çözüm olarak dönük sınırlayıcı kutular önerilmektedir.

Sonuçlar:

Bu çalışmada, Mask R-CNN ile uydu görüntülerindeki gemiler tespit edilmiş ve performansı değerlendirilmiştir. Model, 604 gemiden 558'ini doğru tespit ederken 58 yanlış alarm vermiştir. Birbirine yakın gemiler bazen tek bir gemi olarak algılanmış ya da gözden kaçmıştır. Ayrıca, bazı kara bölgeleri yanlışlıkla gemi olarak tanımlanmış, bazı kıyıdaki gemiler ise atlanmıştır. Gelecek çalışmalarda bu hataların nedenleri araştırılacak ve farklı modellerle karşılaştırmalar yapılacaktır.

YOLO Algoritması:

YOLO, nesne algılama görevini hesaplama karmaşıklığını azaltan tek bir regresyon problemi olarak ele almaktadır. Omurga, boyun ve baş olmak üzere üç parçadan oluşmaktadır:

- **Omurga:** Öznitelik çıkarımı
- **Boyun:** Birleştirme ve seçme
- **Baş:** Tahminleme

YOLOv8 Algoritması:

YOLOv8, C2f modülü sayesinde gradyan akışını optimize ederek öğrenme sürecini hızlandırır ve daha hızlı, verimli nesne tespiti sağlar.

YOLOv9 Algoritması:

YOLOv9, özellik çıkarımı için CSPDarknet53'ü, özellik birleştirme için SPP ve PAN modüllerini kullanarak daha verimli nesne tespiti sağlar. RepNCSPPELAN4 modülü, YOLOv5, YOLOv6 ve YOLOv7'den alınan bileşenleri birleştirerek modelin öğrenme kapasitesini ve doğruluğunu artırır.

Veri Seti:

1658 uydu görüntüsünden oluşan veri seti, farklı hava koşulları, ışık seviyeleri ve gemi türlerini içermektedir.

Veri seti, %86 eğitim (1420), %10 doğrulama (159) ve %5 test (79) görüntüsü olarak üçe ayrılmıştır. Görüntüler, 640x640 piksele yeniden boyutlandırılmış ve yönlendirme işlemi uygulanmıştır.

Veri artırma sürecinde, her görüntü yatay çevrilmiş ve 90° döndürülerek üç farklı versiyon üretilmiştir. Tüm gemiler, YOLOv5 modeli için sınırlayıcı kutularla etiketlenmiştir.

Bulgular:

En uygun model, 25 iterasyon ile eğitilmiştir. YOLOv8 ve YOLOv9'un karşılaştırılmasında iterasyon arttıkça hata oranlarının azaldığı gözlemlenmiştir. YOLOv8'in daha stabil olduğu, YOLOv9'un ise daha hızlı yakınsadığı tespit edilmiştir. Zamanla hata oranı azalsa da dalgalanmalar gözlemlenmiştir. YOLOv9'un doğrulama setinde daha iyi genelleme yaptığı belirlenmiştir. YOLOv9 başlangıçta daha iyi değerler alırken, YOLOv8 zamanla aynı seviyeye ulaşmaktadır. YOLOv9, küçük ve kısmen gizli nesneleri tespitte daha başarılıdır ve özellikle başlangıç iterasyonlarında daha yüksek mAP değerlerine ulaşmaktadır. Genel olarak, YOLOv9'un performansı daha yüksek bulunmuştur.

Sonuçlar:

YOLOv9'un performansı ve verimliliği, onu gemi tespiti uygulamaları için özellikle cazip bir seçenek haline getirirken, YOLOv8 daha geniş bir uygulama yelpazesi sunmaktadır. Bu nedenle seçim, belirli bir uygulamanın gereksinimlerine ve mevcut kaynaklara bağlı olacaktır.