**GELİŞMİŞ DENİZ GÖZLEMİ : SAR TABALI GEMİ TESPİTİ İÇİN CNN ALGORİTMALARININ KULLANIMI**

Bu makalede, Sentinel-1 uydusundan alınan SAR (Sentetik Açıklıklı Radar) verileri kullanılarak Faster R-CNN algoritması ile gemi tespiti yapılmıştır. Çalışmada, %86.11 doğruluk oranı elde edilmiştir.

**Bu araştırmanın önemi Nedir?**

Bu araştırma, gemi tespitinin deniz güvenliği, gemi trafiği yönetimi, çevresel izleme ve arama kurtarma operasyonları gibi kritik alanlardaki rolünü vurgulamaktadır. Uydu ve radar görüntüleme teknolojilerindeki ilerlemeler ile derin öğrenme algoritmalarının birleşimi, gemi hareketlerini hızlı ve doğru bir şekilde tespit ve takip etmeyi mümkün kılarak, korsanlık, kaçakçılık ve çevresel kirlilik (petrol sızıntıları) gibi tehditlere karşı önemli çözümler sunmaktadır. Uydu verileriyle nesne tespiti, tarım, şehir planlama ve afet yönetimi gibi çeşitli alanlarda birçok potansiyel uygulama ile hızla gelişen bir araştırma alanıdır.

**Faster R-CNN Algoritması**  
Faster R-CNN, nesne tespiti için yaygın olarak kullanılan bir derin öğrenme modelidir. Bölge Öneri Ağı (RPN) ve algılama ağı olmak üzere iki ana bileşenden oluşur. Algoritma, özellikle SAR görüntülerinde gemi tespiti ve trafik sahnelerinde yaya tespiti gibi uygulamalarda başarı göstermiştir.

**SAR Görüntüleri ve Gemi Tespiti**  
SAR (Sentetik Açıklıklı Radar) görüntüleri, hava koşullarından bağımsız olarak yüksek çözünürlüklü deniz yüzeyi görüntüleri sağlayarak optik sensörlere alternatif oluşturmaktadır.

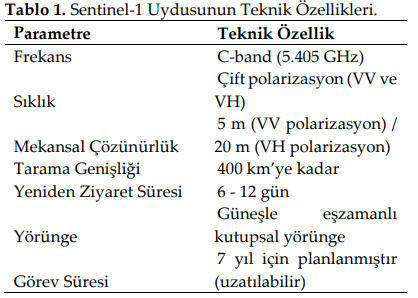
**CNN Tabanlı Gemi Tespiti ve Zorluklar**  
CNN tabanlı derin öğrenme algoritmaları, gemi tespitinde umut verici sonuçlar verse de, deniz ortamındaki değişkenlik, deniz yüzeyinin karmaşıklığı ve gemi boyutlarının farklılık göstermesi gibi zorluklar bulunmaktadır.

**Materyal ve yöntem**

**Çalışma Alanı:** Mersin Limanı

**Veri**  
Gemi tespiti için Sentinel-1 uydusunun sağladığı SAR (Sentetik Açıklıklı Radar) görüntüleri kullanılmıştır. Bu uydu, farklı polarizasyon modlarına (VV(dikey) ve VH(yatay)) sahip olup, küçük gemileri ve balıkçı teknelerini daha iyi tespit edebilmek için VH polarizasyonu kullanılmıştır.

**Polarizasyon**, radar dalgalarının yayılma ve geri yansıma özelliklerini tanımlayan bir kavramdır.



**5.405 GHz:** Sentinel-1’in kullandığı özel frekanstır ve yağmur, bulut ve atmosfer koşullarından minimum etkilenen bir SAR (Sentetik Açıklıklı Radar) sinyali sağlar.

**VV Polarizasyon:** Dikey gönderilen ve dikey geri alınan sinyaller. Büyük gemiler, deniz dalgaları, şehir yüzeylerini tespit eder.

**VH Polarizasyon:** Dikey gönderilen, yatay olarak geri alınan sinyaller. Küçük tekneler, arazi örtüsü, çevresel analiz yapmada kullanılır.

**5 metre (VV Polarizasyonu)** → Daha yüksek detay ve keskin görüntüler sunar.

**20 metre (VH Polarizasyonu)** → Daha geniş alanları kapsayan ama biraz daha düşük çözünürlüklü görüntüler sağlar.

**400 km’ye kadar** alanı tek geçişte tarayabilir.

**6 - 12 gün**, bir bölgenin tekrar görüntülenmesi için geçen süredir.

**Yöntem**

**Gemi Tespiti ve SAR Görüntüleri**  
SAR (Sentetik Açıklıklı Radar) görüntüleri, gece ve kötü hava koşullarında bile güvenilir veri sağladığı için gemi tespitinde önemli bir araçtır.

**SARfish Algoritması**  
Bu algoritma:

* SAR görüntülerinden gürültü ve benekleri temizleyerek ön işleme yapar.
* Faster R-CNN nesne algılama çerçevesini kullanarak gemileri tespit eder.
* %92.9 doğruluk oranı ile yüksek hassasiyet sağlar.
* Açık kaynaklı olup özelleştirilebilir, denizcilik araştırmalarına katkı sunar.

**Faster R-CNN Algoritması**

* Bölge Öneri Ağı (RPN) kullanarak gemi adaylarını belirler.
* İlgi Bölgesi (RoI) havuzlama katmanı, tespit edilen alanları iyileştirir.
* Sınıflandırıcı ağ, gemilerin konum ve boyutlarını hassas şekilde belirler.
* Bölge önerme ve sınıflandırma işlemlerini tek bir çerçevede birleştirerek en iyi tespit performansını sunar.

**Algoritmanın Uygulanması**

1. SAR görüntüleri toplanır (uydu sağlayıcılarından elde edilir).
2. Ön işleme yapılır (gürültü giderme ve özellikleri iyileştirme).
3. Faster R-CNN tabanlı SARfish algoritması çalıştırılır (gemi konum ve boyutları belirlenir).
4. Elde edilen veriler analiz edilerek deniz güvenliği, trafik yönetimi ve çevresel izleme için kullanılır.

**RPN Nasıl Çalışır?**

**Giriş (Input):** RPN, bir konvolüsyonel sinir ağı (CNN) tarafından çıkarılan özellik haritasını kullanır. Özellik haritası, görüntüdeki önemli detayları (nesnelerin kenarları, şekilleri vb.) içerir.

**Kenar Çapaları (Anchors) Oluşturma:** RPN, her bir özellik haritası noktasına farklı ölçeklerde ve en-boy oranlarında dikdörtgen bölgeler (anchors) yerleştirir. Örneğin, her nokta için 3 farklı ölçek ve 3 farklı oran kullanıldığında toplamda 9 anchor oluşturulur. Bu anchors sayesinde model, farklı boyuttaki nesneleri daha iyi tespit edebilir.

**Öneri Değerlendirme (Region Proposal Scoring):** RPN, her anchor için bir "Nesne mi / Arka Plan mı?" tahmini yapar. Ayrıca, nesne olduğu tahmin edilen bölgelerin konumunu hassas bir şekilde ayarlamak için regresyon işlemi uygular.

**Zayıf Bölgeleri Eleme (Non-Maximum Suppression - NMS):** Aynı nesne için birden fazla anchor önerisi varsa Non-Maximum Suppression (NMS) yöntemi ile en yüksek olasılığa sahip anchor seçilir, diğerleri elenir.

**Çıktı (Output):** RPN, belirli sayıda en iyi bölgeyi önerir (örneğin, 2000 bölge yerine en iyi 300 bölge seçilir). Bu önerilen bölgeler, nesne olup olmadığını belirlemek için sınıflandırıcı ağına (Faster R-CNN'nin ikinci aşaması) gönderilir.

**Bulgular**

Faster R-CNN tabanlı gemi tespit algoritması, Sentinel-1 VH SAR görüntülerine uygulanmış ve farklı gemi türlerini başarıyla tespit etmiştir.

Konvolüsyonel sinir ağları (CNN) ve Bölge Öneri Ağı (RPN) kombinasyonu, gemilerin konumlarını doğru bir şekilde belirleyerek bir harita üzerinde görselleştirilmiştir.

Kargo gemileri, balıkçı tekneleri ve konteyner gemileri gibi farklı boyut ve şekillerdeki gemiler yüksek doğrulukla tespit edilmiştir.

Yanlış pozitif oranı düşük olup, model görüntüdeki diğer nesneleri gemi olarak yanlış tanımlamamıştır.

Elde edilen performans sonuçları:

* Genel doğruluk: %86,11
* Kesinlik (Precision): %84,54
* Geri çağırma (Recall): %89,03

**MASK R-CNN İLE UYDU GÖRÜNTÜLERİNDE GEMİ TESPİTİ**

Çalışmada, Mask R-CNN modeli kullanılarak gemilerin tespiti gerçekleştirilmiştir. 1 metre mekânsal çözünürlüğe sahip 1838 uydu görüntüsü indirerek, içeriğindeki gemiler GIS yazılımı ile etiketlenmiş ve veri setleri oluşturulmuştur. Sonuçlar, Mask R-CNN modelinin gemileri zorlu koşullarda bile tespit etme yeteneğine sahip olduğunu, ancak birbirine yakın gemilerin tespiti konusunda zorluk yaşandığını göstermektedir. Bu durum, modelin bazı sınırlamaları olduğunu ortaya koymaktadır.

**Giriş**

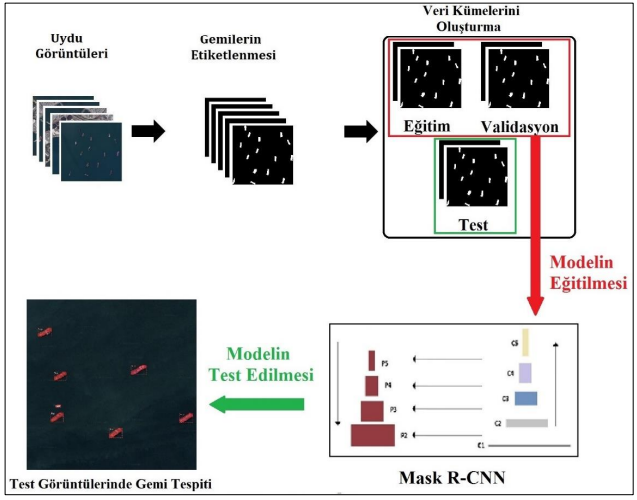
Son yıllarda, yüksek çözünürlüklü uydu görüntüleri ve yeni algılama modelleri ile gemi tespiti daha otomatik hale gelmiştir. Makine öğrenimi, nesne özelliklerini öğrenerek daha hızlı ve doğru sonuçlar üretmiş, ancak geleneksel algoritmalar (örneğin, SVM) veri sapmalarında performans kayıpları yaşanmıştır.

Konvolüsyonel sinir ağları (CNN), nesne tespitinde başarılı sonuçlar elde edilmesine yardımcı olmuştur. Ancak, nesnelerin tüm görüntüye yayılmadığı durumlarda, nesne tespiti için daha fazla uyarlama gerekmektedir. Başlangıçta, kayan pencere yöntemi ile her bir görüntü parçası üzerinde sınıflandırma yapılmıştır, ancak bu yöntem işlem zamanını arttırmıştır. Bunun yerine, CNN tabanlı iki farklı nesne tespiti yaklaşımı geliştirilmiştir:

1. **Bölge-tabanlı CNN** yaklaşımları (Faster R-CNN, Mask R-CNN), nesne aday bölgeleri belirler ve bu bölgelerde sınıflama yapar. Mask R-CNN, nesne konumlarını öğrenme yoluyla belirler ve sınırlayıcı kutular ile maskelerle işaretler.
2. **YOLO ve SSD** gibi bölge öneri ağına gerek duymayan yöntemler, doğrudan görüntü özelliklerinden nesne tespiti yapar.

Bu çalışma, Mask R-CNN modelini kullanarak gemi tespiti ve maske kullanımının tespit sonuçları üzerindeki etkilerini incelemektedir.

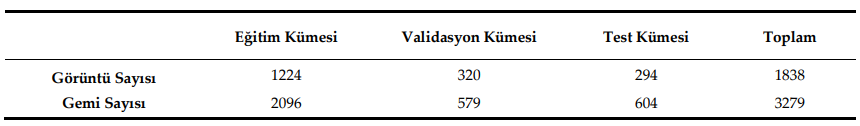
**Materyal ve Yöntem**

****

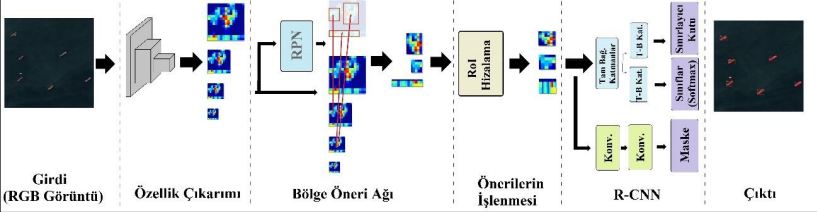
Bu görsel, uydu görüntülerinden gemi tespiti sürecini adım adım göstermektedir.

1. **Uydu Görüntüleri:** İlk olarak, uydu görüntüleri elde edilir.
2. **Gemilerin Etiketlenmesi:** Görüntülerdeki gemiler işaretlenerek etiketlenir ve veri kümesi oluşturulur.
3. **Veri Kümesi Bölünmesi:** Etiketli görüntüler eğitim, doğrulama (validasyon) ve test setlerine ayrılır.
4. **Modelin Eğitilmesi:** Mask R-CNN modeli kullanılarak model eğitilir.
5. **Test Aşaması:** Eğitilen model, test görüntülerindeki gemileri tespit eder.

**Veri Setleri:** Çalışmada kullanılan veriler Google Earth tarafından sağlanan 1 metre mekânsal çözünürlüklü, geotif formatındaki RGB uydu görüntüleridir. Açık denizlerden, kıyı yakınlarından ve iç sulardan örnekler bulunmaktadır. 768x768 piksel büyüklüğünde toplam 1838 görüntü elde edilmiştir. Görüntülerdeki toplam 3279 gemi bir GIS yazılımı kullanılarak sayısallaştırılmıştır. Oluşturulan eğitim ve validasyon setleri kullanılarak model eğitilmiş ve doğrulaması yapılmıştır.

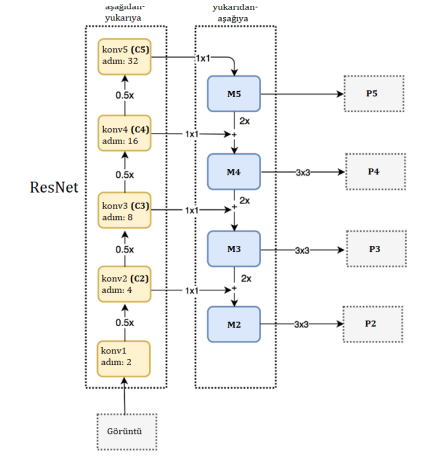
****

**Yöntem**



Mask R-CNN dört ana modülü (özellik çıkarım ağı, bölge öneri ağı, öneri işleme modülü ve bölge-tabanlı konvolüsyonel sinir ağı) :

1. **Özellik Çıkarım Ağı (Feature Extraction Network):**
   * Girdi görüntüsünden konvolüsyonel özellik haritaları elde edilir.
   * Özellik Piramit Ağı (FPN) kullanılarak, yüksek çözünürlüklü ancak sığ semantik değere sahip katman özellikleri, semantik değeri yüksek derin katman özellikleriyle kaynaştırılır.
   * FPN, yukarıdan-aşağıya ve aşağıdan-yukarıya yollarla özellik haritalarını birleştirir ve mekânsal çözünürlüğü artırır.



**Feature Pyramid Network (FPN) Nedir?**

**Aşağıdan Yukarıya Yol :** Backbone ağı (örneğin ResNet) kullanılarak farklı ölçeklerde özellik haritaları (C2, C3, C4, C5) üretilir. Üst katmanlara çıkıldıkça mekânsal boyut azalır, ancak semantik bilgi artar.

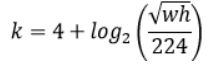
**Yukarıdan Aşağıya Yol :**C5’ten başlayarak, kanal sayısı 256’ya düşürülür ve üst-örnekleme ile çözünürlük artırılır. Daha düşük seviyedeki özellik haritalarıyla (C4, C3, C2) birleştirilir. Bozucu etkileri azaltmak için 3x3 konvolüsyon uygulanır.

**Çıktı (Prediction Layers):** Nihai piramit seviyeleri {P5, P4, P3, P2} olup, her biri nesne tespiti için aynı sınıflandırıcı ve kutu regresörü ile kullanılır.

1. **Bölge Öneri Ağı (Region Proposal Network - RPN):**
   * Görüntüde nesne olması muhtemel bölgeler önerilir.
   * Özellik haritalarında çapalar oluşturulur ve sınırlayıcı kutular belirlenir.
   * Önerilen bölgelerin nesne olup olmadığına dair bir güven eşiği tanımlanır.
2. **Öneri İşleme Modülü (RoI Hizalama):**
   * RPN'den gelen öneriler, belirli bir boyuta getirilir ve hizalanır.

**RoI Seçimi:** RoI, görüntü üzerinde odaklanılması gereken alanı belirleyerek gereksiz veriyi elemeye ve işlem yükünü azaltmaya yardımcı olur. Derin öğrenme tabanlı nesne tespiti algoritmalarında RoI seçimi, RPN (Region Proposal Network) tarafından önerilen farklı boyutlardaki bölgelere göre belirlenir.

RoI seçimi, RPN (Region Proposal Network) tarafından önerilen nesne bölgelerinin uygun özellik haritasına yönlendirilmesi ile yapılır. FPN (Feature Pyramid Network) gibi çok katmanlı özellik haritalarını kullanan sistemlerde, RoI’nin hangi katmanda işleneceğini belirlemek için aşağıdaki formül kullanılır:



Burada:

* **w, h** → RoI’nin genişliği ve yüksekliği
* **k\_0** → Referans ölçek için belirlenen sabit bir değer (genellikle 4)
* **224** → Standart bir referans boyutudur (ImageNet’te kullanılan)
* **k** → Seçilecek FPN seviyesidir (P2, P3, P4, P5)

 **Küçük RoI’ler (w, h küçük)** → Daha **yüksek çözünürlüklü** katmanlardan (P2, P3) alınır.

 **Büyük RoI’ler (w, h büyük)** → Daha **düşük çözünürlüklü** katmanlardan (P4, P5) çıkarılır.

 **Formül yardımıyla en uygun ölçek belirlenerek RoI buradan alınır.**

1. **Bölge-Tabanlı Konvolüsyonel Sinir Ağı (R-CNN):**
   * RPN önerileri, gemi sınıfına atanır ve sınırlayıcı kutular düzgün bir şekilde ayarlanır.
   * Nesnelerin maskeleri oluşturulur ve görüntüdeki nesneler konumları ve maskeleriyle tespit edilir.

Böylece, dört aşama sonrasında bir girdi görüntüsünden; (1) özellik haritaları oluşturulmuş, (2) RPN ile nesne önerileri çıkarılmış ve (3) bunlar RoI hizalaması vasıtasıyla belirli bir formata getirilerek tam bağlantılı katmanları da içeren R-CNN modülünde işlenmeye hazır hale getirilmişler ve (4) son modülde görüntüdeki nesneler konumları ve maskeleriyle beraber tespit edilebilir hale gelmişlerdir.

**Model Eğitimi**: ResNet101 kullanılmış. Modelde ikisi RPN, üçü R-CNN için olmak üzere toplam 5 adet kayıp fonksiyonu tanımlıdır. Model, başlangıç değeri 0,001 olan öğrenme oranı kullanılarak toplam 80 bin iterasyonda eğitilmiştir. Eğitimde 0,0001’lik bir ağırlık bozunumu (Overfitting’i önlemek için kullanılır. L2 regularizasyon) ve 0,9’luk momentum değerleriyle momentumlu Stokastik Gradyan İniş yöntemi ve geriyayılım algoritmaları kullanılmıştır.

**Sonuç:**

Güven eşiği 0.5 olarak belirlenmiş. Bu değerle model, test verisindeki 604 gemiden 558'ini doğru tespit etmiş, 46 gemiyi gözden kaçırmış ve 58 yanlış alarm üretmiştir. Modelin geri getirme (recall) değeri 0,9238, kesinlik (precision) değeri 0,9058 ve F1-skoru 0,9148 olmuştur.

**Modelin Tespit Performansı:**

* + Singapur Limanı'ndaki bir görüntüde 8 gemiden 7'si doğru tespit edilmiş, ancak bir küçük gemi gözden kaçırılmıştır.
  + Hong Kong'daki bir askeri limanda 10 gemiden 8'i doğru tespit edilmiş, ancak bir kara parçası gemi olarak yanlış sınıflandırılmış ve yan yana duran iki gemi tek gemi olarak işaretlenmiştir.
  + Rotterdam'daki bir limanda 9 gemiden 7'si doğru tespit edilmiş, ancak yan yana duran iki gemi tek gemi olarak işaretlenmiştir.

**GEMİ TESPİTİ UYGULAMASINDA YOLOV8 VE YOLOV9 ALGORİTMALARININ PERFORMANS DEĞERLENDİRMESİ**

YOLOv8 ve YOLOv9 mimarileri kullanılarak uzaktan algılama yöntemleriyle gemi tespiti çalışmaları gerçekleştirilmiştir. "Ships in Google Earth" adlı 1658 görüntüden oluşan veri seti kullanılarak bu iki modelin performansı karşılaştırılmıştır. Modeller, eğitim ve doğrulama kayıpları, kesinlik (precision), duyarlılık (recall) ve ortalama hassasiyet (mAP) kriterleri açısından değerlendirilmiştir. YOLOv9 modeli, özellikle başlangıçta daha hızlı yakınsama ve genel tespit performansında üstünlük sağlamıştır.

**Materyal ve metot**

YOLO, nesne tespiti görevini hesaplama karmaşıklığını azaltan tek bir regresyon problemi olarak ele alır. Görüntüyü ağa girdi olarak alır ve çıkış katmanında sınırlayıcı kutu koordinatlarını ve sınıf olasılıklarını doğrudan döndürür.

**YOLO Mimarisi:**

YOLO mimarisi üç ana bölümden oluşur:

* 1. Omurga (Backbone): Girdi olarak verilen resimlerin özniteliklerinin çıkarılması işlemi gerçekleştirilir.
  2. Boyun (Neck): Çıkarılan özniteliklerin birleştirilmesi ve seçilmesi işlemleri yapılır.
  3. Baş (Head): Tahminleme işlemi gerçekleştirilir.

**YOLO Versiyonları:**

* **YOLOv1 (2015):** İlk versiyon, nesne tespiti için temel bir model sunmuştur.
* **YOLOv2 (2016):** "Batch normalization" kullanılarak eğitim süresi kısaltılmış ve model doğruluğu artırılmıştır. Darknet-19 ağ yapısı geliştirilmiştir.
* **YOLOv3 (2018):** Darknet-53 ağ yapısı kullanılarak çoklu ölçeklerde tespit yapılabilmesi sağlanmış ve küçük nesnelerin tespitinde performans iyileştirilmiştir.
* **YOLOv4 (2020):** CSPDarknet53 mimarisi, mozaik veri artırma, CIOU kaybı ve SAM bloğu gibi yenilikçi tekniklerle daha yüksek doğruluk ve hız sağlanmıştır.
* **YOLOv5 (2020):** PyTorch tabanlı, hızlı ve hafif bir model olarak tanıtılmış, eğitim süreci basitleştirilmiştir.
* **YOLOv6 (2022):** YOLOv5'e alternatif olarak geliştirilmiş, hız ve doğruluk açısından iyileştirmeler içermektedir.
* **YOLOv7 (2022):** Gerçek zamanlı uygulamalar için optimize edilmiş, yüksek doğruluğa sahip bir modeldir.
* **YOLOv8 (2023):** Mimari, hız, doğruluk ve kullanım kolaylığı açısından çeşitli geliştirmeler içermektedir.
* **YOLOv9 (2024):** Derin bir sinir ağına dayalı yeni bir nesne algılama algoritması olarak geliştirilmiştir.

**YOLOv8 Algoritması:**

Mimari, kayıp fonksiyonları ve segmentasyon yeteneklerindeki ilerlemelerle öne çıkmaktadır.  YOLOv8, C2f (Cross Stage Partial) modülünü kullanarak derin öğrenme ağlarındaki gradyan akışını iyileştirir. Bu modül, katmanlar arası bilgi kaybını minimize eder ve öğrenme sürecini daha verimli hale getirir.

**YOLOv9 Algoritması:**

* **Omurga Ağı:** YOLOv9, CSPDarknet53 omurga ağını kullanarak giriş görüntüsünden özellikler çıkarır. Bu ağ, özellik haritasını iki bölüme ayırır ve bu bölümleri yeniden birleştirerek özellik gösterimini geliştirir.
* **Özellik Birleştirme:** YOLOv9, Mekansal Piramit Havuzlama (SPP) ve Yol Toplama Ağı (PAN) modüllerini kullanarak farklı ölçeklerdeki özellik haritalarını entegre eder. Bu, modelin öğrenme kapasitesini artırır ve daha karmaşık nesne tespiti görevlerinde daha yüksek doğruluk oranları elde edilmesini sağlar.
* **Entegre Yapı:** YOLOv9, YOLOv5'ten CSP Bloğu, YOLOv6'dan RepVGGBlock ve YOLOv7'den ELAN bloğunu bir araya getirerek RepNCSPELAN4 modülünü oluşturur. Bu entegre yapı, modelin öğrenme kapasitesini artırır.

**Veri seti**

* **Veri Seti:** Çalışmada, Kaggle üzerinde bulunan ve Roboflow tarafından sunulan "Ships in Google Earth" veri seti kullanılmıştır. Veri seti, 1658 uydu görüntüsünden oluşur ve farklı hava şartları, ışık koşulları ve gemi tiplerini içerir.
* **Veri Bölünmesi:** Veri seti, eğitim (%86, 1420 görüntü), doğrulama (%10, 159 görüntü) ve test (%5, 79 görüntü) olmak üzere üç kümeye ayrılmıştır.
* **Veri Ön İşleme:** Görüntüler, otomatik yönlendirme ve 640x640 piksel olacak şekilde yeniden boyutlandırma işlemlerinden geçirilmiştir.
* **Veri Artırma:** Veri artırma sürecinde, görüntüler yatay olarak çevrilmiş ve 90 derece döndürülmüştür. Her gemi, YOLOv5'e uygun sınırlayıcı kutularla etiketlenmiştir.

**Bulgular**

Yapılan çalışmada her iki YOLO mimarisi de 25 iterasyon çalıştırılmıştır. Her iki model içinde batch\_size değeri 16 olarak seçilmiştir. 25 iterasyon sonrasında modelin eğitimdeki başarısında değişiklik azaldığı için eğitim süreci tamamlandırılmıştır.

**Tartışma ve Sonuç:**

* **Genel Performans:** YOLOv9, genel olarak mAP ve duyarlılık açısından YOLOv8'e göre daha iyi performans sergilemektedir. Özellikle erken iterasyonlarda daha hızlı yakınsama göstererek, daha az iterasyonla yüksek performans elde edebilmektedir.
* **Model Seçimi:** Eğer öncelik tespit kalitesi ise, YOLOv9 daha iyi bir seçim olabilir. Ancak, seçim çıkarım hızı ve hesaplama verimliliği gibi diğer faktörlere de bağlı olabilir. YOLOv8, daha geniş bir uygulama yelpazesi sunmaktadır.
* **Gelecek Çalışmalar:** YOLO mimarilerinin başka türdeki veri setleri üzerinde de başarılı sonuçlar verdiği literatürde belirtilmektedir. Gelecekteki çalışmalarda, sadece gemi türleri değil, deniz araçlarının birçoğunu kapsayan bir veri seti ile çalışılması planlanmaktadır.