

İnsansız Hava Aracı Görüntülerinden Beton Yüzeyi Çatlaklarının YOLO Mimarileri İle Tespiti

Detection of Concrete Surface Cracks From Unmanned Aerial Vehicle Images Using YOLO Architectures

Ebrar ÖDÜBEK¹ , Muhammed Enes ATİK¹ 

¹ İstanbul Teknik Üniversitesi, İnşaat Fakültesi, Harita Mühendisliği Bölümü, İstanbul, 34000, Türkiye

Öz

Köprü, baraj ve kule gibi yapıların sağlamlığının denetlenmesi, sağlıklı ve güvenilir bir çevre ortamının oluşturulmasında kritik öneme sahiptir. Yüzey çatlaklarının tespiti, yapı sağlamlığının denetlenmesinde stratejik öneme sahiptir. Yapı yüzeylerindeki çatlakların hafif ve zararsız görünümü değişen zaman ve hava koşullarıyla daha tehlikeli hale gelebilmektedir. Yüzey çatlaklarının manuel yöntemlerle tespitinin gerçekleştirilmesi insan gücüne dayalı yüksek performans gerektirdiğinden ulaşılması zor ve riskli yapılarda düşük doğruluğa sebebiyet verebilmektedir. Derin öğrenme teknikleriyle yüzey çatlaklarının tespitinin gerçekleştirilmesi, manuel yöntemlerin yaratmış olduğu yüksek maliyet, zaman ve iş gücü problemlerine bir çözüm sunmaktadır. Bu çalışmada tek aşamalı nesne tespitini sağlayan You Only Look Once (YOLO) algoritmasının güncel versiyonlarının beton yüzeylerindeki çatlakları tespit performansları karşılaştırmalı şekilde analiz edilmiştir. Çalışmada, insansız hava aracı (İHA) görüntülerinden oluşan açık kaynak şeklinde sunulan CRACK veri seti kullanılmıştır. YOLOv9s 0.885 mean average precision (mAP) değeriyle en yüksek doğruluğa sahip algoritmadır.

Abstract

The inspection of the structural integrity of constructions such as bridges, dams, and towers is critical to creating a healthy and reliable environment. Detecting surface cracks is of strategic importance in monitoring structural integrity. Due to the high performance required for manual detection of surface cracks, it often leads to lower accuracy in hard-to-reach and risky structures, as it relies heavily on human effort. The seemingly mild and harmless appearance of surface cracks on structures can become more dangerous over time and under varying weather conditions. Detecting surface cracks using deep learning techniques has provided a solution to the high cost, time, and labor problems caused by manual methods. In this study, a comparative analysis was conducted on detecting cracks on concrete surfaces using the latest versions of the You Only Look Once (YOLO) algorithm, which enables single-stage object detection. The CRACK dataset, which is publicly available and consists of UAV images, was used in the study. It was concluded that the YOLOv9s algorithm achieved the highest accuracy with a mean average precision (mAP) of 0.885.

Anahtar Kelimeler: İnsansız hava aracı, Derin öğrenme, Nesne tespiti, YOLO, Beton yüzeyi çatlakları

Keywords: Unmanned aerial vehicle, Deep learning, Object detection, YOLO, Concrete surface cracks

1. Giriş

Kule, köprü, bina ve baraj gibi yapılarda beton yüzeyler meteorolojik koşullar ile zaman içerisinde yıpranmaya maruz kaldıkça yüzey üzerindeki hasarlar belirginleşmeye ve çevresel ortamda risk oluşturmaya başlamaktadır. Kullanılan düşük kaliteli inşaat malzemeleri, yapısal bozulma ve çürümeler yüzey hasar oluşumu ve gelişiminde olumsuz yönde rol oynamaktadır. Sağlıklı ve güvenli bir çevre sistemi oluşturulması amacıyla yapı hasarlarının hızlı ve otomatik şekilde tespit edilmesi önemlidir. Bunu gerçekleştirmek için insansız hava araçları (İHA) son zamanlarda öne çıkan teknoloji haline gelmiştir. Yüksek ve riskli yapılarda hasar tespiti çalışması yürütüldüğü esnada otomatik bir şekilde görüntü yakalanması yapının ilk durumunun değerlendirmesinde kritik bir role sahiptir. İHA ile yapının çeşitli açı ve perspektifinden yüksek kaliteli, detaylı ve yakın görüntülerini elde etmek hem daha uygun maliyetli hem de daha pratik bir yoldur. Buna ek olarak, çalışılması ve ulaşılması zor alanlarda, daha geniş alan tarayabilme özelliği sebebi ile İHA'lar yaygın şekilde kullanılmaktadır.

İHA görüntülerinin tespit, izleme ve analiz çalışmalarında kullanımı uzun bir süre manuel yöntemlerle gerçekleştirilmiştir. Manuel yöntemler beraberinde getirdiği daha fazla insan gücü, zaman ve maliyet gereksinimi ile tespit ve denetleme çalışmalarında performansı ve doğruluğu negatif yönde etkileyebilmektedir. Manuel yöntemlerde operatörün sürekli olarak İHA'yı takip etmesi zaman bakımından olumsuz yönde etki oluşturmaya birlikte operatörün süreç içerisinde anlık dikkatinin dağılması da hatalı nesne tespitine ve yüksek maliyete sebep olabilmektedir [1]. Araştırmacılar, görsel incelemelerin otomatik bir şekilde gerçekleştirilmesinin geleneksel insan gücü barındıran yöntemlerin oluşturduğu teknik sınırlamalarını azaltabilmesi sebebiyle nesne tespiti çalışmalarında bilgisayarlı görü tabanlı sistemlerin önemi üzerinde durmaktadır [2]. Bu sebeple, son zamanlarda beton yüzeylerdeki çatlakların otomatik bir şekilde tespit edilebilmesi için derin öğrenme tabanlı algoritmalar geliştirilmeye başlanmıştır. Çatlak tespiti çalışmaları doğrultusunda kullanılan her bir görüntü pikseline dayalı çeşitli gradyan özellikleri tasarlanmakta ve bir görüntü pikselinde çatlak olup olmadığına dair belirleyici ikili bir sınıflandırıcı kullanılmaktadır. Yerel ikili desen (YİD) tabanlı ve Gabar filtresine dayalı yaklaşımlar çatlak tespitine yönelik çalışmalarda kullanılmakta ve önerilmektedir [3].

İnsansız hava araçları tarafından elde edilen görüntülerde derin öğrenmeye dayalı algoritmaların kullanımı, nesne tespiti, segmentasyonu, sınıflandırılması, denetlenmesi ve izlenmesi gibi görev aşamalarının yüksek doğruluk ve performansla gerçekleştirilmesini sağlamaktadır. Derin öğrenme teknikleriyle nesne tespiti yöntemlerinde, görüntünün tek bir geçişiyle gerçek zamanlı nesne tespitini sağlayan YOLO algoritması (You Only Look Once), bölgesel öneri ağı desteğiyle tahmini nesne konumu oluşturan Faster R-CNN (Region- based Convolutional Neural Network) [5], nesne tespitine ek olarak segmentasyon görevini de tamamlayan Mask R-CNN [6] ve bir başka tek geçişli algoritmaya sahip nesne tespiti ve sınıflandırma algoritması olan SSD (Single Shot Multibox Detector) [7] gibi algoritmalar bulunmaktadır. Bu algoritmaların birbirlerine göre avantajları ve dezavantajları bulunmaktadır. Faster R-CNN'nin algılama sınırlayıcı kutuları, aynı veri seti ve görüntü üzerinde Mask R-CNN'in sınırlayıcı kutularına göre daha fazladır. Bu durum, ortak eğitim yöntemi nedeniyle sınırlayıcı kutu regresyonunda ortaya çıkan problemlerden kaynaklanmakta olup, Faster R-CNN'nin performansının daha yüksek olmasını sağlamaktadır [9]. Faster R-CNN, önce özellik haritasını oluşturan ve sonrasında bu haritayı ilgi alanı bölgelerine gönderen bir Bölge Öneri Ağı (RPN) kullanır. Daha sonra bu bölgelerde nihai sınıflandırma ve sınırlayıcı kutu regresyonu yapılır. Bu sebeple, Faster R-CNN doğruluk bakımından üstün bir algılama performansı gösterir. Ancak, Faster R-CNN'in tek katmanlı özellik eşlemesi, küçük hedef nesnelerin algılanmasında etkili olmadığından, algılama verimliliği açısından tek aşamalı dedektörlere göre daha düşük performans sergiler [10]. Xu vd. (2022)' ne göre Faster R-CNN'nin algılama sınırlayıcı kutuları, aynı veri seti ve görüntü için Mask R-CNN'in algılama sınırlayıcı kutularından daha fazla olup, ortak eğitim yönteminin sebep olduğu sınırlayıcı kutu regresyonundaki problemlerden dolayı Faster R-CNN'nin performansı daha yüksektir. Faster R-CNN, gerçek zamanlı nesne tespitinde sınırlı performans ve doğruluğa sahip olsa da, ağır bölge tabanlı aşamasından dolayı gömülü platformlarda kullanıma uygun değildir. Buna karşın, YOLO, tek bir sinir ağı ile tespit aşamasını optimize ederek,

genelleştirilebilir uçtan uca bir algoritma ile sınıflandırma ve tahmin aşamalarında daha fazla esneklik sunmaktadır [8].

Bu çalışmada derin öğrenme tekniklerinin beton yüzey çatlakları tespitinde kullanımı karşılaştırmalı şekilde analiz edilmiştir. Uygulama aşamasında YOLOv5 versiyonları, YOLOv8s ve YOLOv9s modelleri CRACK veri seti ile eğitilerek test edilmiş ve sonuçları birbirleriyle karşılaştırılmıştır. Karşılaştırma sonucunda modellerin performansları ve hassasiyet doğrulukları değerlendirilmiştir.

2. Literatür Araştırması

Literatürde derin öğrenme teknikleriyle İHA görüntülerinden beton yüzeyi çatlaklarının tespit edilmesi konusunda birçok çalışma mevcuttur. Woo vd. [1], İHA ve derin öğrenme tekniklerini kullanarak beton bir binanın dış yüzeyindeki yapısal çatlakları tespit etmek için bir yöntem önermektedir. Yöntem, İHA görüntüleri, nokta bulutu verisi ve tasarım çizimlerini birleştirerek yapısal çatlakları belirlemektedir. Süreç veri temini, ortofoto görüntü üretimi, derin öğrenme tabanlı çatlak tespiti, tasarım çizimleriyle yapı analizi ve çatlakların tanımlanması adımlarından oluşmaktadır. Çalışmada SDNET2018 ve İHA ile elde edilen görüntülerle oluşturulan veri seti kullanılmıştır ve YOLOv5 modeliyle çatlak tespiti gerçekleştirilmiştir. Modelde ortalama hassasiyet değeri (mAP) 0.46 değerindedir. Silva ve Lucena [12], beton yüzeylerdeki çatlakları tespit etmek amacıyla makine öğrenmesi tabanlı bir model geliştirmeyi hedeflemektedir. Modelin İHA ile entegre edilerek beton altyapı incelemelerinde otomasyon seviyesinin artırılması amaçlanmaktadır. Çatlak tespiti için evrişimsel sinir ağlarına (ESA) dayanan derin öğrenme algoritması kullanılmış ve beton yüzeylerdeki çeşitli koşulları dikkate alacak şekilde geliştirilmiştir. Modelin doğruluğu, öğrenme oranı, düğüm sayısı ve veri seti boyutu gibi parametreler üzerinde yapılan 48 deneyle test edilmiştir. Çalışma sonucunda %92.27 doğruluğa ulaşılmıştır. Yang vd. [13], yapısal güvenlik açısından kritik olan yüzey çatlaklarının tespiti için derin öğrenme algoritmalarını incelemektedir. 1024 x 1024 boyutlarında 2000 çatlak görüntüsü toplanmış, ancak veri yetersiz olduğu için görüntü artırımı uygulanmıştır. 416 x 416 boyutlarında 1600 eğitim ve 200 test görüntüsü içeren bir veri seti oluşturulmuş, görüntüler çatlak konumu ve sınıfı ile etiketlenmiştir. AlexNet, VGGNet13 ve ResNet18 modelleri kullanılarak çatlak tespiti yapılmış ve performansları YOLOv3 modeliyle karşılaştırılmıştır. Zhang vd. [17], köprülerdeki çatlakların manuel yöntemlerle tespitinin denetçilerin deneyimine dayalı olması nedeniyle yanlış değerlendirmelere yol açabileceğini vurgulayarak, çatlakların görüntü üzerinden otomatik tespiti ve izlenmesi için dört algoritmayı karşılaştırmıştır. İncelenen algoritmalar The Fast Haar Transformation, The Fast Fourier Transformation, Sobel Algoritması ve Canny Algoritması'dır. 1024 x 1024 piksel boyutlarında 1500 görüntüden oluşan bir veri seti kullanılmış ve çatlaklar etiketlenmiştir. Çalışma sonucunda %89.16 hassasiyet ve %91.16 geri çağırma değeri elde edilmiştir. Li vd. [18], baraj yüzeylerindeki çatlakların tespitini kolaylaştırmak amacıyla İHA ve YOLOv8'e dayalı bir derin öğrenme yöntemi geliştirmeyi hedeflemiştir. YOLOv8, gerçek zamanlı performansı ve baraj yüzeyi arka planının basitliği nedeniyle tercih edilmiştir. Çalışmada, 56.000'den fazla beton çatlak görüntüsü içeren SDNET2018 veri seti kullanılmıştır. Çalışma kapsamında YOLOv8 algoritması ile %95.2 hassasiyet, %95.8 geri çağırma ve %95.5 F1 skor elde edilmiştir. Gupta ve Dixit [19], yapıların çatlaklarının manuel tespiti yerine, derin öğrenme tabanlı yöntemlerle daha hızlı, maliyet etkin ve doğru bir şekilde tespit edilmesini amaçlamıştır. Araştırmacılar, açık erişim veri setlerinden toplam 22.000 görüntü (11.000 çatlaklı, 11.000 çatlaksız) toplamış ve bu görüntülerin çözünürlüklerini 256 x 256 piksel olarak ayarlamışlardır. İnce çatlakların daha iyi tespiti için kontrast sınırlı adaptif histogram eşitleme (CLAHE) yöntemi kullanılmış ve veri setine kaydırma, döndürme ve kesme gibi veri artırma işlemleri uygulanmıştır. VGG16, VGG19 ve ResNet50 modelleri transfer öğrenme yöntemiyle eğitilmiş ve performansları karşılaştırılmıştır. Çalışma sonucunda, ResNet50 ile %99,2, VGG16 ile %97,2 ve VGG19 ile %97,3 doğruluk elde edilmiştir.

3. Veri ve Yöntem

3.1. Veri seti

Beton yüzeylerindeki çatlakların belirlenmesi için farklı veri setleri bulunmaktadır. Bu çalışmada Google Dataset Search aracılığı ile Roboflow üzerinde oluşturulmuş ve açık kaynak CRACK veri seti [14] kullanılmıştır (Şekil 1). Görüntü sayısı nispeten az olsa da, farklı çatlak yapıları iyi bir şekilde temsil edilmiştir. Böylece eğitim esnasında çatlakların yeterli şekilde tanınması için veri sağlanmaktadır. Crack veri setinde çatlaklı ve çatlaksız olarak toplamda 1551 görüntü bulunmaktadır. Veri setinin YOLO mimarilerinde eğitime aşamasında aşırı uyumunu önlemek adına ön işleme adımları Roboflow platformu üzerinde gerçekleştirilmiştir. Veri artırma yöntemi olarak saat yönü ve tersi yönünde döndürme, $\pm 15^\circ$ döndürme, pozlama ve gürültü ekleme yöntemleri uygulanarak 2929 görüntü sayılı veri seti oluşturulmuştur. Görüntülere döndürme ve gürültü ekleme tekniği ile testi gerçekleştirilecek modele daha fazla nesne örneği sağlanmıştır. Görüntüler %70 eğitim, %10 doğrulama ve %20 test seti olacak şekilde bölünmüştür. Veri setindeki görüntüler 640 x 640 piksel boyutlarında oluşturulmuştur. Veri seti, isimler, sınıf sayısını ve veri seti hakkında bilgi içeren konfigürasyon dosyasına sahip YOLOv5, YOLOv8 ve YOLOv9 algoritmalarına uygun formatta modellerde test edilmek üzere hazır hale getirilmiştir.



Şekil 1. CRACK veri seti beton çatlak görüntüsü örnekleri [14].

3.2. Yöntem

Çalışma kapsamında, beton yüzeyi çatlaklarının tespitini gerçekleştirmek amacıyla son zamanlarda yaygın bir şekilde kullanılmakta olan derin öğrenme tekniklerinden biri olan YOLO mimarileri kullanılmıştır. YOLOv5'in eğitilmiş olan beş modeli: nano (n), küçük (s), orta (m), büyük (l) ve ekstra büyük (x) modelleri, YOLOv8s ve YOLOv9s modelleri CRACK veri setinde test edilmiştir.

3.2.1. You Only Look Once (YOLO) Algoritması

YOLO [4], nesne tespiti için kullanılan derin öğrenme tabanlı bir algoritmadır. Bu algoritma, ilgili görüntüyü tek seferde sinir ağından geçirerek, tespit edilen nesnenin koordinatlarını belirlemekte ve nesnenin hangi sınıfa ait olduğunu tahmin etmektedir. YOLO, nesne tespitini bir regresyon problemi olarak ele almaktadır [20]. YOLO nesnelerin konumlarını, kategorilerini ve ilgili güven puanlarını hesaplayarak nesneleri gerçek zamanlı olarak tespit edebilmektedir. Gerçek zamanlı nesne tespiti sağladığı için, YOLO yüksek hız ve doğruluk sunmaktadır [15]. Çalışma prensibi, görüntüleri ızgaralara bölerek her hücredeki nesneleri potansiyel sınıflarına atamaktır. İşlemi tek bir sinir ağı ile yürüten YOLO, hızlı performansı nedeniyle yüksek hassasiyet gerektirmeyen işler için geniş ölçüde kullanılmış ve derin öğrenme dünyasında popüler bir algoritma haline gelmiştir. 2015 yılında geliştirilen ilk YOLO modelinden bu yana, farklı amaçlar ve çalışma koşullarına uyum sağlayabilmesi için çeşitli versiyonları oluşturulmuştur. Bu çalışma kapsamında, beton yüzeyindeki çatlakları tespit etmek için performansları değerlendirilen ve karşılaştırılan YOLO algoritmalarından YOLOv5 [16] (versiyonlar: YOLOv5s, YOLOv5x, YOLOv5n, YOLOv5m, YOLOv5l), YOLOv8s [16] ve YOLOv9s [21] modellerinin özelliklerine sırasıyla yer verilmiştir (Çizelge 1).

Çizelge 1. Modellerin öğrenilebilir parametre sayıları.

<i>Model</i>	<i>Parametre Sayısı</i>
YOLOv5s	7.24M
YOLOv5x	86.2M
YOLOv5n	1.8M
YOLOv5m	20.9M
YOLOv5l	46.1M
YOLOv8s	11.1M
YOLOv9s	7.3M

YOLO algoritmasının çalışma prensibi, tek bir sinir ağında uçtan uca eğitim stratejisiyle beraber nesnelerin konumunu ve sınıfını görüntünün tek bir geçişinde tespit etmektir [8]. ESA prensibindeki görüntü üzerinde filtre kaydırma yöntemi yerine YOLO algoritması, görüntüyü bir dizi $S \times S$ ızgara hücrelerine bölerek nesne tespiti aşamasını başlatmaktadır. Bölünen her hücrede nesne var ise sınırlayıcı kutu ve nesnenin sahip olduğu sınıf olasılık puanı hesabı yapılmaktadır. Bu sınırlayıcı kutu nesne ile ilgili 5 parametre bilgisini içermektedir. Sınırlayıcı kutunun merkez noktasının hücreye göre olan yatay ve dikey konumları (x, y), genişliği (w), yüksekliği (h) ve bu sınırlayıcı kutunun belirli bir nesneyi içermeye olasılığına bağlı oluşturduğu güven skoru değeri sınırlayıcı kutunun parametreleridir. Güven skoru C , Denklem 2'deki ızgara hücrelerinde ifade edilen sınıf olasılığını ve tahmin edilen sınırlayıcı kutunun temel gerçeğe göre uyum derecesine göre hesaplanmaktadır [8].

Denklem 1'de verilen eşitlikte $P(\text{Class}_i | \text{Object})$ bir sınıfın koşullu olasılığını ifade etmektedir. Hücredeki sınırlayıcı kutu ızgara hücrelerinde nesneler bulunuyorsa $P_r(\text{Object}) = 1$, bulunmuyorsa $P_r(\text{Object}) = 0$ olarak ifade edilmektedir. $\text{IOU}_{\text{pred}}^{\text{truth}}$ ise temel doğruluk kutusu (A) ve tahmin edilen sınırlayıcı kutunun (B) alanları arasındaki kesişimin birleşime oranını ifade etmektedir. Sınırlayıcı kutuları ızgara hücrelerindeki nesnelere atama işlemi, sınırlayıcı kutunun sınıfa özgü belirlediği güven skoru eşik değerine göre yapılmaktadır. Ek olarak, ızgara hücrelerinde gereksiz sınırlayıcı kutu tahmini ortadan kaldırmak için maksimum olmayan bastırma yöntemi kullanılmaktadır. YOLO algoritması toplam kare hatasının çok parçalı kayıp fonksiyonunu işleme alarak tespit işlemini optimize etmektedir [8].

$$C = P(\text{Class}_i | \text{Object}) P_r(\text{Object}) * \text{IOU}_{\text{pred}}^{\text{truth}} \quad (1)$$

$$\text{Intersection over Union (IoU)}_{\text{pred}}^{\text{truth}} = \frac{A \cap B}{A \cup B} \quad (2)$$

Denklem 3'te fonksiyon, L kaybını, yerleştirme ve sınıf olasılığı ile ilgili beş terim ile ortaya koymaktadır. Denklem 3'ün ilk iki terimi koordinat kayıpları ve sınırlayıcı kutuların boyutlarına, üçüncü ve dördüncü terimi bir nesnenin tespit edilmesinin güvenilirliğini tahmin etmeye, beşinci terim ise her bir ızgara hücrelerindeki nesnenin sınıflandırılmasına atıfta bulunmaktadır.

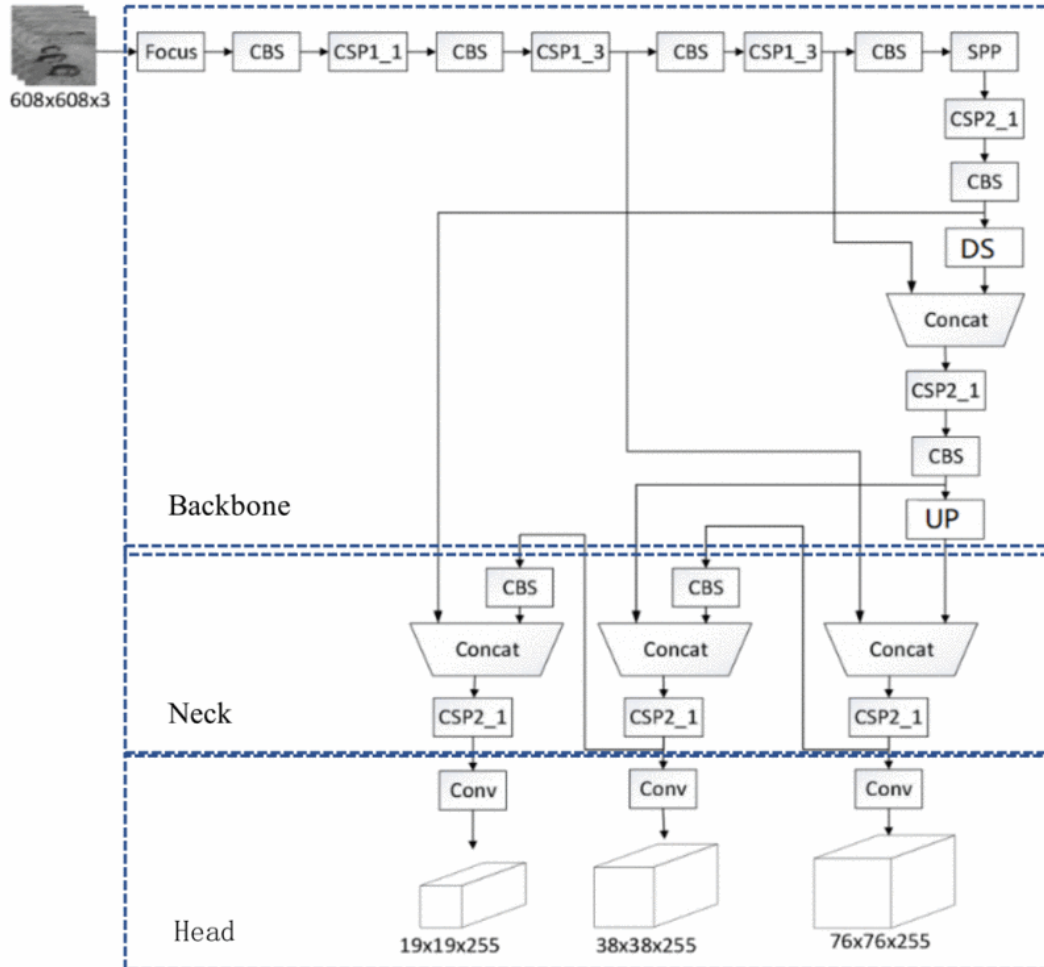
$$\begin{aligned}
L = & \beta_{\text{coord}} \sum_{i=0}^{S \times S} \sum_{j=0}^B 1_{ij}^{\text{obj}} [(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2] + \beta_{\text{coord}} \sum_{i=0}^{S \times S} \\
& \times \sum_{j=0}^B 1_{ij}^{\text{obj}} \left[(\sqrt{w_i} - \sqrt{\hat{w}_i})^2 + (\sqrt{h_i} - \sqrt{\hat{h}_i})^2 \right] + \sum_{i=0}^{S \times S} \\
& \times \sum_{j=0}^B 1_{ij}^{\text{obj}} (C_i - \hat{C}_i)^2 + \text{noobj} \sum_{i=0}^{S \times S} \sum_{j=0}^B 1_{ij}^{\text{noobj}} (C_i - \hat{C}_i)^2 + \sum_{i=0}^{S \times S} 1_i^{\text{obj}} \\
& \cdot \sum_{c \in \text{classes}} (p_i(c) - \hat{p}_i(c))^2
\end{aligned} \quad (3)$$

Denklem 3'te 1_{ij}^{obj} i 'inci ızgara hücrelerinin bir kısmını içerdiğini göstermektedir. 1_{ij}^{obj} i 'inci ızgara hücrelerine yönelik tahmini gerçekleştiren j 'inci sınırlayıcı kutuyu ifade etmektedir. x_i, y_i, w_i ve h_i o ızgara

hücresi i için sınırlayıcı kutunun koordinatlarını, genişliğini ve yüksekliğini ifade etmektedir. C_i ve $p_i(c)$ parametreleri ise i 'inci ızgara hücresindeki nesne varlığını ve sınıf olasılığını, B sınırlayıcı kutuların sayısını, β koordinatı koordinat kaybının ağırlığını ve noobj parametresi ise nesneleri olmayan sınırlayıcı kutular için kaybın ağırlığını ifade etmektedir [8].

3.2.2. YOLOv5 Modeli

Ultralytics tarafından algılama doğruluğu ve hız açısından önceki modellere göre geliştirilen YOLOv5 modeli, diğer YOLO modelleri gibi girdi, omurga, boyun ve kafa bölümlerinden meydana gelmektedir. Girdi tarafında, her bir eğitim verisi grubu, veri kümesini zenginleştirmek için rastgele ölçeklendirme, rastgele kırpmaya, rastgele döndürme ve Mosaic veri genişletme işlemleri gerçekleştirebilen bir veri yükleyici aracılığıyla iletilmektedir [10]. Girdi katmanını görüntüyü dilimleme ve derinlemesine birleştirme özelliğini taşımaktadır [11]. Omurga kısmında, gradyan akışını farklı yollara bölen ve her yolda yalnızca belirli bir kısmın hesaplanmasını sağlayan CSPDarknet53 modeli kullanılmaktadır. Bu, aşamalar arası hiyerarşik birleştirme ile hesaplama çabasını azaltmakta ve özellik çıkarma kapasitesini artırmaktadır [10]. Omurga, görüntü özelliği çıkarımı için Çapraz Aşamalı Kısmi (CSP) ağı ve Uzamsal Piramit Havuzu'nu (SPP) derleme özelliğine sahiptir [11]. Boyun kısmı, çok ölçekli özellik füzyonu için Path Aggregation Feature Pyramid Network (PAFPN) özellik piramit yapısını kullanarak, güçlü semantik özellikleri ve sığ konum bilgilerini yüksek seviyelerde birleştirmektedir. "Yukarıdan aşağıya + aşağıdan yukarıya" iki kanallı bir yaklaşım benimsenmektedir. Kafa bölümünde, tam IoU (CIoU) kayıp fonksiyonu kullanılarak, tekrar tespit çerçevesinin örtüşme alanı, merkez mesafesi ve tahmin çerçevesinin hedef çerçeveye en boy oranı gibi faktörler dikkate alınmaktadır. Bu uygun kayıp fonksiyonu, ağırlı öğrenme etkisini ve hızını artırmaktadır [10]. YOLOv5'in performansı, hiperparametre optimizasyonu ve deney takibi entegre edilerek Ultralytics tarafından iyileştirilmiştir [11]. YOLOv5'in genel ağ yapısı Şekil 2'de sunulmuştur.

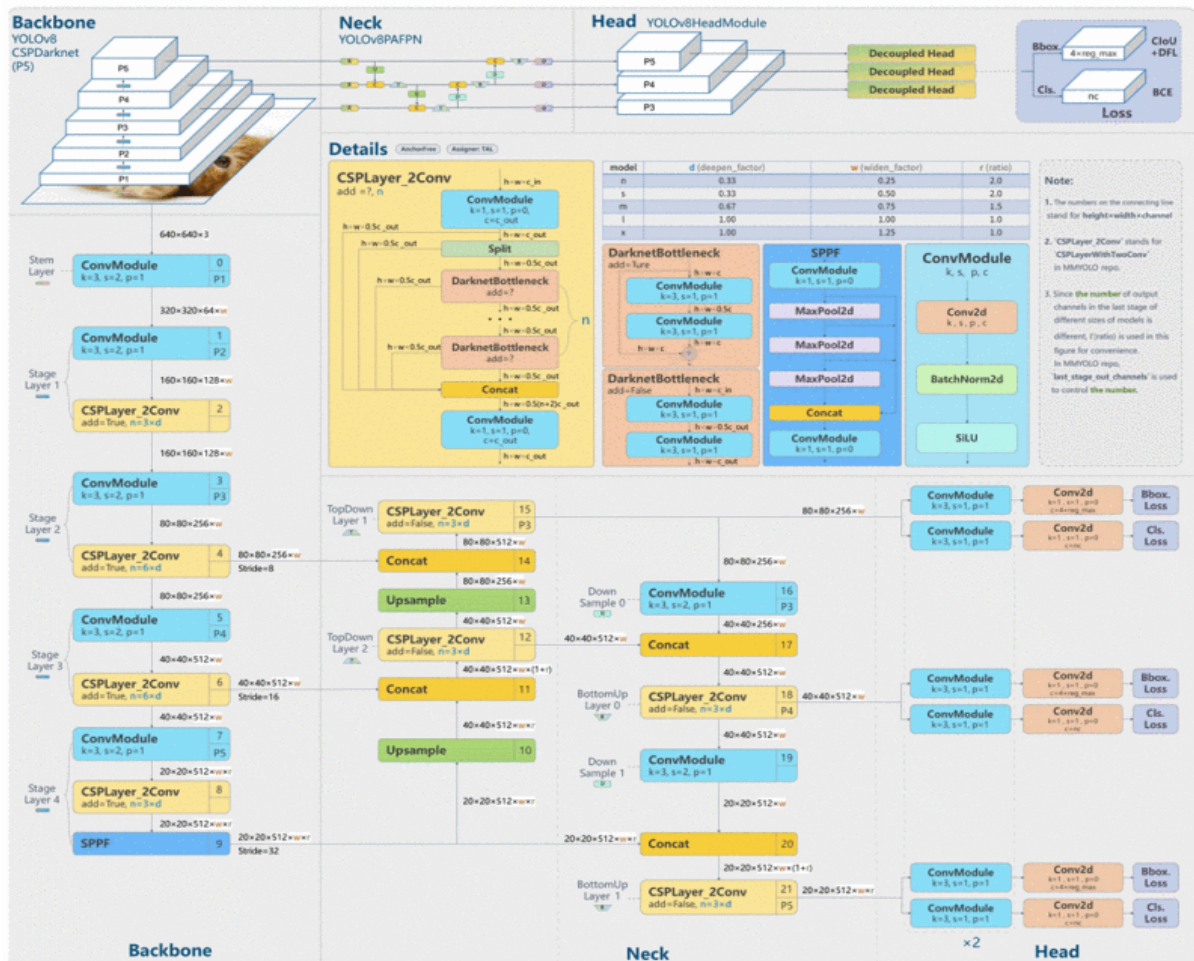


Şekil 2. YOLOv5'in genel yapısı [11].

YOLOv5s modeli, nesne tespitini tek bir geçişte gerçekleştiren, az sayıda parametreye sahip olduğu için küçük ve hızlı bir model olarak öne çıkan YOLOv5 algoritmalarından biridir. Düşük parametre sayısı nedeniyle hesaplama gereksinimi de düşüktür. Ancak, bu özellikleri nedeniyle yüksek hassasiyet gerektiren çalışmalarda genellikle beklenen performansı verememektedir. YOLOv5x modeli, büyük ve karmaşık veri setleri için daha uygun olan, tek geçişte nesne tespiti sağlayan ve yüksek hız ile doğruluk sunan YOLOv5 algoritmalarından biridir. Çok sayıda parametre içermesi ve büyük, karmaşık bir yapıya sahip olması nedeniyle, detaylı nesnelerin algılanmasında yüksek performans gösterir. Ancak, bu yüksek parametre sayısı, modelin eğitim süresini uzatarak olumsuz yönde etkileyebilir. YOLOv5n, YOLOv5m ve YOLOv5l algoritmaları, daha önce eğitilmiş YOLOv5 modellerinin üzerine inşa edilmiş, geliştirilmiş ve farklı hiperparametreler ile model boyutlarına sahip varyantlardır. Bu algoritmalar, farklı veri setleri ve amaçlar için uygun doğruluk ve hassasiyet seviyeleri sunmak üzere tasarlanmıştır. Diğer YOLOv5 mimarileri gibi, tek bir sinir ağıyla tek geçişte nesne tespiti gerçekleştirdikleri için, gerçek zamanlı nesne tespiti alanında yaygın olarak kullanılmaktadırlar.

3.2.3. YOLOv8 Modeli

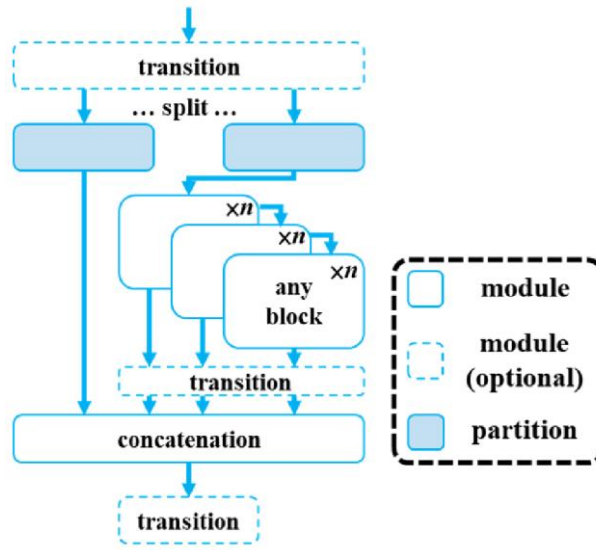
YOLOv8 [22] algoritması, önceki YOLO sürümlerinin avantajlarından yararlanarak daha yüksek hız, doğruluk ve hassasiyet sağlamak amacıyla geliştirilen, YOLO'nun en son versiyonlarından biridir. YOLOv8, algılama, segmentasyon ve sınıflandırma için kullanılan beş ayrı modele sahiptir. Bu 5 model, s, n, l, x ve m olarak adlandırılmaktadır. YOLOv8n, en hızlı ve en düşük doğruluğa sahip modelken, YOLOv8x ise en yavaş ama en yüksek doğruluğa sahip model olarak öne çıkmaktadır. Modeller arasındaki temel fark, boyutları ve bu boyutların performansa etkisidir. YOLOv8'in ağ detayları Şekil 3'te gösterilmiştir.



Şekil 3. YOLOv8'in genel yapısı [11].

3.2.4. YOLOv9 Modeli

YOLOv9 [22], Programlanabilir Gradyan Bilgisi (PGI) ve Genelleştirilmiş Verimli Katman Toplama Ağı (GELAN) (Şekil 4) yeni özellikler sunarak gerçek zamanlı nesne tespiti için ileriye bir model ortaya koymaktadır. Bu model, MS COCO veri setinde verimlilik, doğruluk ve uyarlanabilirlik açısından yeni ölçütler sunarak son derece önemli gelişmeler göstermektedir. YOLOv9 projesi, Ultralytics ve YOLOv5 tarafından sağlanan kod ve mimariler üzerinde geliştirilerek inşa edilmiştir. PGI ve çok yönlü GELAN mimarisi (Şekil 4), modelin öğrenme kapasitesini ve algılama süreci boyunca önemli bilgilerin korunmasını sağlayarak doğruluk ve performans açısından diğer modellere göre üstünlük göstermektedir. YOLOv9 modelinin temel yenilikleri Bilgi Darboğazı İlkesi ve Tersinir İşlevlerin yenilikçi kullanımı, model mimarisinin merkezinde yer almakta ve YOLOv9'un yüksek verimlilik ve doğruluğa sahip olmasını sağlamaktadır.



Şekil 4. GELAN Mimarisi [22].

4. Uygulama

Uygulama Python programlama dilinde Google Colaboratory platformu üzerinde gerçekleştirilmiştir. Google Colaboratory ücretsiz şekilde 12.7 GB RAM, Tesla T4 GPU ve 78 GB depolama alanı sağlamaktadır. Test edilen modeller 100 epok ve 16 batch size olacak şekilde eğitilmiştir. YOLOv5 (ve versiyonları), YOLOv8s ve YOLOv9s algoritmaları optimum parametreler ile CRACK veri seti üzerinde eğitilmiş ve test edilmiştir. Eğitim parametreleri donanım özelliklerine bağlı olarak en iyi metrik değerleri elde edilecek şekilde belirlenmiştir.

Deney ortamında gerçekleştirilen model testleri arasından en iyi modeli seçebilmek için hassasiyet (P), geri çağırma (R) ve ortalama hassasiyet (mAP) değerlerine ihtiyaç duyulmaktadır. P hassasiyet, doğru tahmin edilen nesnelerin sayısının tahmin edilen toplam nesne sayısına bölünmesiyle elde edilmektedir. Denklem 4'te doğru pozitif (DP) değeri modelin doğru bir şekilde tespit ettiği nesneleri, yanlış pozitif (YP) değeri ise modelin yanlış bir şekilde tespit ettiği yani aslında olmayan nesneleri ifade etmektedir. Geri çağırma değeri ise, gerçek nesnelerden kaç tanesinin doğru bir şekilde tespit edildiğini ölçmektedir. Denklem 5'te yanlış negatif (YN) değeri modelin tespit edemediği aslında var olan nesneleri ifade etmektedir. Modelde mAP değeri ise tüm sınıflara karşılık gelen hassasiyet değerinin (AP) değerlerinin ortalamasıdır. Modelin genel algılama performansını ölçmek için kullanılmaktadır. F1 skoru ise hassasiyet ve geri çağırmanın harmonik ortalamasına karşılık gelmektedir. F1 skoru, hassasiyet ve geri çağırma arasındaki dengeyi ölçmektedir. Bir modelin yüksek hassasiyete sahip olması, tespit edilen nesnelerin büyük çoğunluğunun doğru olduğu anlamına gelmektedir. Ancak, bu model aynı zamanda

birçok nesneyi kaçırabilmektedir (düşük geri çağırma). Benzer şekilde, yüksek geri çağırmaya sahip bir model, doğru olan çoğu nesneyi tespit edebilmektedir, ancak aynı zamanda birçok yanlış tespit de yapabilmektedir (düşük hassasiyet). Değerlendirme metrikleri şu şekilde hesaplanmaktadır:

$$\text{Hassasiyet} = \frac{\text{Doğru Pozitif (DP)}}{\text{Doğru pozitif (DP)} + \text{Yanlış Pozitif (YP)}} \quad (4)$$

$$\text{Geri Çağırma} = \frac{\text{Doğru Pozitif (DP)}}{\text{Doğru pozitif (DP)} + \text{Yanlış Negatif (YN)}} \quad (5)$$

$$\text{F1 skor} = 2 \times \frac{(\text{Hassasiyet} \times \text{Geri çağırma})}{(\text{Hassasiyet} + \text{Geri çağırma})} \quad (6)$$

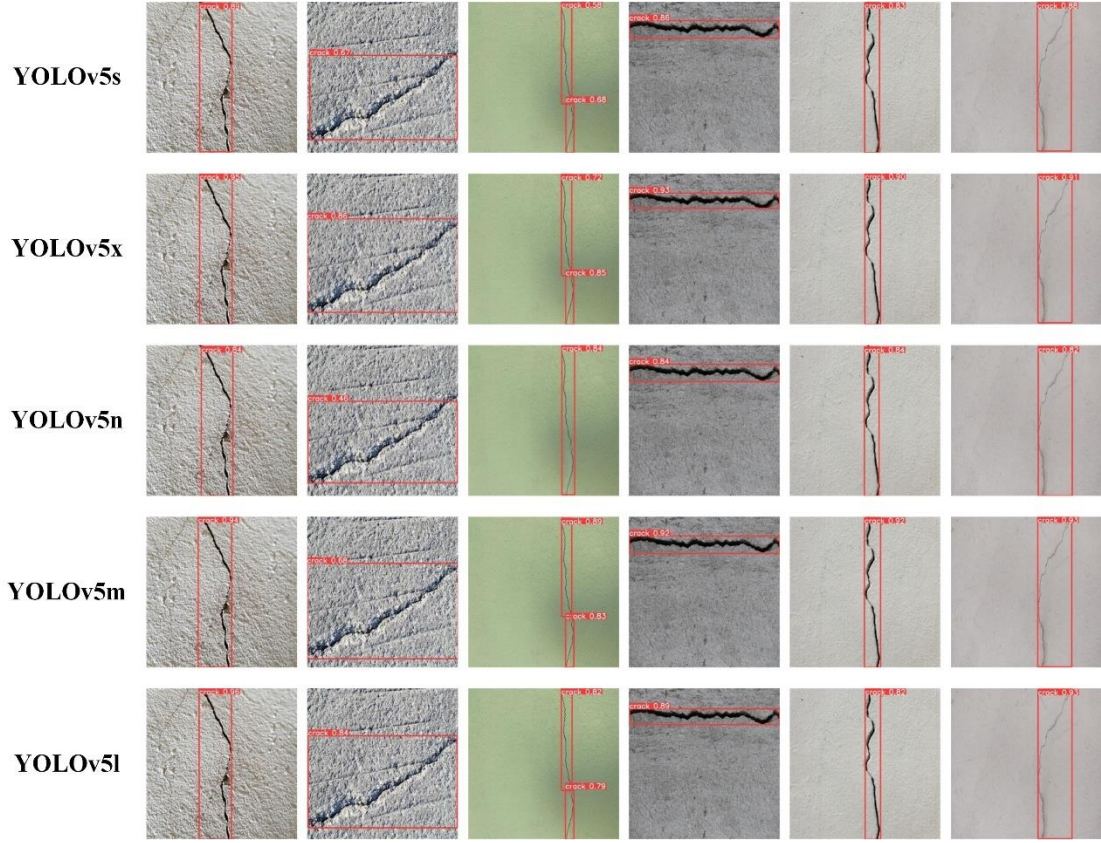
5. Bulgular ve Tartışma

Deney kapsamında kullanılan CRACK veri setinden test için belirlenen 580 çatlaklı çatlaksız görüntüde YOLOv5 versiyonları, YOLOv8s ve YOLOv9s modellerin test edilmiştir. 580 test görüntüsü içerisinde 710 adet çatlak etiketlenmiştir. Elde edilen hassasiyet (P), geri çağırma (R), F1-skor ve mAP değerleri Çizelge 2’de verilmiştir.

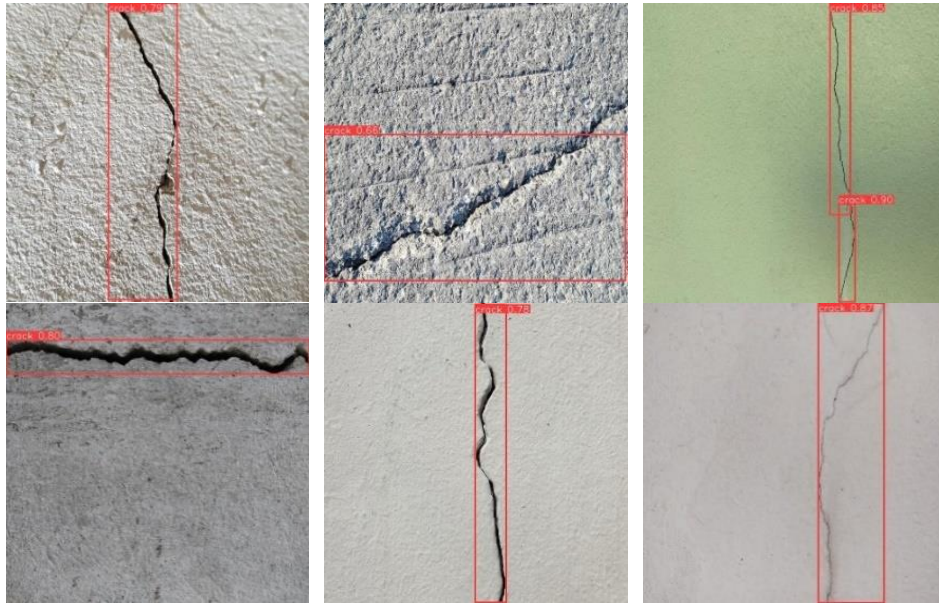
CRACK veri setinin YOLOv5 versiyonları (s, x, l, m, n), YOLOv8s ve YOLOv9s modellerindeki test işleminden sonra elde edilen değerler sonucunda en yüksek hassasiyet (P) ve geri çağırma değerleri sırasıyla 0.885 ve 0.776 ile YOLOv9s modelinde elde edilmiştir. En yüksek ortalama hassasiyet değerine (mAP) sahip model ise 0.836 değeri ile YOLOv9s modelidir. En yüksek hassasiyet (P) değerinin YOLOv9s modelinde elde edilmesi, YOLOv9s modelinin tespit edilen nesnelerde sayıca en fazla doğru nesne tespiti gerçekleştirdiğini ortaya koymaktadır. Bu durum, YOLOv9s modelinin nesneleri tespit etmedeki hassasiyetini ve detaycılığını ön plana çıkarmaktadır. Ortalama hassasiyet değerine göre de YOLOv9s modeli en başarılı algoritma olmuştur. Beton çatlak görüntülerinden oluşturulan veri seti, yakın mesafeden çekilmiş ve farklı nesne türlerini içermeyen görüntülerden meydana gelmektedir. Hassasiyet bakımından YOLOv5x algoritması içerdiği yoğun parametre özellikleriyle birlikte daha yüksek performans sağlama kabiliyetine sahip olmasına rağmen bu veri setinde YOLOv9s ortalama hassasiyet bakımından daha iyi performans göstermesi detay içermeyen veri setleri için yoğun parametre ihtiyacının gerekmediği ve en hızlı olan algoritmanın bu veri setleri için daha uygun olduğunu çıkarımını desteklemektedir. Algoritmalara ait tahmin görselleri Şekil 5, Şekil 6 ve Şekil 7’de sunulmuştur. Her bir algoritmanın hata matrisleri Şekil 8’de gösterilmiştir.

Çizelge 2. CRACK veri setinin YOLOv5, YOLOv8 ve YOLOv9 model test sonuçları.

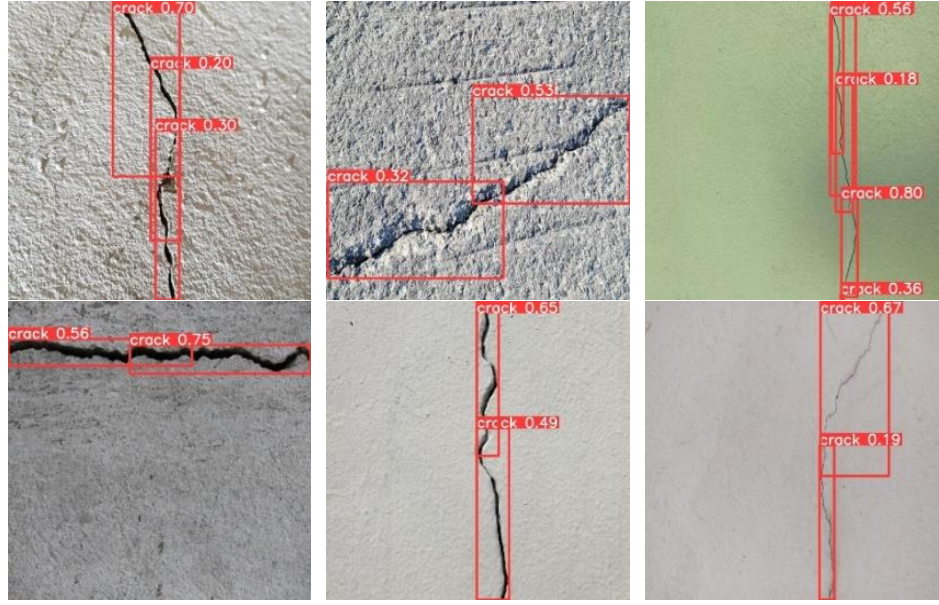
<i>Model</i>	<i>Hassasiyet</i>	<i>Geri Çağırma</i>	<i>F1 skor</i>	<i>mAP50</i>	<i>mAP50 - 95</i>
YOLOv5s	0.805	0.765	0.784	0.780	0.587
YOLOv5x	0.860	0.752	0.802	0.810	0.640
YOLOv5n	0.809	0.764	0.782	0.802	0.589
YOLOv5m	0.833	0.757	0.793	0.797	0.628
YOLOv5l	0.852	0.745	0.795	0.785	0.615
YOLOv8s	0.806	0.759	0.782	0.782	0.603
YOLOv9s	0.885	0.776	0.827	0.836	0.678



Şekil 5. Nesne tespitinin görsel sonuçları YOLOv5.



Şekil 6. Nesne tespitinin görsel sonuçları YOLOv8s.



Şekil 7. Nesne tespiti için görsel sonuçları YOLOv9s.

		GERÇEK					GERÇEK					GERÇEK					GERÇEK		
		Çatlak	Arkaplan	Toplam			Çatlak	Arkaplan	Toplam			Çatlak	Arkaplan	Toplam			Çatlak	Arkaplan	Toplam
TAHMİN	Çatlak	597	329	926	TAHMİN	Çatlak	595	290	885	TAHMİN	Çatlak	601	312	913	TAHMİN	Çatlak	588	308	896
	Arkaplan	113	0	113		Arkaplan	115	0	115		Arkaplan	109	0	109		Arkaplan	122	0	122
Toplam		710	329		Toplam		710	290		Toplam		710	312		Toplam		710	308	

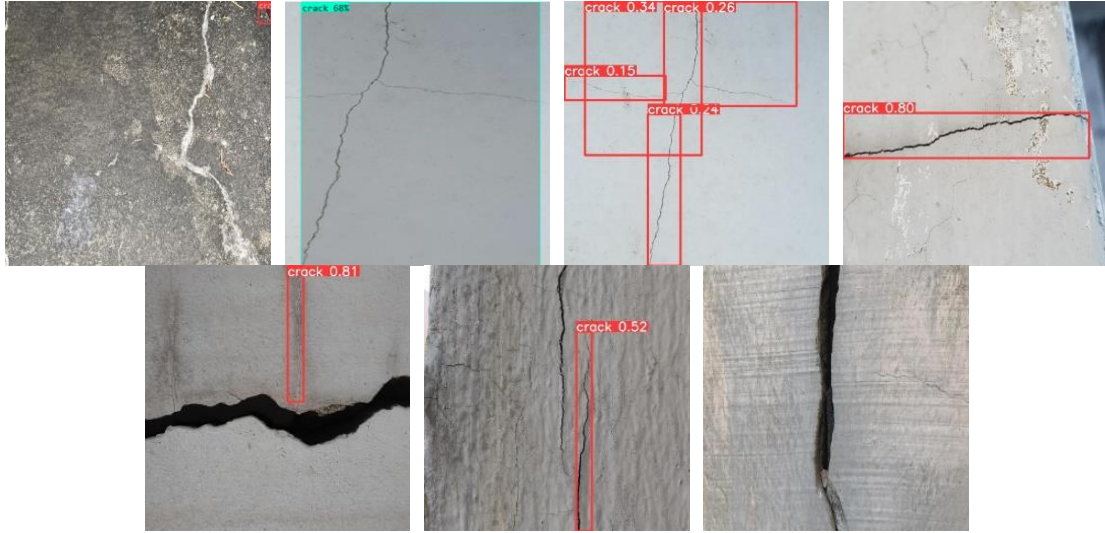
(a) (b) (c) (d)

		GERÇEK					GERÇEK					GERÇEK		
		Çatlak	Arkaplan	Toplam			Çatlak	Arkaplan	Toplam			Çatlak	Arkaplan	Toplam
TAHMİN	Çatlak	574	248	822	TAHMİN	Çatlak	603	250	853	TAHMİN	Çatlak	600	343	943
	Arkaplan	136	0	136		Arkaplan	107	0	107		Arkaplan	110	0	110
Toplam		710	248		Toplam		710	250		Toplam		710	343	

(e) (f) (g)

Şekil 8. Her bir yönteme ait hata matrisi. (a) YOLOv5s; (b) YOLOv5x; (c) YOLOv5n; (d) YOLOv5m; (e) YOLOv5l; (f) YOLOv8s; (g) YOLOv9s.

Yanlış tespit edilen veya hiç tespit edilemeyen çatlaklar da bulunmaktadır. Şekil 9'da yanlış ve eksik tahminler için örnek görseller sunulmuştur. YOLOv5x ve YOLOv8s modelleri derin çatlakları tespit edebilmiş ancak yatay şeklinde devam eden ince çatlakları tam olarak tespit edememiştir. Bununla beraber derin ve belirgin çatlakların tespit edilemediği durumlar da bulunmaktadır. YOLOv5m ve YOLOv5s modelleri derin çatlakları tespit edememiştir. Bazı durumlarda bir çatlak bölünerek farklı çatlaklar gibi tespit edilmiştir. Özellikle YOLOv5s yapısı gereği basit ve küçük nesneleri tespit etmede daha başarılı olduğundan bu tip çatlakları tespit etmede düşük performans göstermiştir. Test süreleri göz önüne alındığında, gerçek zamanlı uygulamalar için uygun sonuçlar elde edilmektedir. 24 fps'lik standart bir video görüntüsünde, kullanılan tüm algoritmalar uygun şekilde tespit yapabilmektedir (Çizelge 3).



Şekil 9. Algoritmaların eksik veya yanlış tespitler için örnek görseller.

Çizelge 3. Çalışmada kullanılan modellerin CRACK veri setindeki eğitim ve test süreleri.

Model	Eğitim Süresi	Test Süresi (1 görüntü için)
YOLOv5s	36d 36s	9.2 ms
YOLOv5x	127d 30s	32.5 ms
YOLOv5n	38d 48s	10.2 ms
YOLOv5m	53d 18s	8.3 ms
YOLOv5l	82d 42s	26.6 ms
YOLOv8s	127d 24s	11.7 ms
YOLOv9s	187d 30s	13.2 ms

Çalışma sonuçları literatürde Crack veri setini kullanmış çalışmaların sonuçları ile karşılaştırılmıştır. Karşılaştırma metrikleri olarak hassasiyet, geri çağırma, F1 skor ve mAP50 değerleri seçilmiştir. Çalışma kapsamında literatürdeki sonuçlara yakın değerler elde edilmiştir. Özellikle geri çağırma metriğinde [25]'ten daha iyi bir sonuç elde edilmiştir. Sonuçlar değerlendirilirken, çalışma kapsamında kullanılan donanımın kısıtı göz önünde bulundurulmalıdır. Eğitim parametreleri bu donanıma bağlı olarak belirlenmiştir. Karşılaştırmalı sonuçlar Çizelge 4'te sunulmuştur.

Çizelge 4. Çalışma sonuçlarının literatür ile karşılaştırılması.

Çalışma	Hassasiyet	Geri çağırma	F1 skor	mAP50
YOLOv7 [23]	0.850	0.750	0.800	0.820
YOLOv7 [24]	0.845	0.835	0.840	0.820
YOLOv8s-CA3SC [25]	0.944	0.731	0.824	-
YOLOv5s*	0.805	0.765	0.784	0.780
YOLOv5x*	0.860	0.752	0.802	0.810
YOLOv5n*	0.809	0.764	0.782	0.802
YOLOv5m*	0.833	0.757	0.793	0.797
YOLOv5l*	0.852	0.745	0.795	0.785
YOLOv8s*	0.806	0.759	0.782	0.782
YOLOv9s*	0.885	0.776	0.827	0.836

* Bu çalışmada kullanılan yöntemler.

6. Sonuç

Bu çalışmada yedi farklı YOLO versiyonu ile İHA görüntülerinden beton yüzeyindeki çatlaklar derin öğrenme teknikleriyle tespit edilmiştir. Derin öğrenme yöntemleri otomatik çatlak tespiti uygulamalarında iş gücü, zaman ve maliyet bakımından büyük avantaj sağlamaktadır. Bu sebeple, en uygun yöntemin ve parametrelerin seçilmesi çatlak tespiti işlemlerinde yapı ve çevre sağlığını oluşturma ve geliştirme kapsamında büyük öneme sahiptir. İHA görüntüleri bu çatlakların otomatik denetlenmesi için zengin bir veri kaynağı oluşturmaktadır. Beton yüzeylerdeki çatlakların tespitinde en yüksek metrikler YOLOv9s modelinde elde edilmiştir. YOLO algoritmalarının çatlak tespitinde göstermiş olduğu yüksek performans çatlak tespitinin hızlı ve yüksek doğrulukla yapılmasına büyük oranda katkı sağlamaktadır. Ancak veri setlerinin içerdiği görüntülerin detay seviyesi, içerdiği nesnelerin boyutları ve algılanabilirlik özellikleri veri setlerinde testi gerçekleştirilen modellerden elde edilen değerlere büyük oranda etki etmektedir. Modellerin sahip olduğu parametre sayısı ve boyut genişliği de test aşamasındaki performanslarını önemli ölçüde etkilemektedir. Bu kısıtların model ve testlerdeki performanslara olan etkileri göz önünde bulundurularak analizler yapılmalı ve en uygun model ve parametreler seçilmelidir. Beton yüzeyi çatlaklarının tespiti çalışmalarında kullanılacak modellerin birtakım optimizasyon ve transformatörlerle iyileştirilmesi ve geliştirilmesi modellerin performanslarına pozitif yönde katkı sağlayabileceği öngörülmektedir.

Çalışma özellikle beton yüzeylerindeki deformasyonların görüntüler üzerinden izlenmesi için umut verici sonuçlara sahiptir. Ayrıca deprem sonrası hasar tespitinde bina taşıyıcı elemanlarının görüntüleri YOLO algoritmaları ile incelenebilir. Böylece hasar tespitinde otomasyon artırılarak insan gücüne olan ihtiyaç azaltılabilir. Karar vericilere ulaştırılacak hızlı sonuçlar sayesinde afet yönetim sistemi çerçevesinde müdahale ve iyileştirme çalışmalarında etkin bir çözüm üretilmesine destek sağlanabilir. Sadece beton yüzeyleri değil asfalt gibi farklı malzeme türlerinde de uygulanabilirlik potansiyeline sahiptir. Özellikle İHA görüntüleri ile geniş alanlarda yol ve kaldırımlardaki bozulmalar otomatik şekilde tespit edilebilir. Gelecek çalışmalarda gerçek zamanlı olarak veri setlerindeki görüntülerin nesne tespitlerini gerçekleştirmek için videolar üzerinden anlık tespit gerçekleştirilmesi planlanmaktadır. Bu çatlak veri setlerinin olası deprem sonrası çalışmalarda kullanılmak üzere geliştirilmesi ve iyileştirilmesi ile birlikte geliştirilmiş derin öğrenme tekniklerinin kullanımıyla ilgili çalışmaların kullanım yelpazesi genişletilebilmektedir. Veri setleri, kullanılan derin öğrenme algoritmaları, çalışma alanı ve koşullarının çeşitlendirilmesi ve iyileştirilmesiyle birlikte daha yüksek performans ve doğrulukta sonuçların elde edileceği öngörülmektedir.

Çıkar Çatışması Beyanı

Makale yazarları herhangi bir kurum, kuruluş, kişi ile kişisel ve finansal çıkar çatışması olmadığını beyan etmektedirler.

Kaynaklar

- [1] Woo HJ, Hong WH, Oh J, Baek, SC. Defining Structural Cracks in Exterior Walls of Concrete Buildings Using an Unmanned Aerial Vehicle, Drones 2023; 7(3):149.
- [2] Rahai A, Rahai M, Iraniparast M, Ghatee M. Surface Crack Detection using Deep Convolutional Neural Network in Concrete Structures. In: 2022 IEEE 5th International Conference on Image Processing Applications and Systems (IPAS); 2022; Genova, Italy. doi: 10.1109/IPAS55744.2022.10052790.
- [3] Chen X, Liu C, Chen L, Zhu X, Zhang Y, Wang C. A Pavement Crack Detection and Evaluation Framework for a UAV Inspection System Based Sciences. Appl. Sci. 2024; 14(3): 1157.
- [4] Redmon J, Divvala S, Girshick R, Farhadi A. You only look once: Unified, real-time object detection, In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2016; 779-788.
- [5] Girshick, R. Fast r-cnn. In Proceedings of the IEEE international conference on computer vision; 2015; 1440-1448.
- [6] He K, Gkioxari G, Dollár P, Girshick R. Mask r-cnn. In: Proceedings of the IEEE international conference on computer vision; 2017; 2961-2969.

- [7] Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu CY, Berg AC. Ssd: Single shot multibox detector. In: Computer Vision ECCV 2016: 14th European Conference; 2016; Amsterdam, The Netherlands.
- [8] Qiu Q, Lau D. Real-time detection of cracks in tiled sidewalks using YOLO-based method applied to unmanned aerial vehicle (UAV) images. *Autom Constr* 2023; 147: 104745.
- [9] Xu X, Zhao M, Shi P, Ren R, He X, Wei X, Yang H. Crack Detection and Comparison Study Based on Faster R-CNN and Mask R CNN. *Sensors* 2022; 22(3):1215.
- [10] Xing J, Liu Y, Zhang GZ. Improved YOLOV5-Based UAV Pavement Crack Detection. *IEEE Sens J* 2023; 23(14):15901-15909.
- [11] Tang Z, Chamchong R, Pawara P. A Comparison of Road Damage Detection Based on YOLOv8. In: 2023 International Conference on Machine Learning and Cybernetics (ICMLC); 2023; Adelaide, Australia.
- [12] Silva WRLD, Lucena DSD. Concrete cracks detection based on deep learning image classification. *Proceedings* 2018; 2(8): 489.
- [13] Yang C, Chen J, Li Z, Huang Y. Structural Crack Detection and Recognition Based on Deep Learning, *Appl. Sci.* 2021; 11(6): 1-13. <https://doi.org/10.3390/app11062868>
- [14] Dataset available online: <https://universe.roboflow.com/university-bswxt/crack-bphdr/dataset/1> (Erişim Tarihi: 07.05.2025)
- [15] Cepni S, Atik ME, Duran Z. Vehicle detection using different deep learning algorithms from image sequence, *Balt. J. Mod. Comput.* 2020; 8(2): 347-358.
- [16] Ultralytics. YOLOv5, Available online: <https://github.com/ultralytics/yolov5> (Erişim tarihi: 29.02.2024)
- [17] Zhang Y, Huang J, Cai F. On Bridge Surface Crack Detection Based on an Improved YOLOv3 Algorithm, *IFAC PapersOnLine* 2020; 53(2): 8205-8210. <https://doi.org/10.1016/j.ifacol.2020.12.1994>
- [18] Li X, Li L, Liu Z, Peng Z, Liu S, Zhou S, Chai X, Jiang K. Dam Crack Detection Studies by UAV Based on YOLO Algorithm. In: 2023 2nd International Conference on Robotics, Artificial Intelligence and Intelligent Control (RAIIC); 2023; Mianyang, China. doi: 10.1109/RAIIC59453.2023.10281120.
- [19] Gupta P, Dixit M. Performance Evaluation of Deep Learning Models For Surface Crack Detection. In: 1st International Conference on Innovations in High Speed Communication and Signal Processing (IHCSP); 2023; Bhopal, India. doi: 10.1109/IHCSP56702.2023.10127175.
- [20] Biyik MY, Atik ME, Duran Z. Deep learning-based vehicle detection from orthophoto and spatial accuracy analysis, *INT J ENG GEOSCI* 2023; 8(2): 138-145.
- [21] Jocher G, Chaurasia A, Qiu J. Ultralytics YOLO (Version 8.0.0) [Computer software]. Available at: <https://github.com/ultralytics/ultralytics> 2023; (Erişim tarihi: 23.09.2024)
- [22] Nusari ANM, Ozbek IY, Oral EA. Automatic Vehicle Accident Detection and Classification from Images: A Comparison of YOLOv9 and YOLO-NAS Algorithms. In: 32nd Signal Processing and Communications Applications Conference (SIU); 2024; Mersin, Türkiye. doi: 10.1109/SIU61531.2024.10600761.
- [23] Schetakakis N, Koutmos V, Papoutsakis N, Stavrakakis K, Stavroulakis GE, Stavrakakis G. Towards geometric digital twins, including damage detection, from photos of residential buildings facades. In: 2024 IEEE International Workshop on Metrology for Living Environment (MetroLivEnv); 2024; Chania, Greece. doi: 10.1109/MetroLivEnv60384.2024.10615756.
- [24] Panduman YYF, Funabiki N, Sukaridhoto S. An Idea of Drone-Based Building Crack Detection System in SEMAR IoT Server Platform. In: IEEE 12th Global Conference on Consumer Electronics; 2023; Nara, Japan. doi: 10.1109/GCCE59613.2023.10315417.
- [25] Hu W, Qin J, Liang S, Jiang Q. A method for concrete crack detection based on improved YOLOv8s. In: Fourth International Conference on Green Communication, Network, and Internet of Things; 2024; Guiyang, China. doi: 10.1117/12.3052752.