

GERİ YAYILIMLI BİRLİKTE EVRİM İLE İYİLEŞTİRİLMİŞ DERİN SİNİR AĞLARI KULLANILARAK YOL ÇATLAK TESPİTİ

Emirhan Mustafa ANIK *^{ID}

Turan ARSLAN **^{ID}

Alınma: 25.04.2024; düzeltme: 22.07.2024; kabul: 02.08.2024

Öz: Karayolu esnek üstyapılarındaki çatlaklar genellikle trafik yükleri ve hava koşullarından kaynaklanır. Bu çatlakların genişlemeden tespit edilip gerekli bakımlarının yapılması, yol konforunun sürekliliğini sağlamanın yanı sıra bakım maliyetlerini de azaltacaktır. Bu çalışma, yoldaki çatlakları gerçek zamanlı ve yüksek doğrulukla tespit etmeyi amaçlamaktadır. Bu bağlamda, Geri Yayimli Birlikte Evrim yaklaşımıyla İyileştirilmiş Derin Sinir Ağları ve görüntü işleme yöntemleri birlikte kullanılmıştır. Ayrıca, çeşitli sayı ve çözünürlüklerde çatlaklı görsel veriler içeren EdmCrack600, AsphaltCrack, CFD ve CrackSegmentation veri setleri kullanılarak yeni bir veri seti oluşturulmuş ve bu veri seti üzerinde Derin Sinir Ağları tabanlı öğrenme gerçekleştirilmiştir. Modelin doğruluğu, CFD veri seti kullanılarak Kesinlik, Duyarlılık ve F1-Skoru ile değerlendirilmiştir. Değerlendirme sonucunda, önerilen yöntemin saniyede 48 görsel üzerinde çatlak tespit edebildiği ve %92,74 Kesinlik, %88,92 Duyarlılık ve %89,61 F1-Skoru başarı oranlarına ulaştığı gözlemlenmiştir.

Anahtar Kelimeler: Çatlak tespiti, Geri yayımlı birlikte evrim, Derin sinir ağları, Görüntü işleme, Esnek Üstyapı

Road Crack Detection Using Deep Neural Networks Developed via Cooperative Coevolution with Backpropagation

Abstract: Cracks in highway flexible pavements are primarily caused by traffic loads and weather conditions. Detecting these cracks before they expand and conducting necessary maintenance will not only ensure the continuity of road comfort but will also reduce maintenance costs. This study aims to detect cracks on the road in real-time and with high accuracy. In this context, Deep Neural Networks Developed via Cooperative Coevolution with Backpropagation and image processing methods were used together. Moreover, a new data set was obtained by using EdmCrack600, AsphaltCrack, CFD, and CrackSegmentation datasets containing cracked visual data in various numbers and resolutions, and Deep Neural Networks-based learning was performed on this dataset. The accuracy of the model was evaluated with Precision, Recall, and F1-Score using the CFD dataset. As a result of the evaluation, it has been observed that the proposed method can detect cracks on 48 images per second, while it can reach 92.74% Precision, 88.92% Recall, and 89.61% F1-Score success rates.

Keywords: Road crack detection, Cooperative coevolution with backpropagation, Deep learning, Image processing, Flexible Pavement

1. GİRİŞ

Havaalanları, karayolları ve belediyeler, yol çatlağı tespiti konusunda çeşitli çalışmalar yürütmektedir. Havaalanları genellikle yüksek hassasiyet gerektiren alanlardır ve bu nedenle daha ileri teknolojilere yatırım yapmaktadır. Ancak, bu yüksek teknoloji çözümleri maliyetli olabilmekte ve sürekli bakım gerektirmektedir. Karayolları ve belediyeler ise geniş yol ağlarını

* emirhanik16@gmail.com – Bursa Uludağ Üniversitesi İnşaat Mühendisliği Bölümü

** arsltur@gmail.com - Bursa Uludağ Üniversitesi İnşaat Mühendisliği Bölümü

İletişim Yazarı: Emirhan Mustafa ANIK (emirhanik16@gmail.com)

yönetmek zorundadır, bu nedenle etkin ve ölçeklenebilir çözümler aramaktadırlar. Manuel incelemeler yaygın olmakla birlikte, otomatik sistemler kullanılarak verimliliği artırmak ve hata oranlarını azaltmak hedeflenmektedir. Ancak, bu sistemler genellikle yüksek maliyetli ve karmaşık altyapılar gerektirmektedir (Vyas ve diğ., 2019).

Diğer yandan yol bakımı özellikle de çatlak tespiti açısından, ulaşım altyapı yönetimi kapsamında kritik bir görev üstlenmektedir. Dolayısıyla birçok bölgede hâlâ yaygın olarak manuel inceleme yöntemleri uygulanmaktadır. Ancak manuel incelemeler zahmetli, zaman alıcı ve hatalara açık olmaktadır. Bu tür hataların yansımaları ise hem sürücüler için güvenlik tehlikeleri yaratmakta hem de bakım maliyetlerini artırmaktadır (Staniek, 2017).

Manuel denetim ve geleneksel tespit yöntemlerinin eksikliklerinin farkına varan araştırmacılar, özellikle görüntü sensörleri ve bilgisayarla görme algoritmalarından yararlanan otomatik tespit yöntemlerine giderek daha fazla yönelmiştir (Munawar ve diğ., 2021). Bu metodolojiler çatlak tespit süreçlerini kolaylaştırmayı ve manuel denetim ve geleneksel tekniklerle ilişkili sınırlamaları azaltmayı amaçlamıştır.

Bilgisayarla görmeye dayalı yöntemler, çatlakları tespit etmek amacıyla görüntülerden özellik çıkarımı için çeşitli algoritmalar kullanmıştır. Bunlar arasında eşikleme (Oliveira ve Correia, 2009; Peng ve diğ., 2015; Tsai ve diğ., 2010), kenar algılama (Oliveira ve Correia, 2014; Santhi ve diğ., 2012; Yeum ve Dyke, 2015), wavelet (dalgalık) dönüşümü (Zhou, 2006) ve mini yol seçimi (Amhaz ve diğ., 2016; Zou ve diğ., 2016) gibi ilk teknikler yer almaktadır. Makine öğrenimi önem kazandıkça, araştırmacılar yapay sinir ağları (Lee ve Lee, 2004; Moon ve Kim, 2011), destek vektör makineleri (Y.-J. Cha ve diğ., 2016; Gavilán ve diğ., 2011; O'Byrne ve diğ., 2013) ve rastgele orman algoritması (Shi ve diğ., 2016) gibi farklı yöntemleri kullanarak çatlak tespiti uygulamaya başlamıştır.

Son yıllarda, Derin Sinir Ağlarının (DSA), özellikle de Evrişimli Sinir Ağları (ESA), otomatik yol çatlakı tespitinde devrim yaratmıştır. Manuel özellik çıkarımına dayanan geleneksel yöntemlerin aksine, DSA geri yayılım yoluyla sinir katmanı parametrelerini otomatik olarak güncellemek için büyük miktarda yol verisinden yararlanmaktadır. Bu veri odaklı algoritmalar genel olarak görüntü sınıflandırma, nesne algılama ve semantik segmentasyon ağları olarak kategorize edilmektedir. Görüntü sınıflandırma ağları (Krizhevsky ve diğ., 2017) pozitif (çatlak) ve negatif (çatlak olmayan) yol görüntüleri arasında ayırım yaparken (J. Fan ve diğ., 2021), nesne algılama ağları örnek düzeyinde yol çatlaklarını tanımlamaktadır (Du ve diğ., 2021). Semantik segmentasyon ağları (Chen ve diğ., 2020; Dung ve Anh, 2019; Huyan ve diğ., 2020; J. Liu ve diğ., 2020; Qu ve diğ., 2022; Zhang ve diğ., 2022; Zhu ve diğ., 2023) piksel bazında çatlak tespit sonuçları elde etmekte ve hassasiyetleri nedeniyle giderek daha fazla tercih edilmektedir. Ancak bu piksel bazlı tespit algoritmaları büyük ölçüde denetimli öğrenmeye dayanmakta, yoğun emek gerektiren ek açıklama süreçleri gerektirmekte ve farklı senaryolarda sınırlı genelleştirilebilirlik göstermektedir.

Genel olarak yapılan çalışmalara bakıldığında araştırmacılar tarafından çok çeşitli çatlak tespit yöntemlerinin önerildiği gözlenmektedir. Ancak çoğunlukla sunulan yöntemlerde çatlak tespitinde sadece yüksek doğruluğa odaklanılmıştır (Doğan ve Ergen, 2022; Fei ve diğ., 2020; Gao ve diğ., 2023; Mei ve Gül, 2020). Diğer taraftan gerçek zamanlı çatlak tespitini amaçlayan yöntemlerde mevcuttur. Önerilmiş gerçek zamanlı çatlak tespit yöntemlerine bakıldığında ise genel olarak yüksek teknik donanımlar gerekmektedir (Mave diğ., 2022; Naddaf-Sh ve diğ., 2020; Wang ve Su, 2021; Zhang ve diğ., 2023) ve bu tür donanımların gerçek hayatta kullanılabilmesi için büyük bütçeler harcanmalıdır.

Bu çalışmada, esnek üstyapı üzerinde maliyeti olabildiğince azaltmak amacıyla gerçek zamanlı yüksek doğrulukta çatlak tespiti yaparken minimum donanım gerektiren bir model tasarlanmaya çalışılmıştır. Önerilen yöntemin mevcut yöntemlerden ayrılan yönlerine bakıldığında; ilk kez çatlak tespiti çalışmalarında Geri Yayılımlı Birlikte Evrim (GYBE) (Gong ve diğ., 2021) yöntemi uygulanmakta ve çatlak tespiti için önceden DSA kullanılarak elde edilmiş parametreler görüntü işleme teknikleriyle birlikte kullanılmaktadır. Ayrıca yapılan çalışmaların

aksine bu çalışmada gerekli donanım gücünü minimum da tutarak piksel bazlı çatlak tespiti yerine esnek üstyapılar üzerinde alan tabanlı çatlak tespiti yapılmaktadır.

2. MATERYAL ve METOT

2.1. Veri Seti

Bu çalışmada Tablo 1'de verilen EdmCrack600 (Mei ve Gül, 2020; Mei ve diğ., 2020), AsphaltCrack (Jayanth Balaji ve diğ., 2019), CFD (Cui ve diğ., 2015; Shi ve diğ., 2016) ve CrackSegmentation (K. Liu ve Chen, 2023; K. Liu ve diğ., 2019) veri setleri ilgili makalelerde verilen GitHub ve Mendeley adreslerinden erişilerek kullanılmıştır. Bu veri kümeleri kullanılarak 80 x 80 piksel boyutlarında 7500 tane çatlak içeren ve 7500 tane çatlak içermeyen görüntü oluşturulmuştur. Elde edilen bu görseller GYBE yönteminin performansını değerlendirmek için kullanılmıştır. Önerilen modelin performansı ise CFD veri setinin eğitimde kullanılmayan 80 görseli kullanılarak değerlendirilmiştir.

Tablo 1. Çalışmada kullanılan veri setlerinin örnek sayısı ve çözünürlüğü

Veri Seti	Resim Sayısı	Çözünürlük
EdmCrack600	600	1920x1080
AsphaltCrack	200	448x448
CrackForest	155	480x320
CrackSegmentation	428	448x448

2.2. GYBE

Geri yayılım-gradyan iniş (GY) (Rumelhart ve McClelland, 1987) ve Birlikte Evrim tabanlı optimizasyon (BE) (Potter ve Jong, 1994), DSA parametrelerinin öğrenme yöntemleridir ve her ikisinin de zayıf ve güçlü olduğu özellikleri vardır. GY iyi bir hesaplama verimliliğine sahiptir, ancak başlatma hassasiyeti konusunda bazı sorunlarla karşılaşmakta ve düşük yerel optimumlara takılma eğilimindedir. Diğer taraftan, BE gradyan inişi içermez ve bu nedenle hesaplama için çok fazla zaman harcamaktadır. Ancak BE, GY'nin maruz kaldığı sorunlardan kaçınılabilir. Araştırmacılar, GY ve BE'nin maruz kaldığı öğrenme sorunlarını gidermek için GY ve BE'yi birleştiren GYBE modelini önermişlerdir.

GY algoritması, DSA'larının parametrelerini öğrenmek için yaygın bir gradyan iniş tabanlı yöntemdir. GY, öğrenilen çıktılar ile gerçek çıktılar arasındaki hataları Eş. 1 kullanarak hesaplar ve hesaplanan hatalar kullanılarak ağ parametreleri güncellenir. Burada D eğitim verisini, x_k girdi değerlerini, y bilinen çıktı değerlerini, W bir DSA'daki tüm bağlantı ağırlıklarını, β ise düzenleme terimini temsil etmektedir.

$$F(P) = \frac{2}{2|D|} \sum_{(x,y) \in D} \|o^L(x_k) - y\|_2^2 + \beta \|W\|_2 \quad (1)$$

BE bir tür Evrimsel Algoritmadır (EA) (Xin Yao, 1999). EA rastgele bir sürece dayanır ve sınırlı bilgi içeren problemlerin üstesinden gelmek için geliştirilmiştir. BE algoritması, N boyutlu bir karar vektörünü L alt bileşene böler. Daha sonra, L alt bileşen EA ile ayrı ayrı geliştirilir ve bu, sonraki her işlemde arama alanını büyük ölçüde azaltır. EA tabanlı optimizasyon sistemi 3 temel aşamadan oluşmaktadır: Ayrıştırma, Seçim Stratejisi ve Optimizasyon Süreci.

1) Ayırıştırma: Ayırıştırma yöntemi tam olarak sayısal bir yöntem değildir ve sadece nöronları ayırma işlemidir. Bu yöntemde, her bir ağdaki her bir nöron kendi ağındaki nöronlardan bağımsız hale getirilir, böylece olgunluk durumu ayrı ayrı kontrol edilebilir.

2) Seçim Stratejisi: Öğrenme aşamasında bazı nöronlar yeterli seviyeye ulaşmış olmalarına rağmen öğrenmeye ve ağırlıklarını güncellemeye devam ederler. Ancak bu durum gereksiz zaman kaybına yol açacaktır. Dolayısıyla belli bir olgunluğa erişmiş nöronların öğrenmesinin durdurulması ve hala olgunlaşmamış nöronların öğrenmeye devam etmesi zaman maliyeti açısından son derece verimli olacaktır. Bu durumdan yola çıkarak Eş. 2 ile her bir alt görev için olgunluk derecesi, Eş. 3 ve Eş. 4 kullanılarak da nöronun olgunluğu hesaplanır. Burada, x_k girdi değerlerini, w_{ir} ağırlık değerlerini, b_r girdi değerlerini, l katman sayısını ve α eşik parametresini temsil etmektedir.

$$o_r^l(x_k) = A \left(\sum_{i=1}^n w_{ir}^{(l-1)l} \cdot o_i^{l-1}(x_k) + b_r^l \right) \quad (2)$$

$$G(s) = \begin{cases} 0, & \alpha \ll s \ll 1 - \alpha \\ 1, & \text{diğer koşullarda} \end{cases} \quad (3)$$

$$M_r = \sum_{i=1}^N G(o_i^{l-1}(x_k)) \quad (4)$$

3) Optimizasyon Süreci: Optimizasyonu sağlamak için, seçilen her alt görevin en uygun parametre değerlerine sahip olması gerekir. Bu gereksinimi karşılamak için optimum bir alt görev popülasyonuna ihtiyaç vardır. Araştırmacılar tarafından bu popülasyonu belirlemek için Diferansiyel Evrim Algoritmasının yüksek performanslı yöntemlerinden biri olan Kendi Kendine Uyarlanabilir Diferansiyel Evrim (KUDE) (Qin ve diğ., 2009) yöntemi kullanılmıştır. KUDE algoritması ile uygun alt görevler seçildikten sonra her bir alt görevin hata değerini hesaplamak için Eş. 5 kullanılmış ve Eş. 6 ile popülasyon sayısı kadar birey içeren bir popülasyon oluşturulmuştur. Burada D_s eğitim verisinden rastgele seçilmiş küçük alt kümeyi, σ_s ağırlık parametrelerini, NP ise popülasyon boyutunu temsil etmektedir ve bu çalışmada NP değeri 50 olarak belirlenmiştir.

$$F_s(P_{sub_r}) = \frac{1}{2|D_s| \sum_{(x,y) \in D_s} \|o^L(x_k) - y\|_2^2 + \beta \|W_{sub_r}\|_2 + 10^{-8} + \sigma_s M_r} \quad (5)$$

$$P_{sub_r}^{i_p}(j) = \begin{cases} P_{sub_r}^*(j) \cdot rand(0, 2), & i_p = 1, \dots, NP - 1 \\ P_{sub_r}^*(j), & i_p = NP \end{cases} \quad (6)$$

2.3. GYBE ile Öğrenme Süreci

Bu süreç bir dizi adımdan oluşmaktadır.

- Test ve eğitim veri setleri yazılıma aktarılmıştır.
- Görsel veri setlerine OpenCV kütüphanesinde yer alan eşik yöntemleri uygulanmıştır.
- Güncellenmiş veri kümeleri üzerinde GYBE kullanarak farklı ağ yapıları için gerekli ağ parametreleri öğrenilmiştir.
- Öğrenilmiş ağ parametrelerinin başarısını belirlemek için Kesinlik, Doğruluk ve F1-Score yöntemleri uygulanmıştır.
- Optimum ağ yapısı belirlenmiş ve ağın parametreleri pickle formatında dışa aktarılmıştır.

2.4. Değerlendirme Metrikleri

Eğitilen modeli değerlendirmek için yaygın olarak kullanılan Kesinlik (Pr), Duyarlılık (Re) ve F1-Score (F1) yöntemleri uygulanmıştır. Kesinlik ve Duyarlılık temel olarak Pozitif olarak tahmin edilen değerlerin gerçekten kaç adedinin Pozitif olduğunu Eş. 7 ve Eş. 8 kullanarak hesaplayan yöntemlerdir. F1-Score ise Kesinlik ve Duyarlılık değerlerinin harmonik ortalamasını olarak daha güvenilir bir başarı oranı veren bir yöntemdir ve Eş. 9 ile hesaplanmaktadır.

$$P_r = \frac{TP}{TP + FP} \quad (7)$$

$$R_e = \frac{TP}{TP + FN} \quad (8)$$

$$F1 = \frac{2 \times P_r \times R_e}{P_r + R_e} \quad (9)$$

Burada TP çatlak tespiti yapılan bölgede çatlak olduğu tahmin edilen ve gerçekte çatlak bulunduran, FP çatlak tespiti yapılan bölgede çatlak olduğu tahmin edilen ama gerçekte çatlak bulundurmuyan, FN ise çatlak tespiti yapılan bölgede çatlak olmadığı tahmin edilen ve gerçekte de çatlak bulundurmuyan tüm test örneklerinin sayısıdır.

2.5. Önerilen Yöntem

Bu çalışmada, yol üzerinde bulunan çatlakları hem görüntü işleme teknikleri hem de GYBE yöntemi kullanılarak gerçek zamanlı olarak tespit etmeyi amaçlayan bir model önerilmiştir.

Modelde, çatlak tespitini tüm görsel üzerinde yapmak yerine çatlak olma potansiyeli olan bölgelerde tespit işleminin yapılması hem zaman hem de başarı açısından katkı sağlayacağı düşünülmüştür. Bu nedenle ilk aşamada görsel üzerinde aşağıda belirtilen bir dizi görüntü işleme tekniği uygulanmıştır.

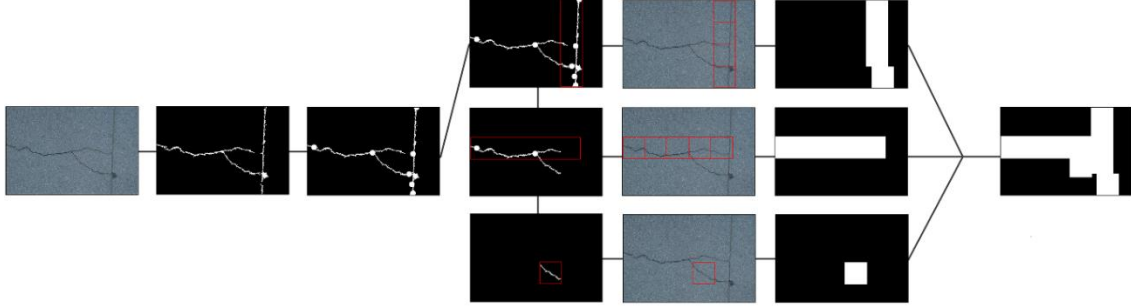
1) Görüntüdeki çatlak olmayan küçük nesnelerin kaldırılması: Çatlak tespiti yapılmak istenen görüntüde çatlak olmayan ancak çatlak gibi gösterilen nesnelerin bulunması muhtemeldir ve bu durum tespit işlemi sırasında çeşitli sorunlara yol açabilir. Bu nesneleri görüntüden çıkarmak için görüntülerdeki nesnelerin geometrik özellikleri kullanılmış ve küçük nesneler görüntüden silinmiştir.

2) Görüntüdeki çatlakların birleştirilmesi: Görüntülerde çatlaklar arasındaki boşluk görünmemesine rağmen bilgisayar ortamında çatlaklar arasındaki boşluklar tespit edilebilmektedir. Bu durum çatlakların birbirleri ile bağlantılı olmamasına ve farklı nesneler olarak algılanmasına neden olmaktadır. Farklı olarak algılanmayı ortadan kaldırmak ve tespit aşamasındaki hızı ve doğruluğu artırmak için bu nesneler birleştirilmiştir.

3) Görüntüdeki büyük çatlak nesnelerinin silinmesi: Çatlak tespitine yönelik görüntülerde genellikle küçük boyutlu çatlak nesneleri tespit edilmekte ve bunlar önceki aşamalarda kullanılan yöntemler kullanılarak temizlenebilmektedir. Ancak bu yöntemler sadece küçük nesneleri temizleyebilmekte, nispeten daha büyük nesneler çatlak nesnesi olarak algılandığından aynı kalmaktadır. Bu büyük boyutlu çatlak olmayan nesneleri ortadan kaldırmak için nesnelerin geometrik özellikleri kullanılmıştır.

Görüntü işleme teknikleri uygulanan görselde çatlak olma ihtimali bulunmayan nesneler görüntüden çıkarılmıştır. Ardından sadece çatlak içeren nesneler parçalanarak 80x80 piksel

boyutlarında alanlara ayrılmıştır. Böylece GYBE yöntemi ile öğrendiğimiz veriler bu alanlarda kullanılarak Şekil 1’de gösterildiği gibi çatlak tespiti yapılabilmektedir.



Şekil 1:
Çatlak bölgeleri belirlenmiş alanlarda çatlak tespitinin yapılması

3. BULGULAR VE TARTIŞMA

Bu çalışmada, GYBE yönteminin eğitimi 12000 adet 80x80 piksel çözünürlükte görsel üzerinde 6400-200-2 ağ yapısı ile yapılmıştır. Önerilen yöntemin performansı ise eğitimde kullanılmayan CFD veri setinin 80 görseli kullanılarak değerlendirilmiş ve çatlak alanları tespit edilen bazı görseller Şekil 2’de gösterilmiştir. Eğitim ve test sırasındaki tüm deneyler NVIDIA GeForce GTX 1650 GPU ve Intel i7-10750H @2.60GHz CPU (6 Çekirdek 12 Mantıksal İşlemci) bulunduran bir bilgisayarda gerçekleştirilmiştir.

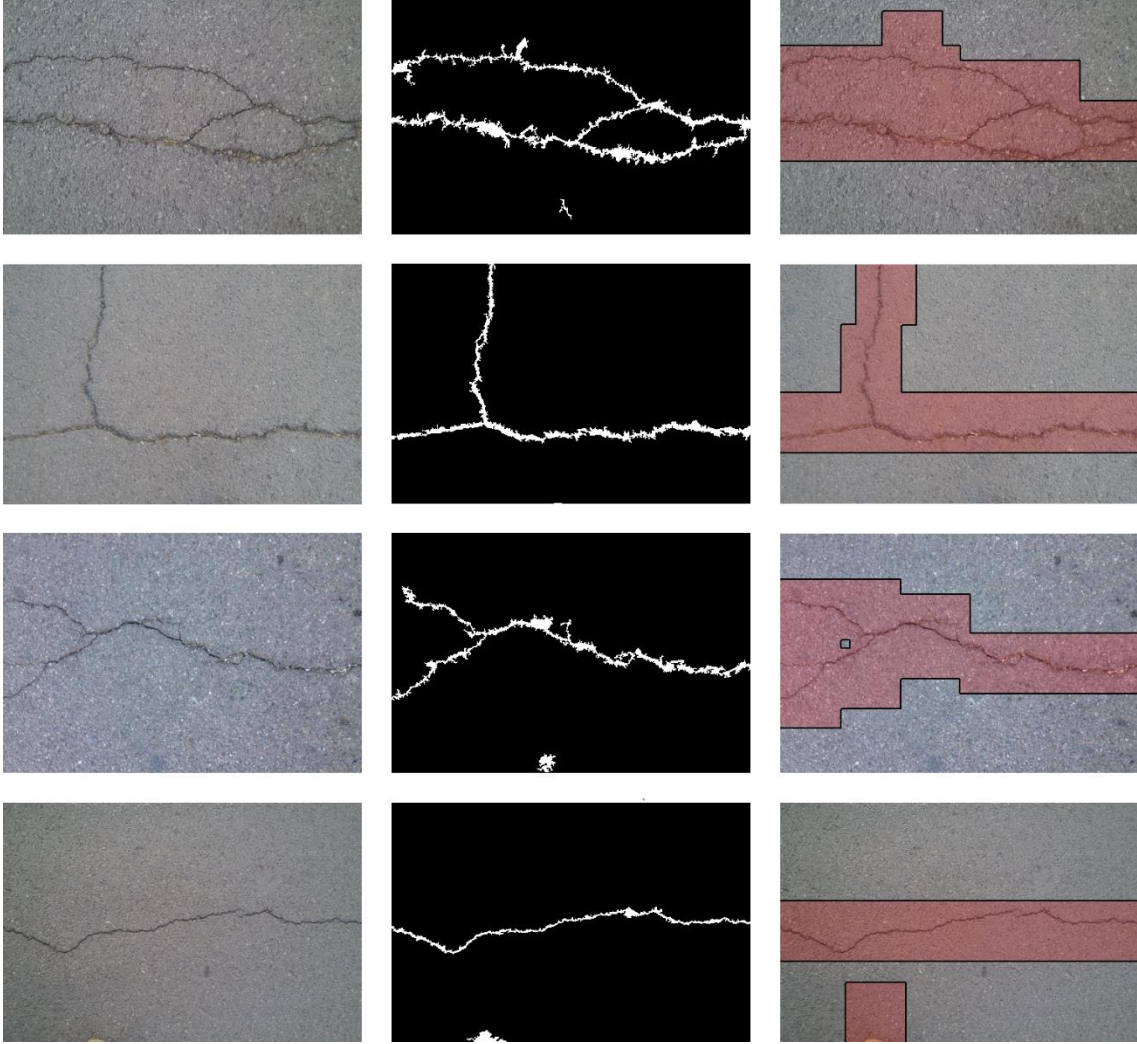
CFD veri seti üzerinde deneysel sonuçları bazı başarılı çalışmalarla karşılaştırılmıştır. Tablo 2’ye bakıldığında gerçek zamanlı tespiti amaçlamayan yöntemler ile önerdiğimiz yöntemin Kesinlik, Doğruluk ve F1-Score açısından karşılaştırılması gösterilmektedir.

Tablo 2. CFD veri seti kullanarak çatlak tespiti yapan yöntemler ve tespit başarıları

Yöntemler	Kesinlik	Duyarlılık	F1-Score
CrackTree	%73,22	%76,45	%70,80
CrackForest	%82,28	%89,44	%85,71
FFA	%78,56	%68,43	%73,15
CrackNet-V	%92,58	%86,03	%89,18
ConnCrack	%96,79	%87,75	%91,96
Parallel ResNet	%96,21	%95,12	%95,63
Önerilen Model	%92,74	%88,92	%89,61

Burada CrackTree (Zou ve diğ., 2012) minimum kapsamlı ağaç yapısına dayanan bir yöntem iken CrackForest (Shi ve diğ., 2016) rastgele yapılandırılmış ağaç modelini temel almaktadır. FFA (Nguyen ve diğerleri, 2011) ise çatlak tespiti için her türlü serbest biçimli ve yol boyunca hesaplanan özellikleri kullanmaktadır. CrackNet-V (Fei ve diğ., 2020) yönteminde girdilerin boyutunu yedeklemek ve çatlakları tespit etmek için havuzlama katmanı kullanmayan bir yapı önerilmektedir. ConnCrack (Mei ve Gül, 2020) metodu koşullu Wasserstein Çekişmeli Üretici

Ağ ile bağlantı haritaları tekniğini birleştirmektedir. Paralel ResNet (Z. Fan ve diğ., 2022) ise yüksek performanslı çatlak tespiti ve ölçüm sistemi oluşturmayı amaçlayan yeni bir derin artık evrişimli sinir ağı yöntemidir.



Şekil 2:

Çatlak tespiti yapılan görsellerinin sırasıyla orijinal, çatlak olmayan nesnelere arındırılmış ve çatlak tespit edilen bölgeleri belirtilmiş halleri

Tablo 2’de görüldüğü üzere önerilen yöntem CrackTree ve FFA yöntemlerinden daha iyi başarıya sahipken önerdiğimiz yöntem CrackNet-V ve CrackForest yöntemiyle benzer derecede başarıya ulaşabilmiştir. ConnCrack ile Paralel ResNet yöntemlerinin ise genel olarak tespiti doğruluğu açısından daha başarılı bir yöntem oldukları açıkça görülmektedir.

Diğer taraftan yukarıda verilen yöntemler sadece çatlak tespitinde başarıya odaklanmış ve çatlak alanından ziyade doğrudan çatlak tespit etmeye çalışan dolayısıyla farklı başarı değerlendirme yolları izleyen metotlardır. Bu nedenle çalışmanın başarısını daha iyi anlayabilmek amacıyla gerçek zamanlı çatlak tespiti yapmaya amaçlayan yöntemler ile de karşılaştırma yapılması gerektiğine inanılmıştır. Bu nedenle önerilen yöntemin F1-Score başarı değerleri ve maksimum FBS değerleri Tablo 3’te gösterilen bazı başarılı yöntemlerin değerleriyle karşılaştırılmıştır.

Burada EfficientDetDD (Naddaf-Sh ve diğ., 2020) yol kusurlarını gerçek zamanlı olarak tespit etmek amacıyla ölçeklenebilir ve verimli modeller eğitmek için derin öğrenme tabanlı bir şemayken DenseASPP (Yang ve diğ., 2018) yoğun bir şekilde çok ölçekli özellik üreten bir dizi atröz evrişimli katmanları birbirine bağlayan bir yöntemdir. SegNet (Badrinarayanan ve diğ., 2017) ise semantik piksel bazında segmentasyon için derin tamamen evrişimli bir sinir ağı mimarisi kullanılmaktadır. Son olarak, DLCS (Wang ve Su, 2021) ikili segmentasyon ağına dayalı yüzeysel bir çatlak segmentasyon modeliyken MFLCD (Ma ve diğ., 2022) birden çok özellikli katmanına sahip evrişimli sinir ağına dayalı bir çatlak tespit sistemidir.

Tablo 3. CFD veri seti kullanarak gerçek zamanlı çatlak tespiti yapan yöntemler ve FBS değerleri

Yöntemler	F1-Score	FBS
EfficientDetDD	%56,00	180
DenseASPP	%81,71	14,2
SegNet	%81,91	15,5
DLCS	%82,70	46,4
MFLCD	%98,22	96,6
Önerilen Model	%88,97	47,7

Tablo 3'e bakıldığında önerdiğimiz yöntem DenseASPP, SegNet ve DLCS yöntemlerinden başarılı olsa da EfficientDetDD yönteminden FBS açısından geri kalmaktadır. MFLCD yönteminin ise F1-Score açısından açık bir şekilde en başarılı yöntem olduğu görülmektedir.

4. SONUÇ

Bu çalışmada GYBE yöntemi ile görüntü işleme teknikleri kullanılarak yeni bir çatlak tespiti yöntemi önerilmiştir. Bu kapsamda çatlak görsel verileri sistem içerisinde tanımlanarak çatlak tespitinin yapılması öğretilmiş ve öğrenen model kullanılarak yeni görsel veriler üzerinde çatlak tespiti yapılabilmektedir. Ancak model ile sadece 80 x 80 piksel boyutlarına sahip görsellerde çatlak olup olmadığı belirlenmektedir. Bunun yanında çatlak tespiti yapılmak istenen görseller çok daha büyük boyutlardadır ve görseller içerisinde çatlağın konumu tam olarak bilinmemektedir. Bu amaçla ilk başta görsel içerisinde yer alan çok küçük boyutlarda çatlak olmayan nesnelere temizlenmekte ve nesnelere içerisinde yer alan çok küçük boşluklar doldurulmaktadır. Bu işlemin ardından küçük boyutta bulunan nesnelere, veri setindeki görsellerde yer alan nesnelere geometrik özelliklerinden yararlanarak görsel içerisinden çıkarılmaktadır. Bu işleme takiben birbirine oldukça yakın olan çatlak nesnelere birleştirilmekte ve büyük boyutlu çatlak içermeyen nesnelere çeşitli geometrik özellik değerleri kullanılarak silinmektedir. Son olarak da görsel içerisinde yer alan nesnelere bölgelere parçalanarak taranacak alanlar belirlenmekte ve belirlenen alanlarda GYBE tekniğiyle öğrenilen parametreler kullanılarak çatlak tespiti yapılmaktadır. Çalışmanın ana katkıları ise aşağıdaki gibidir:

1) Yapılacak çalışmalarda kullanılacak farklı veri setlerinden elde edilen 80x80 piksel çözünürlüğe sahip 7500 adet çatlak içeren 7500 adet içermeyen görseller oluşturulmuştur.

2) Daha önce çatlak tespitinde kullanılmamış BPCC yöntemini ile görüntü işleme tekniklerini birlikte kullanılan ve tespit yapılacak görselin tüm alanı yerine resim içerisinde çatlak olabilecek nesnelere üzerinde alan tabanlı çatlak tespiti yapan bir model önerilmiştir. Başarı

değerlendirmesinde ise ulaştığı %92,74 Kesinlik, %88,92 Duyarlılık ve %89,61 F1-Score değerleriyle güncel yöntemlere oldukça yakın başarı elde ettiği görülmüştür.

3) Modelin sadece çatlak olabilecek yerlerde tespiti yapması performans açısından avantaj sağlamış ve maksimum saniye 48 görsel üzerinde çatlak tespiti yapabilesiyle saha çalışmalarında da kullanabileceğini kanıtlamıştır.

Genel olarak bu çalışma kapsamında yol çatlak tespitinde zaman ve başarı açısından performans artırabilecek bir model önerilmiştir. Gelecekte, GPS ve kamera verilerini işleyebilen bir cihazın şehrin çoğu bölgesine seyahat eden kamu ya da özel araçlarında önerilen yöntemi kullanarak konumları bilinen çatlak görsel verileri elde edilebilir. Böylece hem şehrin tüm bölgesindeki esnek üstyapı üzerinde çatlak durumu gözlenebilir ve analiz edilebilir hem de ek zaman veya para harcamadan güncel yol görsel verileri üretilebilir.

ÇIKAR ÇATIŞMASI

Emirhan Mustafa ANIK ve Turan ARSLAN bilinen herhangi bir çıkar çatışması veya herhangi bir kurum/kuruluş ya da kişi ile ortak çıkar bulunmadığını onaylamaktadırlar.

YAZAR KATKISI

Emirhan Mustafa ANIK çalışmanın kavramsal ve tasarım süreçlerinin belirlenmesi ve yönetimi, veri toplama, analizi ve yorumlama, makale taslağının oluşturulması ve fikrinsel içeriğin eleştirel incelenmesi süreçlerinde, Turan ARSLAN çalışmanın kavramsal ve tasarım süreçlerinin belirlenmesi ve yönetimi, veri analizi ve yorumlama ve fikrinsel içeriğin eleştirel incelenmesi süreçlerinde katkı sağlamıştır.

KAYNAKLAR

1. Amhaz, R., Chambon, S., Idier, J. ve Baltazart, V. (2016) Automatic Crack Detection on Two-Dimensional Pavement Images: An Algorithm Based on Minimal Path Selection, IEEE Transactions on Intelligent Transportation Systems, 17(10), 2718-2729. doi:10.1109/TITS.2015.2477675
2. Badrinarayanan, V., Kendall, A. ve Cipolla, R. (2017) SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation, IEEE Transactions on Pattern Analysis and Machine Intelligence, 39(12), 2481-2495. doi:10.1109/TPAMI.2016.2644615
3. Cha, Y., Choi, W., Suh, G., Mahmoudkhani, S. ve Büyüköztürk, O. (2018) Autonomous Structural Visual Inspection Using Region-Based Deep Learning for Detecting Multiple Damage Types, Computer-Aided Civil and Infrastructure Engineering, 33(9), 731-747. doi:10.1111/mice.12334
4. Cha, Y.-J., You, K. ve Choi, W. (2016) Vision-based detection of loosened bolts using the Hough transform and support vector machines, Automation in Construction, 71, 181-188. doi:10.1016/j.autcon.2016.06.008
5. Chen, T., Cai, Z., Zhao, X., Chen, C., Liang, X., Zou, T. ve Wang, P. (2020) Pavement crack detection and recognition using the architecture of segNet, Journal of Industrial Information Integration, 18, 100144. doi:10.1016/j.jii.2020.100144
6. Cui, L., Qi, Z., Chen, Z., Meng, F. ve Shi, Y. (2015) Pavement Distress Detection Using Random Decision Forests (ss. 95-102). doi:10.1007/978-3-319-24474-7_14
7. Doğan, G. ve Ergen, B. (2022) Karayollarındaki Asfalt Çatlaklarının Tespiti İçin Yeni Bir Konvolüsyonel Sinir Ağı Tabanlı Yöntem, Fırat Üniversitesi Mühendislik Bilimleri Dergisi, 34(2), 485-494. doi:10.35234/fumbd.1014951
8. Du, Y., Pan, N., Xu, Z., Deng, F., Shen, Y. ve Kang, H. (2021) Pavement distress detection and classification based on YOLO network, International Journal of Pavement Engineering, 22(13), 1659-1672. doi:10.1080/10298436.2020.1714047

9. Dung, C. V. ve Anh, L. D. (2019) Autonomous concrete crack detection using deep fully convolutional neural network, *Automation in Construction*, 99, 52-58. doi:10.1016/j.autcon.2018.11.028
10. Fan, J., Bocus, M. J., Wang, L. ve Fan, R. (2021) Deep Convolutional Neural Networks for Road Crack Detection: Qualitative and Quantitative Comparisons, 2021 IEEE International Conference on Imaging Systems and Techniques (IST) (ss. 1-6), IEEE. doi:10.1109/IST50367.2021.9651375
11. Fan, Z., Lin, H., Li, C., Su, J., Bruno, S. ve Loprencipe, G. (2022) Use of Parallel ResNet for High-Performance Pavement Crack Detection and Measurement, *Sustainability (Switzerland)*, 14(3). doi:10.3390/su14031825
12. Fei, Y., Wang, K. C. P., Zhang, A., Chen, C., Li, J. Q., Liu, Y., ... Li, B. (2020) Pixel-Level Cracking Detection on 3D Asphalt Pavement Images Through Deep-Learning- Based CrackNet-V, *IEEE Transactions on Intelligent Transportation Systems*, 21(1), 273-284. doi:10.1109/TITS.2019.2891167
13. Gao Z., Zhao X., Cao M., Li Z., Liu K. ve Chen B. M. (2023) Synergizing Low Rank Representation and Deep Learning for Automatic Pavement Crack Detection, *IEEE Transactions on Intelligent Transportation Systems*, 24(10), 10676-10690. doi: 10.1109/TITS.2023.3275570
14. Gavilán, M., Balcones, D., Marcos, O., Llorca, D. F., Sotelo, M. A., Parra, I., ... Amírola, A. (2011) Adaptive Road Crack Detection System by Pavement Classification, *Sensors*, 11(10), 9628-9657. doi:10.3390/s111009628
15. Gong, M., Liu, J., Qin, A. K., Zhao, K. ve Tan, K. C. (2021) Evolving Deep Neural Networks via Cooperative Coevolution With Backpropagation, *IEEE Transactions on Neural Networks and Learning Systems*, 32(1), 420-434. doi:10.1109/TNNLS.2020.2978857
16. Huyan, J., Li, W., Tighe, S., Xu, Z. ve Zhai, J. (2020) CrackU-net: A novel deep convolutional neural network for pixelwise pavement crack detection, *Structural Control and Health Monitoring*, 27(8). doi:10.1002/stc.2551
17. Jayanth Balaji, A., Thiru Balaji, G., Dinesh, M. S., Binoy, N. ve Harish Ram, D. S. (2019) Asphalt Crack Dataset, Mendeley Data. doi:10.17632/xnzhj3x8v4.2
18. Krizhevsky, A., Sutskever, I. ve Hinton, G. E. (2017) ImageNet classification with deep convolutional neural networks, *Communications of the ACM*, 60(6), 84-90. doi:10.1145/3065386
19. Lee, B. J. ve Lee, H. "David". (2004) Position-Invariant Neural Network for Digital Pavement Crack Analysis, *Computer-Aided Civil and Infrastructure Engineering*, 19(2), 105-118. doi:10.1111/j.1467-8667.2004.00341.x
20. Liu, J., Yang, X., Lau, S., Wang, X., Luo, S., Lee, V. C. ve Ding, L. (2020) Automated pavement crack detection and segmentation based on two-step convolutional neural network, *Computer-Aided Civil and Infrastructure Engineering*, 35(11), 1291-1305. doi:10.1111/mice.12622
21. Liu, K. ve Chen, B. M. (2023) Industrial UAV-Based Unsupervised Domain Adaptive Crack Recognitions: From Database Towards Real-Site Infrastructural Inspections, *IEEE Transactions on Industrial Electronics*, 70(9), 9410-9420. doi:10.1109/TIE.2022.3204953
22. Liu, K., Han, X. ve Chen, B. M. (2019) Deep Learning Based Automatic Crack Detection and Segmentation for Unmanned Aerial Vehicle Inspections, 2019 IEEE International Conference on Robotics and Biomimetics (ROBIO) (ss. 381-387), IEEE. doi:10.1109/ROBIO49542.2019.8961534
23. Ma, D., Fang, H., Wang, N., Xue, B., Dong, J. ve Wang, F. (2022) A real-time crack detection algorithm for pavement based on CNN with multiple feature layers, *Road Materials and Pavement Design*, 23(9), 2115-2131. doi:10.1080/14680629.2021.1925578

24. Mei, Q. ve Gül, M. (2020) A cost effective solution for pavement crack inspection using cameras and deep neural networks, *Construction and Building Materials*, 256, 119397. doi:10.1016/j.conbuildmat.2020.119397
25. Mei, Q., Gül, M. ve Azim, M. R. (2020) Densely connected deep neural network considering connectivity of pixels for automatic crack detection, *Automation in Construction*, 110, 103018. doi:10.1016/j.autcon.2019.103018
26. Moon, H.-G. ve Kim, J.-H. (2011) Intelligent Crack Detecting Algorithm on the Concrete Crack Image Using Neural Network, *International Association for Automation and Robotics in Construction (IAARC)*. doi:10.22260/ISARC2011/0279
27. Munawar, H. S., Hammad, A. W. A., Haddad, A., Soares, C. A. P. ve Waller, S. T. (2021) Image-Based Crack Detection Methods: A Review, *Infrastructures*, 6(8), 115. doi:10.3390/infrastructures6080115
28. Naddaf-Sh, S., Naddaf-Sh, M.-M., Kashani, A. R. ve Zargarzadeh, H. (2020) An Efficient and Scalable Deep Learning Approach for Road Damage Detection, *2020 IEEE International Conference on Big Data (Big Data)* (ss. 5602-5608), IEEE. doi:10.1109/BigData50022.2020.9377751
29. Nguyen, T. S., Begot, S., Duculty, F. ve Avila, M. (2011) Free-form anisotropy: A new method for crack detection on pavement surface images, *2011 18th IEEE International Conference on Image Processing* (ss. 1069-1072), IEEE. doi:10.1109/ICIP.2011.6115610
30. O'Byrne, M., Schoefs, F., Ghosh, B. ve Pakrashi, V. (2013) Texture Analysis Based Damage Detection of Ageing Infrastructural Elements, *Computer-Aided Civil and Infrastructure Engineering*, 28(3), 162-177. doi:10.1111/j.1467-8667.2012.00790.x
31. Oliveira, H. ve Correia, P. (2009) Automatic road crack segmentation using entropy and image dynamic thresholding, *2009 17th European Signal Processing Conference*, 622-626. <https://ieeexplore.ieee.org/document/6302929> adresinden erişildi.
32. Oliveira, H. ve Correia, P. L. (2014) Automated Visual Inspection of Pavement Crack Detection and Characterization, <https://api.semanticscholar.org/CorpusID:9409925> adresinden erişildi.
33. Peng, L., Chao, W., Shuangmiao, L. ve Baocai, F. (2015) Research on Crack Detection Method of Airport Runway Based on Twice-Threshold Segmentation, *2015 Fifth International Conference on Instrumentation and Measurement, Computer, Communication and Control (IMCCC)* (ss. 1716-1720), IEEE. doi:10.1109/IMCCC.2015.364
34. Potter, M. A. ve Jong, K. A. (1994) A cooperative coevolutionary approach to function optimization (ss. 249-257). doi:10.1007/3-540-58484-6_269
35. Qin, A. K., Huang, V. L. ve Suganthan, P. N. (2009) Differential Evolution Algorithm With Strategy Adaptation for Global Numerical Optimization, *IEEE Transactions on Evolutionary Computation*, 13(2), 398-417. doi:10.1109/TEVC.2008.927706
36. Qu, Z., Chen, W., Wang, S.-Y., Yi, T.-M. ve Liu, L. (2022) A Crack Detection Algorithm for Concrete Pavement Based on Attention Mechanism and Multi-Features Fusion, *IEEE Transactions on Intelligent Transportation Systems*, 23(8), 11710-11719. doi:10.1109/TITS.2021.3106647
37. Rumelhart, D. E. ve McClelland, J. L. (1987) Learning Internal Representations by Error Propagation, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition: Foundations* (ss. 318-362). <https://ieeexplore.ieee.org/document/6302929> adresinden erişildi.
38. Santhi, B., Krishnamurthy, G., S, S. ve Ramakrishnan, P. K. (2012) Automatic detection of cracks in pavements using edge detection operator, *Journal of Theoretical and Applied Information Technology*, 36, 199-205. https://www.researchgate.net/publication/289942204_Automatic_detection_of_cracks_in_pavements_using_edge_detection_operator adresinden erişildi.

39. Shi, Y., Cui, L., Qi, Z., Meng, F. ve Chen, Z. (2016) Automatic Road Crack Detection Using Random Structured Forests, *IEEE Transactions on Intelligent Transportation Systems*, 17(12), 3434-3445. doi:10.1109/TITS.2016.2552248
40. Staniek, M. (2017) Detection of cracks in asphalt pavement during road inspection processes, *Scientific Journal of Silesian University of Technology, Series Transport*, 96, 175-184. doi:10.20858/sjsutst.2017.96.16
41. Tsai, Y.-C., Kaul, V. ve Mersereau, R. M. (2010) Critical Assessment of Pavement Distress Segmentation Methods, *Journal of Transportation Engineering*, 136(1), 11-19. doi:10.1061/(ASCE)TE.1943-5436.0000051
42. Vyas, V., Singh, A. P. ve Srivastava, A. (2019) Entropy-based fuzzy SWOT decision-making for condition assessment of airfield pavements, *International Journal of Pavement Engineering*, 22(10), 1226–1237. doi:10.1080/10298436.2019.1671590
43. Wang, W. ve Su, C. (2021) Deep Learning-Based Real-Time Crack Segmentation for Pavement Images, *KSCE Journal of Civil Engineering*, 25(12), 4495-4506. doi:10.1007/s12205-021-0474-2
44. Xin Yao. (1999) Evolving artificial neural networks, *Proceedings of the IEEE*, 87(9), 1423-1447. doi:10.1109/5.784219
45. Yang, M., Yu, K., Zhang, C., Li, Z. ve Yang, K. (2018) DenseASPP for Semantic Segmentation in Street Scenes, 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (ss. 3684-3692), IEEE. doi:10.1109/CVPR.2018.00388
46. Yeum, C. M. ve Dyke, S. J. (2015) Vision-Based Automated Crack Detection for Bridge Inspection, *Computer-Aided Civil and Infrastructure Engineering*, 30(10), 759-770. doi:10.1111/mice.12141
47. Zhang, A. A., Wang, K. C. P., Liu, Y., Zhan, Y., Yang, G., Wang, G., ... Shang, J. (2022) Intelligent pixel-level detection of multiple distresses and surface design features on asphalt pavements, *Computer-Aided Civil and Infrastructure Engineering*, 37(13), 1654-1673. doi:10.1111/mice.12909
48. Zhang T., Wang D. ve Lu Y. (2023) ECSNet: An Accelerated Real-Time Image Segmentation CNN Architecture for Pavement Crack Detection, *IEEE Transactions on Intelligent Transportation Systems*, 24(12), 15105-15112. doi: 10.1109/TITS.2023.3300312
49. Zhou, J. (2006) Wavelet-based pavement distress detection and evaluation, *Optical Engineering*, 45(2), 027007. doi:10.1117/1.2172917
50. Zhu, G., Liu, J., Fan, Z., Yuan, D., Ma, P., Wang, M., ... Wang, K. C. P. (2023) A lightweight encoder–decoder network for automatic pavement crack detection, *Computer-Aided Civil and Infrastructure Engineering*. doi:10.1111/mice.13103
51. Zou, Q., Cao, Y., Li, Q., Mao, Q. ve Wang, S. (2012) CrackTree: Automatic crack detection from pavement images, *Pattern Recognition Letters*, 33(3), 227-238. doi:10.1016/j.patrec.2011.11.004
52. Zou, Q., Li, Q., Zhang, F., Xiong Qian Wang, Z. ve Wang, Q. (2016) Path voting based pavement crack detection from laser range images, 2016 IEEE International Conference on Digital Signal Processing (DSP) (ss. 432-436), IEEE. doi:10.1109/ICDSP.2016.7868594