

Karayollarındaki Asfalt Çatlaklarının Tespiti İçin Yeni Bir Konvolüsyonel Sinir Ağı Tabanlı Yöntem

Gürkan DOĞAN^{1*}, Burhan ERGEN²

¹ Bilgisayar Mühendisliği, Mühendislik Fakültesi, Munzur Üniversitesi, Tunceli, Türkiye

² Bilgisayar Mühendisliği, Mühendislik Fakültesi, Fırat Üniversitesi, Elazığ, Türkiye

*¹ gurkandogan@munzur.edu.tr, ² bergen@firat.edu.tr

(Geliş/Received: 27/10/2021;

Kabul/Accepted:21/04/2022)

Öz: Karayolu yüzeyindeki asfalt çatlakları en yaygın kaplama hasarlarından biridir. Bu çatlaklar zamanında tespit edilip bakıma alınmazsa büyüyerek trafik güvenliğini ve yoğunluğunu arttıracak boyutlara ulaşmaktadır. Bu nedenle, asfalt çatlaklarının tespitinin hızlı bir şekilde gerçekleştirilmesi çok önemlidir. Geleneksel manuel çatlak tespiti, son derece zaman alıcı, çok maliyetli ve çok çaba gerektirir. Bu nedenle, araştırmacılar odaklarını otomatik asfalt çatlaklarının tespitine yoğunlaştırdılar. Ancak, son yıllarda otomatik çatlak tespiti kapsamlı olarak araştırılmasına rağmen çatlakların farklı yoğunluğa sahip olması ve kaplama ortamının karmaşıklığı nedeniyle hala zor bir görevdir. Bu çalışmada, bu zorluğun üstesinden gelmek için konvolüsyon sinir ağı temelli bir yöntem önerildi. Önerilen yöntem, sınıflandırma ve segmentasyondaki başarısı ve hafifliği bilinen MobileNetv2'nin temelde kullandığı konvolüsyon ve ters artıklı blok yapılarını baz alarak geliştirildi. Yapılan deneysel testler sonucunda, önerilen yöntemin performansı literatürdeki diğer yöntemlere göre daha yüksek olduğu görülmektedir. Bu da, otomatik asfalt çatlak tespitinin daha başarılı olması anlamına gelmektedir.

Anahtar kelimeler: Çatlak tespiti, asfalt çatlakları, piksel tabanlı segmentasyon, konvolüsyonel sinir ağları.

A New Convolutional Neural Network Based Method for Detection of Asphalt Cracks on Highways

Abstract: Asphalt cracks on the highway surface are one of the most common pavement damage. If these cracks are not detected and taken care of on time, they grow and reach dimensions that will increase traffic safety and density. Therefore, it is very important to carry out the detection of asphalt cracks quickly. Traditional manual crack detection is extremely time consuming, very costly and requires a lot of effort. Therefore, the researchers concentrated their focus on the detection of automatic asphalt cracks. However, although automatic crack detection has been extensively investigated in recent years, it is still a difficult task due to the fact that cracks have different densities and the complexity of the pavement environment. In this study, a convolutional neural network-based method is proposed to overcome this difficulty. The proposed method was developed based on the convolution and inverted residual block structures used by MobileNetv2, which is known for its success and lightweight in classification and segmentation. As a result of the experimental tests, it is seen that the performance of the proposed method is higher than the other methods in the literature. This means that automatic asphalt crack detection is more successful.

Key words: Crack detection, asphalt cracks, pixel-wise segmentation, convolutional neural networks.

1. Giriş

Asfalt kaplaması çatlağı, karayolu yüzeylerinde oluşan kusurların en önemlilerindedir. Bu tür çatlakların oluşması, karayolunun yaşlanmasını hızlandırırken yol yataklarının stabilitesini ve sağlamlığını da olumsuz olarak etkiler. Asfalt çatlakları, zamanında bakım ve rehabilitasyona alınmazsa karayolunun güvenliği olumsuz olarak etkilebilir [1]. Ayrıca, çatlaklar bakıma alınmazsa trafik yoğunluğunun artmasına, kaplamanın eskimesine ve zamanla diğer yapılardan bağlarını ayırarak (debonding) yol bakım maliyetlerini artmasına yol açabilir. Bununla ilgili olarak, dünya çapında, kaplama inşası ve bakımı için her yıl 400 milyar \$ harcanmaktadır [2]. Bu bağlamda, yaygın olarak kaplama çatlaklarının tespiti için geleneksel yöntemlerden olan görsel tabanlı manuel inceleme kullanılmaktadır [3]. Bu yöntemler, kaçınılmaz olarak hatalara, düşük verimliliğe, tespit süresinin uzun olmasına ve yüksek maliyete yol açmaktadır. Bu nedenle, araştırmacılar, son yıllarda büyük bir gelişme gösteren görüntü işleme yöntemlerini temel alan çalışmalar üzerine odaklanmaktadır. Kaplama çatlaklarının otomatik olarak tespitinde genellikle gri ölçekli eşik belirlemeli yaklaşımlar kullanılmıştır [4], [5]. Bu çalışmalarda, çatlak bölgesindeki piksellerin değeri ve arka plan piksellerin değeri arasındaki farka dayalı olarak, çatlak piksellerini sınıflandırmak için tüm görüntünün veya görüntü bloğunun bir eşiği hesaplanmaktadır. Bu sayede, çatlak taslağı

* Sorumlu yazar: gurkandogan@munzur.edu.tr. Yazarların ORCID Numarası: ¹ 0000-0003-2497-8348, ² 0000-0003-3244-2615

tespit edilmiştir. Ayenu-Prah ve ark. [6], kaldırım çatlağı tespiti için Sobel kenar dedektörü ile İki Boyutlu Deneysel Mod Ayırıştırma (BDMA) birleştirmiştir. İlk olarak, gürültüyü gidermek amacıyla BDMA 'yı kullanmışlar ve ardından kalan görüntüleri analiz etmek için Sobel kenar dedektörünü kullanırlar. Ancak, kenar belirleme çalışmalarda, çatlak görüntüsü zayıf bir kontrasta sahipse performans kötü olmaktadır. Diğer taraftan, geleneksel görüntü işleme tekniklerinin yanı sıra farklı aydınlatma ortamları ve farklı çatlak şekillerinde daha iyi bir performans gösteren makine öğrenmesini temel alan algoritmalar da kullanılmıştır. Hu ve ark. [7], öznelik çıkarmak için doku analizi ve şekil tanımlayıcıları kullanırken çatlakları sınıflandırmak için de destek vektör makineleri kullanmıştır. Hoang ve ark. [8], asfalt kaplama çatlak tespiti ve sınıflandırma için bir model geliştirmiştir. Mokhtari ve ark. [9], kaplama çatlaklarını tespit etmek için karşılaştırmalı denetimli sınıflandırma teknolojisini kullanmıştır.

Son yıllarda, derin öğrenme teknikleri üstün performanslarıyla dikkat çekmektedir. Özellikle, bilgisayar görmesi alanında konvolüsyonel sinir ağlarının nesne algılama ve bölütlemedeki olağanüstü performansları araştırmacıların kaplama çatlaklarının otomatik tespitine odaklanmasını sağlamıştır. Bu amaçla yapılan çalışmalar genelde bölge tabanlı ve piksel tabanlı yöntemler olmak üzere ikiye ayrılmaktadır. Bölge tabanlı yöntemler, hesaplama yoğunluğu daha az olduğu için araştırmacıların ilgi odağında yer almaktadır. Cha ve ark. [10], önerdikleri yaklaşım ile 256x256 boyutundan daha büyük bir görüntü kayan pencere tekniği ile bölgelere ayrılır. Daha sonra, geliştirdikleri CNN modeli ile her bir bölgede çatlakların olup olmadığı tespit edilebilmektedir. Bölge bazlı yöntemler yalnızca bölgelerin boyutuna ve konumuna göre çatlakların varlığı hakkında bilgi verir. Eğer çatlakların bulunduğu bölgenin boyutu ve konumu doğru verilmezse çatlak tespit edilme başarısı düşer. Bu sorunun üstesinden gelmek için piksel bazlı çatlak tespitinin yapılması gerekir. Zhang ve ark. [11], her bir pikselin çatlaka ait olup olmadığını tespit etmek için derin CNN 'yi kullanmayı önermiştir. Ancak, farklı pikseller arasındaki uzamsal ilişkileri göz ardı ettiği için çatlak genişliğini fazla tahmin ettiler. Diğer bir çalışmada, Zhang ve ark. [12], görüntüdeki her bir pikselin etiketini tahmin etmek için CNN kullanmayı önermiştir. Bu yaklaşımda, öznelikler manuel olarak çıkarılırken CNN modeli de sadece bir sınıflandırıcı görevini üstlenmektedir. Yang ve ark. [13], çatlak tespitini gerçekleştirmek için tam konvolüsyonel ağ (FCN) modelini kullanır. Bu yaklaşımda, kodlayıcı (encoder) ve kod çözücü (decoder) süreçleri sayesinde, çıktı ve girdinin aynı boyutta olması sağlanmıştır. Liu ve ark. [14], beton çatlak tespiti için ilk U-Net modeli kullanılmıştır. U-Net modeli daha küçük veri seti ile daha önceki FCN modellerinden daha yüksek doğruluk başarısı elde ettiği görülmüştür.

Diğer taraftan, gelecekte otonom araçların kara yollarında seyredeceği de düşünüldüğünde asfalt çatlakların otomatik, hızlı, verimli ve etkin maliyetli olarak tespiti çok önemli olmaktadır. Bu amaçla, bu çalışmada, düşük donanım gereksinimi ve yüksek performans başarısı sağlayan CNN modellerinden biri olan MobileNetV2 [15] temel alınmıştır. Literatürde MobileNetV2 'yi temel alan çalışmalarda asfalt çatlak tespiti pek bulunmasa dahi, [16] 'de kentsel manzaralarda bulunan nesnelere tespitinde ve [17] 'de ray yüzeyi kusurlarının tespitinde kullanılmıştır.

Bu çalışmada, MobileNetV2 CNN modeli omurgasında kullanılan konvolüsyon blok yapısı ve ters (inverted) blok yapısı temel alınarak yeni bir asfalt çatlak tespit yöntemi önerilmiştir. Önerilen yöntemin, MobileNetV2 yönteminden en belirgin farkı; kodlayıcı (encoder) olarak blok yapılarının kullanıldığı yığın (stack) yapısı ve kullanılan hiper parametrelerdedir. Ek olarak, önerilen yöntemde artıklı bağlantı (residual connection) kullanılmadığı için MobileNet CNN yöntemlerden ayrılmaktadır. Önerilen yaklaşım piksel bazlı çatlak tespiti yapmaktadır. Bu yönüyle ve daha az donanım gereksinimine ihtiyaç duyması ve yüksek doğruluk başarısı yönüyle diğer çalışmalardan pozitif olarak ayrılmaktadır. Önerilen yöntem, değerlendirilmeli (benchmark) veri setlerinden biri olan Crack500 veri seti [18] kullanılarak eğitim ve test işlemlerine tabi tutulduğunda, deneysel sonuçlar diğer çalışmalara göre gayet iyi bir performans gösterdiğini ortaya koymaktadır.

Bu yazının kalan bölümleri aşağıdaki gibi düzenlenmiştir; Bölüm 2 'de Crack500 veri seti, MobileNetV2 'in kullandığı blok yapıları, öğrenme oranı planlaması ve dice kaybı fonksiyonu ile ilgili bilgiler yer almaktadır. Bölüm 3 'te önerilen yöntem, bölüm 4 'te deneysel testler değerlendirilip analiz edilmiştir. Son olarak, bölüm 5 'te genel sonuçlara yer verilmiştir.

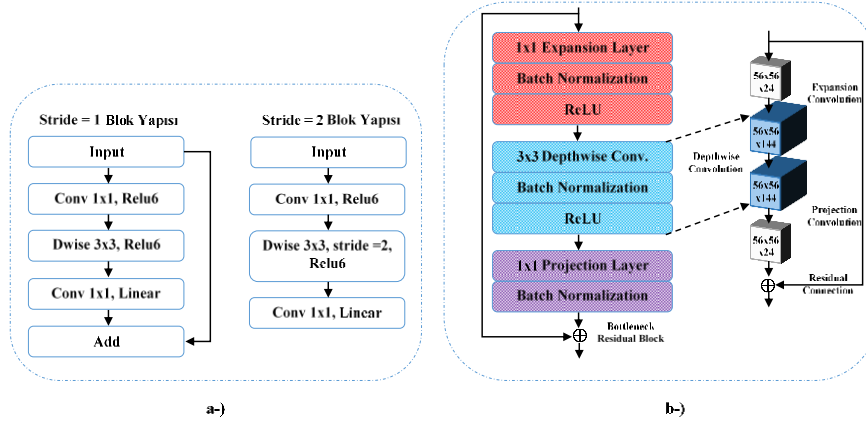
2. Materyal ve Metotlar

2.1. Veri seti

Yang ve ark. [18] tarafından kamuya açık olarak paylaşılan Crack500 veri seti, hücresel telefon kullanılarak Temple University ana kampüsünde 2000x1500 piksel boyutunda 500 tane asfalt çatlağı görüntüsü alınarak toplanılmıştır. Her bir asfalt çatlağı görüntüsüne karşılık piksel bazlı etiketlenen ikili maske görüntüsüne sahiptir. Görüntü sayısının sınırlı, her bir görüntünün büyük boyutlu ve kısıtlı hesaplama kaynaklarından dolayı, her bir

görüntü örtüşmeyen 16 görüntü bölgesine kırılmıştır ve yalnızca 1.000 pikselden fazla çatlak içeren bölgeler kaydedilmiştir. Bu yüzden, Crack500 veri seti, 224x224 boyutunda ve toplamda 3368 görüntüden oluşmaktadır.

Bu çalışmada, eğitim verisi için 2,696 görüntü, doğrulama verisi için 348 görüntü ve test verisi için 324 adet görüntü kullanılmıştır. Ayrıca, sofistike görüntü artırma tekniklerini bir arada bulunduran albumentation kütüphanesi kullanılmıştır. Crack500 veri setine, çevirme (flip), döndürme (rotate); rastgele kontrast (random contrast); rastgele gama (random gamma), rastgele parlaklık (random brightness); elastik dönüşüm (elastic transform), ızgara bozulması (grid distortion), optik bozulma (optical distortion) dönüşüm setleri uygulanmıştır.



Şekil 1. MobileNetV2 mimarisinin kullandığı konvolüsyonel blok (a) ve ters artıklı blok (b) yapıları [15], [19].

2.2. MobileNetV2 konvolüsyonel sinir ağı

MobileNetV1 [20] 'in bir üst sürümü olan ve MobileNetV1 'den daha iyi performans gösteren MobileNetV2 [15], adımlara (stride) göre farklılık gösteren iki tür darboğaz (bottleneck) katmanı olan yeni katman bloğunu uygulanmaktadır. MobileNetV2 'de sadece evrişimli katmandan oluşan darboğaz katmanı yerine, parametre sayısını daha da düşürmeyi amaçladıkları, derinlemesine bir evrişim katmanı kullanılmaktadır [15]. MobileNetV2 CNN modelinin konvolüsyonel blok yapısı ve ters artıklı (inverted residual) blok yapısı Şekil 1 'de verilmiştir. Ters artıklı bloklar, işlem sırasında bellekte depolanan tensör miktarını azaltarak bellek sorunlarını yönetmek için tasarlanmıştır. Öznitelik haritasının sayısındaki artış olan Doğrusal Darboğazlar (Linear Bottleneck), bir öznitelik haritasını sırasıyla 64 'ten 128, 256 ve 512 'ye çıkarır. Şekil 1-b 'de, 24 öznitelik haritasıyla başlayıp sırasıyla 144 haritaya çıkan ve ardından bir sonraki bloğa göndermeden önce öznitelik haritalarının sayısını 24 haritaya düşüren Doğrusal Darboğazlar sürecini göstermektedir.

Bu çalışmada, mobil sistemlerde performansı yüksek modellerden biri olan MobileNetV2 CNN modelinin blok yapıları temel alınmıştır. Bu sayede, asfalt çatlak tespitinin daha etkin maliyetle ve daha yüksek başarımla sağlanması amaçlanmıştır.

2.3. Optimizasyon yöntemi ve öğrenme oranı planı

Bu çalışmada, makine öğrenmesi ve derin öğrenme alanında trend olan Adam [21] optimizasyon tekniği ve Kosinüs Tavlama Öğrenme Oranı Planlaması [22] (Cosine Annealing Learning Rate Scheduling) kombinasyonu kullanılmıştır. Adam optimizasyon tekniği, derin sinir ağlarını eğitmek için tasarlanan uyarlanabilir bir öğrenme algoritmasıdır. Yani farklı parametreler için bireysel öğrenme oranları hesaplar. Bu teknik, birinci ve ikinci gradyan momentini kullanarak her parametre için öğrenme oranlarının hesaplanmasını kolaylaştırır. Hesaplama açısından verimli olan ADAM, daha az bellek gerektirir ve büyük veri kümelerinde daha iyi performans gösterir. Makine öğrenimi sürecinde öğrenme oranı değeri çok önemli bir faktördür. Öğrenme hızı çok küçük olduğunda öğrenme algoritması çok yavaş yakınsar, tam tersine değer çok büyükse ağ kaotik davranır ve yakınsamaz. Optimum bir öğrenme oranını belirlemenin etkili bir yolu, öğrenme oranını Kosinüs Tavlama LRS tekniği ile azaltmaktır. Bu teknikle kullanılarak eğitim ilerledikçe daha dengeli bir öğrenme oranı belirlenir. Kosinüs Tavlama LR tekniğinde, her bir batch 'in öğrenme oranı aşağıdaki denklem yardımıyla hesaplanır;

$$n_t = n_{min}^i + 1/2(n_{max}^i - n_{min}^i)(1 + \cos((T_{cur}/T_i)\pi)) \quad (1)$$

Denklem (1) 'de, n_{min}^i ve n_{max}^i öğrenme oranı aralığını belirtir. T_{cur} , son yeniden başlatmadan beri olan iterasyon (epoch) sayısıdır. i , yürütme indeksi iken t , her bir batch iterasyon sayısını belirtir.

Bu çalışmada, öğrenme oranı planlaması, Adam algoritmasına dayanan Kosinüs Tavlama tekniğiyle yapılmıştır. Bunun için, öğrenme oranı 0.1'e, alfa 0.1'e, bozulma adımı (decay step) 300'e ve eğitim 300 iterasyon boyunca sürdürülmüştür.

2.4. Dice kaybı fonksiyonu

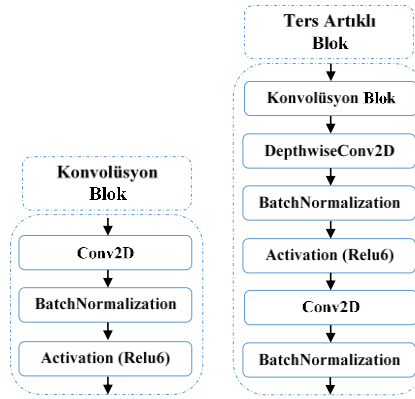
Dice Katsayısının (DC) istatistiksel ifadesi Dice [23] tarafından bireysel referans-görüntü (individual reference-image) nesne çiftlerinin eşleşmelerinin kalitesini ölçmek için kullanılmıştır. DC, ikili sınıflandırma görevlerinden türetilen bir dizi benzerlik ölçüsü fonksiyonudur [24]. $DC \in [0,1]$ olduğundan Dice Katsayısı değeri sıfırdan bire kadar değişmektedir. Görüntü segmentasyon görevlerinde ikili sınıflandırma için Dice Katsayısı aşağıdaki denklem ile hesaplanmaktadır;

$$DC = \frac{2 \sum_i^N p_i g_i + \varepsilon}{\sum_i^N p_i^2 + \sum_i^N g_i^2 + \varepsilon} \quad (2)$$

Denklem (2) 'de p_i ; tahmin edilen maskenin ikili segmentasyon miktarını, g_i ; orginal maskenin (ground truth) ikili miktarı ve N ; numune sayısını ifade etmektedir. ε 'da sıfır paydanın oluşmasını engelleyen minimum değerdir. Dice Kaybı'nın denklemi ise aşağıdaki denklem ile ifade edilmektedir.

$$DC_{loss} = 1 - DC \quad (3)$$

Bu çalışmada, [24]'de önerilen Dice Katsayısı seçilmiştir ve bunun iki nedeni vardır. İlk olarak, hesaplama açısından basittir. İkinci olarak, tıbbi görüntü işleme uygulamalarında kapsamlı ve etkin bir şekilde kullanılmıştır [25], [26].



Şekil 2. Önerilen yöntemde kullanılan konvolüsyonel blok ve ters blok yapılarının şematigi

3. Önerilen Yöntem

Bu çalışmada, asfalt catlak tespiti piksel bazında bir sınıflandırma görevi olarak ele alınmıştır ve önerilen mimari Şekil 3 'te gösterilmiştir. Mimarinin daha hafif ve performansı yüksek olmasını sağlamak için Şekil 2 'de gösterilen konvolüsyon ve ters blok yapıları kullanılmıştır. Ters blok yapısı, genişletme (expansion) katmanı, derinlemesine (depthwise) konvolüsyon ve sunum (projection) katmanı olmak üzere üç temel katmandan meydana gelirken sunum katmanı da 1x1 'lik konvolüsyon ve batch normalizasyon ve aktivasyon alt katmanlarından oluşmaktadır. Derinlemesine konvolüsyon katmanı ise 3x3 derinlemesine konvolüsyon, batch normalizasyon ve aktivasyon katmanından oluşmaktadır. Ayrıca, genişletme ve sunum katmanlarındaki konvolüsyona, noktasal (pointwise) konvolüsyon olarak da adlandırılmaktadır.

24 GB RAM, Tesla P100-PCIE-16GB gibi özelliklere sahip bir iş istasyonu ve Python version 3.7.10, Tensorflow version 2.4.1, Keras version 2.4.3 gibi özelliklere sahip yazılım teknolojileri kullanılmıştır.

Karmaşıklık matrisi, bir sınıflandırma sistemi tarafından tahmin edilen ve gerçek sınıflandırmalar hakkında bilgi içeren bir tablodur. Makine öğrenimi alanında bir sınıflandırma sisteminin performansını tanımlamak için kullanılır. Tablo 1 'de iki sınıfa ait bir karmaşıklık matrisi verilmiştir. Burada kullanılan TP, doğru-pozitifleri; TN, doğru-negatifleri; FP, yanlış-pozitifleri; FN, yanlış-negatifleri belirtmektedir. TP, TN, FP ve FN değerleri ile bir sınıflandırma sisteminin performansını ölçen doğruluk (Acc.), hassasiyet (Prec.), anma (Rec.) ve f1-skoru (F1-scr.) vb. gibi metrikler hesaplanmaktadır. Asfalt çatlağı tespiti için önerilen yaklaşımın performans analizi ve değerlendirmesi için de doğruluk (Acc.), hassasiyet (Prec.), anma (Rec.), f1-skoru (F1-scr.), özgüllük (Spec.), dice katsayısı (DC), eğrinin altında kalan alan (auc) ve ortalama kesişen bölgelerin birleşik bölgelere oranı (IoU) metrikleri kullanılmıştır. Bu performans metriklerinin denklemleri aşağıda verilmiştir;

$$Acc = (TP + TN)/(TP + FN + TN + FP) \quad (4)$$

$$Prec = TP/(TP + FP) \quad (5)$$

$$Rec = TP/(TP + FN) \quad (6)$$

$$F1-Scr = 2TP/(2TP + FP + FN) \quad (7)$$

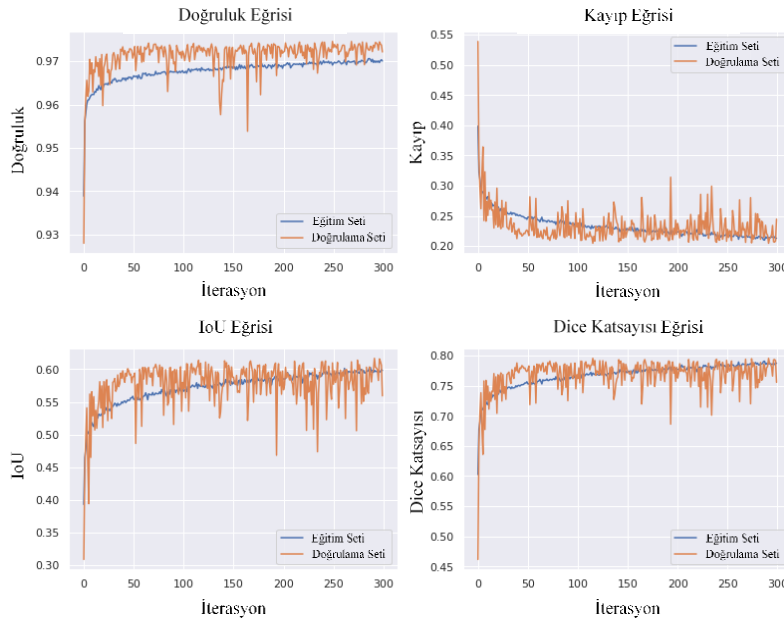
$$Spec = TN/(TN + FP) \quad (8)$$

$$DC = \frac{2.Prec.Rec}{Prec + Rec} \quad (8)$$

$$AUC(f) = \frac{\sum_{t_0 \in D^0} \sum_{t_1 \in D^1} 1[f(t_0) < f(t_1)]}{|D^0||D^1|} \quad (9)$$

$$IoU = (TP)/(TP + FP + FN) \quad (10)$$

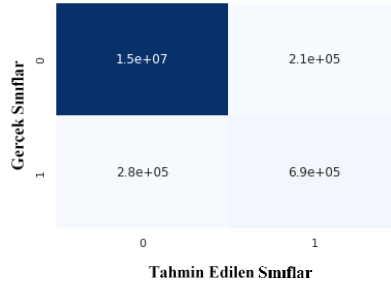
Denklem 9'da D^0 , negatif örneklerin dizisini belirtirken D^1 , pozitif örneklerin dizisini belirtir. $1[f(t_0) < f(t_1)]$ ifadesi; $f(t_0) < f(t_1)$ ifadesinin true olması durumunda 1, false olması durumunda 0 olmaktadır [27].



Şekil 4. Önerilen yöntemin doğruluk, kayıp, IoU ve dice katsayısı eğrileri

Diğer taraftan, önerilen mimarinin donanım birimi kullanım miktarını belirten parametre sayısı (the number of parameters) ve saniyedeki kayan nokta işlemi (floating point operations per second / FLOPs) gibi metriklerde kullanılmaktadır. Bu metriklerden parametre sayısı, bellek tüketimini belirtirken işlemci yükünü yani işlemci kullanım miktarını belirtir [28], [29].

Crack500 veri seti ile önerilen yaklaşım eğitildiğinde Şekil 4 'te gösterilen doğruluk, kayıp, IoU ve Dice Katsayısı grafikleri oluşmaktadır. Doğruluk Eğrisinde görüldüğü gibi eğitim ve doğrulama kümeleri yaklaşık olarak %97 'ye ulaşmıştır. Kayıp Eğrisinde ise eğitim ve doğrulama kümeleri yaklaşık olarak 0.21 'e düştüğü görülmektedir. IoU eğrisinde IoU metriği için eğitim kümesi yaklaşık olarak %60 'e doğrulama kümesi de %62 'e kadar yükselmiştir. Dice Katsayısı eğrisinde dice katsayısı metriği eğitim ve doğrulama kümeleri %79 'a ulaşmıştır.

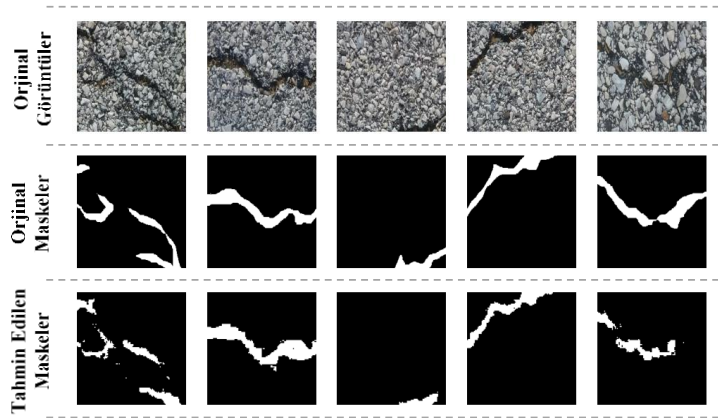


Şekil 5. Önerilen yöntem ile elde edilen karmaşıklık matrisi

Diğer taraftan, önerilen yaklaşımı değerlendirmek için Crack500 veri setinin test seti kullanılmıştır. Dolayısıyla, değerlendirme ve analiz süreçlerinde bu set için performans metriklerinin karşılaştırılması yapılmıştır. Şekil 5 'te önerilen yaklaşım ile veri seti kullanılarak elde edilen karmaşıklık matrisi verilmiştir. Burada, 1 (bir) beyaz pikselleri belirtirken 0 (sıfır) siyah pikselleri belirtmektedir. Bir diğer deyişle, sıfırlar asfalt kaplamasındaki çatlakları, birler ise arka planı belirtir. Önerdiğimiz yöntem bir piksel-bazlı tahmin yaptığı için bir görüntüdeki piksellerin her biri bir etiket (label) olarak düşünülmüştür. Her bir orijinal RGB piksele karşılık gelen ikili segmentasyon pikselleri tahmin edilmiştir. Dolayısıyla, karmaşıklık matrisinde TN, doğru-negatif olarak tahmin edilen piksellerin sayısı 15068666, FP, yanlış-pozitif olarak tahmin edilen piksellerin sayısı 211328, FN, yanlış-negatif olarak tahmin edilen piksellerin sayısı 284266 ve TP, doğru-pozitif olarak tahmin edilen piksellerin sayısı 692764 olmaktadır.

Tablo 2. Önerilen yöntemden elde edilen performans metriklerinin değerleri

Acc.	Prec.	Rec.	F-Scr.	Spec.	DC	AUC	IoU
0.969	0.766	0.709	0.736	0.986	0.736	0.981	0.583



Şekil 6. Önerilen yöntem kullanılarak tahmin edilen maskeler ile orijinal maskelerin kıyaslaması

Önerilen yaklaşım test veri seti ile tahmin işlemine tabi tutulduğunda bazı performans metriklerinin değerleri Tablo 2 'deki gibi olmaktadır. Bu performans metrikleri bir ağın diğer ağlara olan üstünlüğünü ve girdi (input) olarak verilen bir veri setinde ne kadar başarılı olduğunu belirtmesi bakımından önemlidir. Tablo 2 'de görüldüğü gibi bu performans metriklerinden doğruluk değeri 0.969, hassasiyet değeri 0.766, anma değeri 0.709, f1-skoru değeri 0.736, özgüllük değeri 0.986, dice katsayısı değeri 0.736, auc değeri 0.981 ve IoU değeri 0.583 olmuştur.

Önerilen yaklaşım Crack500 test veri seti ile test işlemine tabi tutulduğunda; girdi olarak 3 kanallı bir yol yüzeyi çatlağı görüntüsünden maske tahmin edilir. Bu işlem için veri setinden rastgele olarak seçilen 5 görüntünün kıyaslaması Şekil 6 'da sunulmuştur. Burada, orjinal görüntüler, RGB renklerinden oluşmaktadır ve önerilen ağ modeli bu görüntülerden maskeleri tahmin etmektedir. Bu şekildeki orjinal maskelerle tahmin edilen maskeler dikkatle incelendiğinde orjinal maskelerde olmayan bazı beyaz ve siyah piksellerin tahmin edilen maskelerde ortaya çıktığı gözlenmektedir. Bu da, orjinal ve tahmin edilen maskelerin farklılaştırılan bir faktördür. Bu farklılaşma; IoU, DC vb. gibi önemli bazı performans metriklerinin değerinin azalmasına yol açmaktadır.

Tablo 3. Crack500 veri seti üzerinde farklı mimarilerin kullanılması ile test sonuçlarının kıyaslaması

Metotlar	Prec.	Rec.	DC	IoU
U-Net by Nguyen et al. [30]	0.695	0.674	0.689	0.526
CNN by Fan et al. [31]	0.712	0.695	0.705	0.545
MobileNetV2 by Sandler et al. [15]	0.784	0.724	0.728	0.573
Xception by Chollet [32]	0.792	0.747	0.755	0.574
Split-Attention Network [33]	0.736	0.716	0.729	0.574
U-Net by Lau et al. [34]	0.742	0.728	0.732	0.578
Önerilen Yöntem	0.766	0.709	0.736	0.583

Diğer taraftan, önerilen yaklaşımın literatürde yer alan diğer yöntemlerle karşılaştırması için Tablo 3 'te verilmiştir. Bu karşılaştırma tablosundaki tüm yöntemler Crack500 veri seti kullanılarak eğitim ve testleri gerçekleştirilmiştir. [30], [31] ve [33] 'deki yöntemlerin Crack500 veri seti ile yeniden uygulaması (re-implementation) Lau ve ark.[34] tarafından yapılmıştır. [15] ve [32] 'deki yöntemlerin Crack500 veri seti ile yeniden uygulaması Wang ve ark [35] tarafından yapılmıştır. Önerilen yöntem ile diğer yöntemler performans metriklerinin sonuçları ile kıyaslandığında anma (Rec.) metriği hariç diğer tüm metriklerde en yüksek değerler önerdiğimiz yaklaşımdan elde edilmiştir. Tahmin edilen bir maskenin orijinal maskeye (ground truth) benzerliğini ölçen metriklerden dice katsayısı ve IoU diğer yöntemlerden daha yüksek başarıya ulaşmıştır. Bu da, önerilen modelin diğer modellerden daha iyi bir performans gösterdiğini ifade etmesi açısından önemlidir.

Tablo 4. Literatürdeki farklı hafif (lightweight) mimarilerin performans kıyaslaması

Metotlar	Parametre Sayısı (Milyon)	FLOPs (Milyar)
MobileNetV1 by Howard et al. [20]	11.15	14.25
MobileNetV2 by Sandler et al. [15]	4.52	5.80
Önerilen Yöntem	1.99	0.60

Diğer taraftan, önerilen yöntemin literatürdeki diğer yöntemlerden donanım birimi kullanım miktarını belirten parametre sayısı ve FLOPs gibi metriklerin kıyaslamaları Tablo 4 'te görüldüğü verilmiştir. Bu tabloda, MobileNetV1 ve MobileNetV2 gibi yöntemler, hafif (lightweight) olduğunu iddia ettiklerinden dolayı bu kıyaslama için seçilmiştir. Bu kıyaslama tablosuna göre, önerilen yöntem en yakın rakibi olan MobileNetV2 yöntemine göre yaklaşık olarak 2 kat hafif ve 9 kat hesaplama karmaşıklığı daha azdır. Bu da, önerilen yöntemin diğer yöntemlere göre daha az donanım kullanması anlamına gelmektedir.

Sonuç olarak, önerilen yöntemin yığın yapısında ters blok yapısı kullanılarak parametre sayısı ve ters blok yapısında da artıklı bağlantı kullanılmayarak saniyedeki kayan nokta işlemi sayısı dikkate değer şekilde azaltılmıştır. Dolayısıyla, önerilen yöntem diğer yöntemlerden daha az donanım birimi kullanımına ihtiyaç duyarken karayollarındaki asfalt çatlaklarını da yüksek performansla tespit edebilmektedir.

5. Sonuçlar

Bu çalışmada, asfalt çatlaklarını daha az maliyetle ve daha yüksek performansla tespit eden konvolüsyonel sinir ağı temelli bir yaklaşım önerilmiştir. Önerdiğimiz yöntem, konvolüsyon blokları ve ters blokları temel alan mimarisiyle piksel bazlı çatlak tespiti yapmaktadır. Bu yöntemde, asfalt çatlak görüntüsünün her bir pikseli bir etiket olarak düşünülmüştür. Önerilen yöntem ile Crack500 veri seti kullanılarak eğitim ve test işlemleri gerçekleştirilmiştir. Deneysel testleri değerlendirmek için de hassasiyet, anma, dice katsayısı (DC), ortalama kesişen bölgelerin birleştirilen bölgelere oranı (IoU) gibi metrikler kullanıldı. Yapılan deneysel testler sonucunda; hassasiyet, anma, DC ve IoU değerleri sırasıyla yaklaşık olarak 0.77, 0.71, 0.74 ve 0.58 'dir. Bu sonuçlar, önerdiğimiz yöntemin literatürde yer alan diğer çalışmalardan daha yüksek bir başarı elde ettiğini göstermektedir. Ek olarak, önerilen yöntem konvolüsyon ve ters konvolüsyon yapıları üzerine inşa edildiği için hafif ağırlıklara sahiptir ve mobil sistemlerde kullanılabilir. Bu da, asfalt çatlağının tespitinde donanım maliyetlerinin azaltırken performansın artmasını sağlamıştır.

Kaynaklar

- [1] B. Safaei and N. Safaei, "Studying the risks and factors contributing to motorcycle crashes , and prioritizing strategies to reduce fatalities , and improve community health," no. February, 2021, doi: 10.13140/RG.2.2.23936.35843/1.
- [2] C. Torres-Machi, A. Chamorro, E. Pellicer, V. Yepes, and C. Videla, "Sustainable pavement management: Integrating economic, technical, and environmental aspects in decision making," *Transp. Res. Rec.*, vol. 2523, pp. 56–63, 2015, doi: 10.3141/2523-07.
- [3] J. Eriksson, L. Girod, B. Hull, R. Newton, S. Madden, and H. Balakrishnan, "The Pothole Patrol: Using a Mobile Sensor Network for Road Surface Monitoring," in *Proceeding of the 6th international conference on Mobile systems, applications, and services - MobiSys '08*, 2008, vol. 70, no. 9, p. 29, doi: 10.1145/1378600.1378605.
- [4] N. Otsu, "A Threshold Selection Method from Gray-Level Histograms," *IEEE Trans. Syst. Man. Cybern.*, vol. 9, no. 1, pp. 62–66, Jan. 1979, doi: 10.1109/TSMC.1979.4310076.
- [5] Z. Sun, G. Bebis, and R. Miller, "On-road vehicle detection using evolutionary gabor filter optimization," *IEEE Trans. Intell. Transp. Syst.*, vol. 6, no. 2, pp. 125–137, 2005, doi: 10.1109/TITS.2005.848363.
- [6] N. Attoh-Okine and A. Ayenu-Prah, "Evaluating pavement cracks with bidimensional empirical mode decomposition," *EURASIP J. Adv. Signal Process.*, vol. 2008, 2008, doi: 10.1155/2008/861701.
- [7] Y. Hu, C. X. Zhao, and H. N. Wang, "Automatic pavement crack detection using texture and shape descriptors," *IETE Tech. Rev. (Institution Electron. Telecommun. Eng. India)*, vol. 27, no. 5, pp. 398–405, 2010, doi: 10.4103/0256-4602.62225.
- [8] N. D. Hoang and Q. L. Nguyen, "A novel method for asphalt pavement crack classification based on image processing and machine learning," *Eng. Comput.*, vol. 35, no. 2, pp. 487–498, 2019, doi: 10.1007/s00366-018-0611-9.
- [9] S. Mokhtari, L. Wu, and H. B. Yun, "Comparison of supervised classification techniques for vision-based pavement crack detection," *Transp. Res. Rec.*, vol. 2595, pp. 119–127, 2016, doi: 10.3141/2595-13.
- [10] Y. J. Cha, W. Choi, and O. Büyükoztürk, "Deep Learning-Based Crack Damage Detection Using Convolutional Neural Networks," *Comput. Civ. Infrastruct. Eng.*, vol. 32, no. 5, pp. 361–378, 2017, doi: 10.1111/mice.12263.
- [11] L. Zhang, F. Yang, Y. Daniel Zhang, and Y. J. Zhu, "Road crack detection using deep convolutional neural network," in *2016 IEEE International Conference on Image Processing (ICIP)*, Sep. 2016, pp. 3708–3712, doi: 10.1109/ICIP.2016.7533052.
- [12] A. Zhang *et al.*, "Automated Pixel-Level Pavement Crack Detection on 3D Asphalt Surfaces Using a Deep-Learning Network," *Comput. Civ. Infrastruct. Eng.*, vol. 32, no. 10, pp. 805–819, 2017, doi: 10.1111/mice.12297.
- [13] X. Yang, H. Li, Y. Yu, X. Luo, T. Huang, and X. Yang, "Automatic Pixel-Level Crack Detection and Measurement Using Fully Convolutional Network," *Comput. Civ. Infrastruct. Eng.*, vol. 33, no. 12, pp. 1090–1109, 2018, doi: 10.1111/mice.12412.
- [14] Z. Liu, Y. Cao, Y. Wang, and W. Wang, "Computer vision-based concrete crack detection using U-net fully convolutional networks," *Autom. Constr.*, vol. 104, no. January, pp. 129–139, 2019, doi: 10.1016/j.autcon.2019.04.005.
- [15] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 4510–4520, 2018, doi: 10.1109/CVPR.2018.00474.
- [16] T. Emara, H. E. A. El Munim, and H. M. Abbas, "LiteSeg: A Novel Lightweight ConvNet for Semantic Segmentation," *2019 Digit. Image Comput. Tech. Appl. DICTA 2019*, 2019, doi: 10.1109/DICTA47822.2019.8945975.
- [17] J. H. Feng, H. Yuan, Y. Q. Hu, J. Lin, S. W. Liu, and X. Luo, "Research on deep learning method for rail surface defect detection," *IET Electr. Syst. Transp.*, vol. 10, no. 4, pp. 436–442, 2020, doi: 10.1049/iet-est.2020.0041.
- [18] F. Yang, L. Zhang, S. Yu, D. Prokhorov, X. Mei, and H. Ling, "Feature Pyramid and Hierarchical Boosting Network for Pavement Crack Detection," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 4, pp. 1525–1535, 2020, doi: 10.1109/TITS.2019.2910595.

- [19] P. Enkvetchakul and O. Surinta, "Effective Data Augmentation and Training Techniques for Improving Deep Learning in Plant Leaf Disease Recognition," *Appl. Sci. Eng. Prog.*, 2021, doi: 10.14416/j.asep.2021.01.003.
- [20] A. G. Howard *et al.*, "MobileNets: Efficient convolutional neural networks for mobile vision applications," *arXiv*, 2017.
- [21] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," *3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc.*, pp. 1–15, 2015.
- [22] I. Loshchilov and F. Hutter, "SGDR: Stochastic gradient descent with warm restarts," *5th Int. Conf. Learn. Represent. ICLR 2017 - Conf. Track Proc.*, pp. 1–16, 2017.
- [23] L. R. Dice, "Measures of the Amount of Ecologic Association Between Species," *Ecology*, vol. 26, no. 3, pp. 297–302, Jul. 1945, doi: 10.2307/1932409.
- [24] F. Milletari, N. Navab, and S. A. Ahmadi, "V-Net: Fully convolutional neural networks for volumetric medical image segmentation," *Proc. - 2016 4th Int. Conf. 3D Vision, 3DV 2016*, pp. 565–571, 2016, doi: 10.1109/3DV.2016.79.
- [25] K. O. Babalola *et al.*, "An evaluation of four automatic methods of segmenting the subcortical structures in the brain," *Neuroimage*, vol. 47, no. 4, pp. 1435–1447, 2009, doi: 10.1016/j.neuroimage.2009.05.029.
- [26] L. Wang, C. Wang, Z. Sun, and S. Chen, "An improved dice loss for pneumothorax segmentation by mining the information of negative areas," *IEEE Access*, vol. 8, pp. 167939–167949, 2020, doi: 10.1109/ACCESS.2020.3020475.
- [27] T. Calders and S. Jaroszewicz, "Efficient AUC optimization for classification," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 4702 LNAI, pp. 42–53, 2007, doi: 10.1007/978-3-540-74976-9_8.
- [28] J. Qiu *et al.*, "Going Deeper with Embedded FPGA Platform for Convolutional Neural Network," in *Proceedings of the 2016 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, Feb. 2016, pp. 26–35, doi: 10.1145/2847263.2847265.
- [29] A. Shawahna, S. M. Sait, and A. El-Maleh, "FPGA-Based accelerators of deep learning networks for learning and classification: A review," *IEEE Access*, vol. 7, pp. 7823–7859, 2019, doi: 10.1109/ACCESS.2018.2890150.
- [30] N. T. H. Nguyen, T. H. Le, S. Perry, and T. T. Nguyen, "Pavement crack detection using convolutional neural network," *ACM Int. Conf. Proceeding Ser.*, pp. 251–256, 2018, doi: 10.1145/3287921.3287949.
- [31] F. Yang, L. Zhang, S. Yu, D. Prokhorov, X. Mei, and H. Ling, "Feature Pyramid and Hierarchical Boosting Network for Pavement Crack Detection," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 4, pp. 1525–1535, 2020, doi: 10.1109/TITS.2019.2910595.
- [32] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," *Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017*, vol. 2017-Janua, pp. 1800–1807, 2017, doi: 10.1109/CVPR.2017.195.
- [33] H. Zhang *et al.*, "ResNeSt: Split-Attention Networks," 2020, [Online]. Available: <http://arxiv.org/abs/2004.08955>.
- [34] S. L. H. Lau, E. K. P. Chong, X. Yang, and X. Wang, "Automated Pavement Crack Segmentation Using U-Net-Based Convolutional Neural Network," *IEEE Access*, vol. 8, pp. 114892–114899, 2020, doi: 10.1109/ACCESS.2020.3003638.
- [35] W. Wang and C. Su, "Convolutional Neural Network-Based Pavement Crack Segmentation Using Pyramid Attention Network," *IEEE Access*, vol. 8, pp. 206548–206558, 2020, doi: 10.1109/ACCESS.2020.3037667.