2021 5th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT) | 978-1-6654-4930-4/21/$31.00 ©2021 IEEE | DOI: 10.1109/ISMSIT52890.2021.9604645

# Monoküler Derinlik Çıkarımı Sağlayan Bir Derin Öğrenme Modelinin Web Servisi Aracılığıyla Mobil Cihazlar Üzerinde Test Edilmesi

# Testing of a Deep Learning Model Providing Monocular Depth Estimation on Mobile Devices via Web Service

Alper Tunga Akın
Mühendislik Fakültesi
Karadeniz Teknik Üniversitesi
Trabzon, Türkiye
alpertunga@ktu.edu.tr

Çetin Cömert
Mühendislik Fakültesi
Karadeniz Teknik Üniversitesi
Trabzon, Türkiye
ccomert@ktu.edu.tr

***Abstract*—Augmented reality applications running on smartphones or tablets are becoming increasingly common. It is crucial to extract the physical structure of the scene perceived by the device camera in these applications. In such applications, employed in education, navigation and obstacle notification, the distance information between the device camera and the object must be derived and processed with sufficient accuracy and speed. In this study, the deep learning model, named "From Big To Small (BTS)", with superior performance metrics in depth extraction according to the literature reviews, was transformed into a web service and tested on an Android phone. Thus, a deep learning model with a high computational cost will be available on an Android device with average processing power. Test results were examined, and improvements were discussed.**

***Keywords—Augmented Reality; Web Services; Monocular Depth Estimation; Deep Learning.***

***Özetçe*— Akıllı telefonlar ya da tabletler üzerinde çalışan ve kullanımı gittikçe yaygınlaşan artırılmış gerçeklik uygulamalarında, cihaz kamerasıyla algılanan sahnenin fiziksel yapısının çıkarımı önem arz etmektedir. Eğitim, navigasyon ve engel bildirimi gibi konularda kullanılan bu uygulamalarda, cihaz kamerası ve obje arasındaki mesafe bilgisinin yeterli doğruluk ve hızda türetilip işleme alınması gerekmektedir. Bu çalışmada, derinlik çıkarımı konusunda literatür incelemelerine göre üstün performansa sahip olan "From Big To Small (BTS)" isimli derin öğrenme modeli bir web servisine dönüştürülüp, bir Android telefon üzerinde test edilmiştir. Modelin web servisine dönüştürülmesiyle donanımdan bağımsızlık amaçlanmıştır. Böylelikle, hesaplama maliyeti hayli yüksek bir derin öğrenme modeli, ortalama işlemci gücüne sahip bir Android cihaz vasıtasıyla kullanılabilir duruma gelecektir. Test sonuçları irdelenip iyileştirmeler üzerine tartışılmıştır..**

***Anahtar Kelimeler—Artırılmış Gerçeklik; Web Servisleri; Monoküler Derinlik Çıkarımı; Derin Öğrenme.***

## I. GİRİŞ

Artırılmış gerçeklik (AR) uygulamalarının insan yaşamındaki etki alanı gün geçtikçe artmaktadır. Özellikle eğitim, navigasyon, rehberlik ve engel bildirimi gibi konularda özelleşen AR uygulamalarıyla bireyin yaşam kalitesini artırılması hedeflenmektedir. Bundan ötürü AR alanındaki girişim ve araştırmalar da yüksek bir ivme ile hız kazanmaktadır. AR kısaca, gerçek dünyanın bilgisayar üretimi bilgi ile iyileştirilmesi ya da eş bir deyişle sentetik bilgi ile gerçek çevrenin entegre edilmesidir [1][2]. Dolayısıyla, akıllı telefon ya da tabletler üzerinde çalışan ve bireyin çıplak gözle edinemediği metrik derinlik bilgisini birbirinden farklı yöntemlerle türetip sunan çalışmalar, gerçekliğin artırılmasında kilit rol oynamaktadırlar.

Metrik derinlik bilgisi çıkarımını akıllı telefon ya da tablet gibi mobil cihazlar aracılığıyla sağlayan çalışmalar, geometrik ve algısal yaklaşımlar olmak üzere iki ana gruba ayrılabilir. Geometrik yaklaşımda derinlik bilgisi, çoklu görüntü geometrisi prensiplerine göre, stereo görüntülerin eşleştirmesi ile elde edilir. Telefon kamerası ile alınan görüntü kaydının karelerinin birbirleri arasındaki dış yöneltme elemanlarının bilinmesiyle görüntü kareleri birbirleriyle eşleştirilerek epipolar düzlem üzerinden kamera-obje arasındaki derinlik hesaplanır [3]. Buna en yaygın AR uygulaması örneği olarak Google'a ait ARCore Depth API isimli Unity çapraz platform oyun motoru eklentisi verilebilir. Referans [4] incelenirse bu eklentide gerekli çıktının Time-of-Flight (ToF) sensör barındıran belirli akıllı telefonlar üzerinden, 8 metre ile sınırlı menzilde sağlanabildiği görülmektedir. ToF sensörü lazer ışınlarının seyahat süresi üzerinden derinlik hesaplayan bir sensördür. Ve tüm akıllı telefonlarda bulunmamaktadır. Öyle ki, böyle bir sensör olmadan yalnızca akıllı telefonun kendi içerisindeki Inertial Measurement Unit (IMU) grubu sensörleriyle görüntü kareleri arasındaki dış yöneltme parametrelerini hesaplamak mümkün olmamaktadır.

Tarafımızca ToF sensörü bulunmayan bir akıllı telefonla gerçekleştirilen deneylerde IMU sensörlerinden gelen anlık verinin, özellikle ivmeölçer sensörünün, filtreleme müdahelelerine rağmen barındırdığı yüksek gürültü nedeniyle canlı görüntü kaydı üzerinden derinlik çıkarımı başarısız olmuştur.

Algısal yaklaşım grubunda ise RGB görüntü ve bu görüntüye karşılık gelen derinlik haritaları ile eğitilen derin öğrenme modelleri bulunur. Bu yaklaşımla gerçekleştirilmiş mobil çalışmalar incelendiğinde çoğunlukla indoor (iç mekan) görüntüleri üzerinde çalışıldığı görülmüştür. Referans [5]'teki çalışmada Pytorch derin öğrenme kütüphanesinin ARM işlemciler üzerinde derin öğrenme modellerinin koşturulması amacıyla optimize edilmiş versiyonu olan Pytorch Mobile kütüphanesini kullanarak 0.497m RMSE ve 67 ms gecikme ile indoor görüntüler üzerinden monoküler derinlik çıkarımını başarmıştır [6]. Model, NYU Depth v2 isimli indoor görüntüler ve bu görüntülere karşılık Microsoft Kinect cihazı ile üretilmiş derinlik haritalarından oluşan veri seti ile eğitilmiştir [7]. Benzer şekilde [8]'de indoor görüntüler üzerinden monoküler derinlik çıkarımı için bir istemci-sunucu mimarisi oluşturmuşlardır. Maksimum 0.118m RMSE ve 447.548 ms gecikme ile yüksek performanslı grafik işlemci (GPU) barındıran bir sunucu üzerinden sonuç almayı başarmışlardır.

Bu çalışmada, outdoor (dış mekan) görüntülerinden monoküler derinlik çıkarımı işlemini bir Android telefon üzerinden gerçekleştirmek için, Referans [9]'daki "From Big to Small (BTS)" isimli derin öğrenme modelinin bir web servisine dönüştürülmesi amaçlanmıştır. Bu model, referans [10]'daki literatür incelemesinde doğruluk metriklerine göre, KITTI Raw Depth outdoor veri seti üzerinde en yüksek performansı göstermiş iki derin öğrenme modelinden biridir [11]. Diğer emsal modellerin üstünde bir doğrulukla, 2.005m RMSE ile monoküler derinlik çıkarımı sağlamaktadır. Bu modelin web servisine dönüştürülmesiyle, istemcinin akıllı telefon konfigürasyonundan bağımsız bir şekilde herhangi bir ek donanıma gerek duymadan, yalnızca telefon kamerasıyla çekilmiş RGB outdoor görüntüler üzerinden monoküler derinlik çıkarımının gerçek zamanlı AR kullanımı için denenmesi amaçlanmıştır.

## II. YÖNTEM

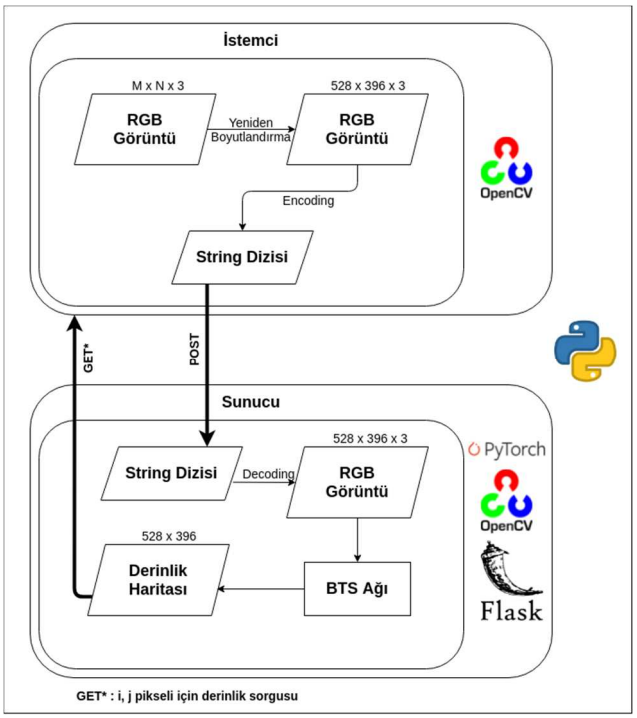
### A. Veri Seti

KTÜ Merkez Kampüsü içerisinde Huawei FIG-LX1 model Android cep telefonu ile 5 adet RGB dış mekan görüntüsü çekilmiştir. Görüntülerin çekildiği noktalardan, görüntülerde kadraja girmiş 31 adet objeye elektronik uzaklık ölçer cihazı ile eğik mesafe ölçümü yapılmıştır. Objelerin tümünün cihaz kamerasına uzaklığı 1 - 28 metre aralığındadır. Bu şekilde test veri seti oluşturulmuştur.

### B. From Big To Small (BTS) Derinlik Çıkarımı Modeli

BTS modeli bir encoding-decoding ardışık yapısına sahip derin öğrenme modelidir. Encoding aşamasında girdi görüntü üzerindeki objelere ait öznitelikler (feature) çıkarılır ve decoder aşamasında bu öznitelikler üzerinden görüntü üzerindeki her bir piksele ait derinlik bilgisi türetilir. BTS modelinin aynı amaca yönelik tasarlanmış derin öğrenme modellerine kıyasla başarısı, girdi görüntü üzerindeki objelere ait özniteliklerle çıktı derinlik haritası arasındaki ilişkiyi emsallerine kıyasla daha etkili biçimde kurmasına dayanmaktadır. Bu üstünlüğü sağlayan yapı, decoding aşamasındaki "local planar guidance" katmanıdır. Bu katman sayesinde encoding aşamasında elde edilen öznitelik haritasındaki özniteliklerin, decoding aşamasında derinlik haritası oluşturulması için upsample edilmesi (sık örnekleme - boyut yükseltme) sırasında oluşan uzamsal bilgi kaybının önüne geçilmektedir. Eş bir deyişle, görüntü keskinleştirmeye benzer bir yaklaşımla, keskinlik kaybının önüne geçen bir yapı ortaya konulmuş ve derinlik haritasındaki obje sınırları emsal modellere kıyasla daha yüksek hassasiyette oluşturulmuştur. Ayrıca bu işlem H, H/2, H/4 ve H/8 olmak üzere 4 farklı çözünürlükteki öznitelik haritaları üzerinde uygulandığı için derinlik haritası üzerinde farklı ölçekteki objelerin temsil edilmesi noktasında da üstün performans sağlamaktadır.

Uygulama kapsamında kullanılan pretrained (ön eğitimli) BTS modeli öznitelik çıkarımı için encoding aşamasında DenseNet-161 ağı ile KITTI veri seti kullanılarak eğitilmiştir [12].




**Şekil 1**. Uygulama Mimarisi

### C. Web Servisi Tasarımı

Derinlik çıkarımı için kullanılan BTS modelinin, [13]'te bağlantısı verilen Github adresindeki Pytorch kullanılarak yazılmış Python implementasyonu, bir web servisine dönüştürülmüştür. Bu implementasyonun web servisine dönüştürülmesinde Flask isimli web geliştirme kütüphanesi kullanılmıştır [14]. İmplementasyon, sunucu tarafında bir Flask fonksiyonuna dönüştürülüp POST metodu olarak bir URL ile ilişkilendirilmiştir.
İstemci tarafında RGB test görüntüsü öncelikle sunucuya POST edilebilmesi için encode edilmekte, yani bir string dizisi olarak ikili dosyaya dönüştürülmektedir. Bu ikili dosya, derinlik sorgusu yapılacak görüntü indis bilgisi ile birlikte paketlenerek BTS modelinin ilişkilendirildiği URL'ye POST metodu ile gönderilmektedir.

Sunucuya POST edilen bu paket, ikili dosya ve indis olarak sunucuda geri ayrıştırılmaktadır. İkili dosya decode edilip, yani sunucu içerisinde tekrar RGB görüntüye dönüştürülüp, BTS modeline girdi olarak verilmektedir. BTS modeli ise bu görüntüye karşılık bir derinlik haritası üretmektedir. Derinlik haritası üzerinden alınan, sunucuya gelen indis bilgilerine karşılık gelen metrik birimdeki derinlik bilgisi, GET metodu ile istemci tarafından çekilmektedir.

Encode-decode işlemleri dahil olmak üzere tüm görüntü üzerindeki işlemlerde OpenCV görüntü işleme kütüphanesi kullanılmıştır [15]. Tasarlanan yapının iş akışı Şekil 1 üzerinden incelenebilir.

### *D. Doğruluk Metrikleri*

Web servisi vasıtasıyla sunulan modelin performansının değerlendirilmesinde, literatürde bu alanda sıkça kullanılan Mean Absolute Error (Ortalama Mutlak Hata) (MAE), Root Mean Square Error (Karesel Ortalama Hata) (RMSE), Square Relative Error (Karesel Bağıl Hata) (SqRel) metrikleri kullanılmıştır. Bu metrikler şu şekilde tanımlanır:

$$RMSE = \sqrt{\frac{1}{N}\Sigma_{i=1}^{N}\left(\frac{d_i - d_i'}{d_i}\right)^2} \quad (1)$$

$$\mathrm{MAE} = \frac{1}{\mathrm{N}}\sum_{\mathrm{i}=1}^{\mathrm{N}}|\mathrm{d_i} - \mathrm{d_i'}| \quad (2)$$

$$\mathrm{SqRel} = \frac{1}{\mathrm{N}}\sum_{\mathrm{i}=1}^{\mathrm{N}}\left(\frac{|\mathrm{d_i} - \mathrm{d_i'}|^2}{\mathrm{d_i}}\right) \quad (3)$$

burada di ölçülen, di' ise model ile çıkarılan derinlik ve N test objesi sayısıdır.

## III. SONUÇLAR

Elektronik uzaklık ölçer ile eğik mesafe okuması yapılan 31 objenin test görüntüleri üzerindeki indisleri ile okunan mesafeler ilişkilendirilmiştir. Ardından test görüntülerinin sırayla, oluşturulan web servisi aracılığıyla, derinlik haritaları edinilmiştir (Şekil 2). Bu derinlik haritaları üzerinden test objelerinin indislerine karşılık gelen derinlik bilgileri çıkarılmış ve sahada ölçülen eğik mesafelerle kıyaslanmıştır (Şekil 3).

Web servisinin, BTS modelinin, cep telefonu kamerası üzerindeki performansı Tablo-1' de doğruluk metrikleriyle verilmiştir.

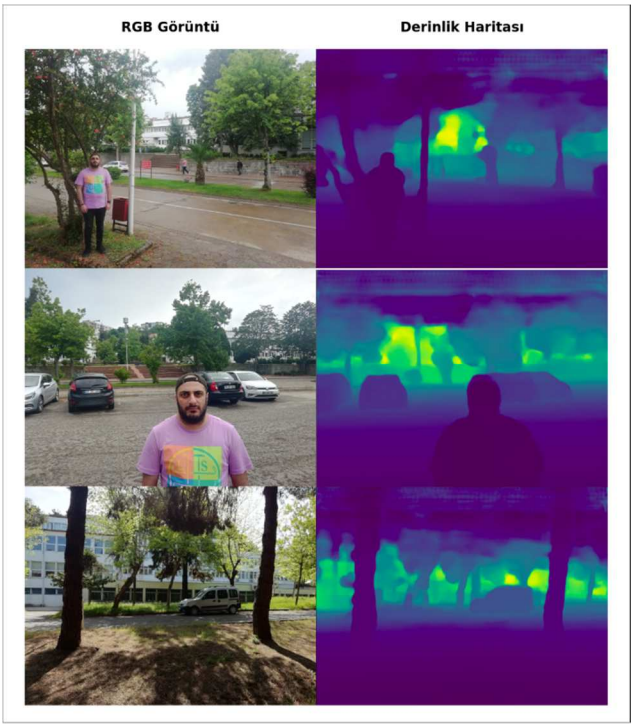
| Mesafe (d) | MAE | RMSE | SqRel |
|---|---|---|---|
| d<10 | 0,173 | 0,245 | 0,036 |
| d<15 | 0,465 | 0,757 | 0,063 |
| d<20 | 0,530 | 0,843 | 0,067 |
| d>20 | 0,718 | 1,107 | 0,080 |

**Tablo I**. Doğruluk Metrikleri

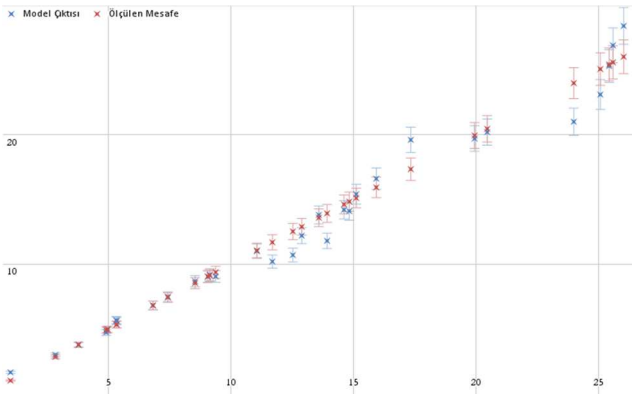
Tablo üzerinden de anlaşılacağı gibi uygulama menzili ve metrikler bakımından tasarlanan web servisinin yanıt doğruluğu, ilgili literatür bağlamında tatmin edici düzeydedir.

## IV. TARTIŞMA

Şekil-1' deki iş akışında da görüleceği üzere, RGB test görüntüsü sunucuya gönderilmeden önce istemci tarafında yeniden boyutlandırılıp, boyutu düşürülmektedir. Bunun iki sebebi vardır. Birincisi, istemci-sunucu arasındaki veri akış hızını artırmak ve dolayısıyla gerçek zamanlı kullanım için çalışma zamanını düşürmektir. İkincisi ve en önemlisi ise, sunucudaki Nvidia GTX 1050 4GB GPU'nun hafıza kısıtlamasıdır. Kullanılan BTS implementasyonundaki DenseNet-161 modelinin hesapladığı parametre sayısının yüksekliğinden dolayı, görüntü boyutunun büyümesi hafıza yönetimi açısından olumsuz etki yaratmaktadır. Öyle ki, görüntüde herhangi bir boyut indirgemesi yapılmaksızın GPU üzerinde işlem yapmak, sunucudan yanıt almak, mümkün olmamaktadır. 528*396 boyutları sunucunun yanıt süresi ve doğruluk metrikleri göz önünde bulundurularak deneysel olarak elde edilmiştir. Bu halde dahi, BTS modelinin sunucu üzerindeki çalışma zamanı ortalama 5 sn düzeyindedir. Doğruluk metriklerinden ödün vermeksizin bu süreyi düşürmek için implementasyonun kullandığı hesaplama yükü açısından oldukça maliyetli DenseNet-161 haricinde, hesaplanan parametre sayısının daha az olduğu, bir öznitelik çıkarıcı ağ kullanılarak BTS modeli cep telefonu kamerasıyla çekilmiş görüntülerle yeniden eğitilebilir. Bu çözüm gelecek çalışma olarak hedeflenmektedir. Böylelikle hem hesaplanan parametre sayısı azalacağı için uygulamanın çalışma zamanı gerçek zamanlı kullanıma daha uygun hale gelecek hem de görüntüdeki boyut indirgemesi oranı düşürülebilecektir. Her ne kadar elde edilen doğruluk metrikleri literatür bağlamında tatmin edici düzeyde olsa da, görüntü boyutlarının düşürülmesi görüntü üzerindeki arka plan objelerine ve küçük ölçekteki objelere ait detayların kaybolmasına sebep olmaktadır. Bu durum aslında uygulamanın doğruluğunu olumsuz yönde etkileyen bir durumdur.




**Şekil 2.** RGB Görüntülere Karşılık Derinlik Haritaları




**Şekil 3.** Ölçülen Mesafe - Model Çıkarımı Mesafe (kırmızı: ölçülen, mavi: model çıkarımı)

## V. KAYNAKÇA


[1] Furht, B., ark., *Encyclopedia of multimedia*. Springer Science & Business Media, 2008.

[2] Bimber, O., & Raskar, R., *Spatial augmented reality: merging real and virtual worlds*. AK Peters/CRC Press, 1-5, 2005.

[3] Hartley, R., Zisserman, A., *Multiple view geometry in computer vision*. Cambridge university press, 262-279, 2003.

[4] Du, Ruofei, ark., "DepthLab: Real-Time 3D Interaction With Depth Maps for Mobile Augmented Reality." *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*, 2020

[5] Wang, Y., "MobileDepth: Efficient Monocular Depth Prediction on Mobile Devices." arXiv preprint arXiv:2011.10189, 2020.

[6] Paszke, A., ark., "Pytorch: An imperative style, high-performance deep learning library." arXiv preprint arXiv:1912.01703, 2019.

[7] Silberman, N., Hoiem, D., Kohli, P., & Fergus, R., "Indoor segmentation and support inference from RGBD images" *In European conference on computer vision,* 746-760, Springer, Berlin, Heidelberg, 2012

[8] Schurischuster, S., Andrija K., Robert S., "In-Time 3D Reconstruction and Instance Segmentation from Monocular Sensor Data." *2020 17th Conference on Computer and Robot Vision (CRV),* IEEE, 2020

[9] Lee, J. H., Han, M. K., Ko, D. W., & Suh, I. H., "From big to small: Multi-scale local planar guidance for monocular depth estimation", arXiv preprint arXiv:1907.10326, 2019

[10] Khan, F., Salahuddin, S., Javidnia, H., "Deep Learning-Based Monocular Depth Estimation Methods—A State-of-the-Art Review", Sensors, 20, 2272.. doi:10.3390/s20082272, 2020

[11] Geiger, A., Lenz, P., Stiller, C., Urtasun, R., "Vision meets robotics: The kitti dataset", *The International Journal of Robotics Research*, 32(11), 1231-1237, 2013

[12] Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K. Q., "Densely connected convolutional networks", *In Proceedings of the IEEE conference on computer vision and pattern recognition,* 4700-4708, 2017

[13] BTS – PyTorch, Erişim Adresi: https://github.com/ErenBalatkan/Bts-PyTorch, 2020

[14] Grinberg, M., *Flask web development: developing web applications with python*., O'Reilly Media, Inc., 2018.

[15] Howse, J., *OpenCV computer vision with python*, Packt Publishing Ltd, 2013.