



U-Net ile Çekirdek Segmentasyonunda Hiper Parametre Optimizasyonu Etkisinin Değerlendirilmesi

Furkan Atlan^{1*}, Emrah Hançer², İhsan Pençe³

¹ Burdur Mehmet Akif Ersoy Üniversitesi, Sosyal Bilimler Enstitüsü, Yönetim Bilişim Sistemleri Bölümü, Burdur, Türkiye (ORCID: 0000-0003-1602-1941)

² Burdur Mehmet Akif Ersoy Üniversitesi, Bucak Teknoloji Fakültesi, Yazılım Mühendisliği Bölümü, Burdur, Türkiye (ORCID: 0000-0002-3213-5191)

³ Burdur Mehmet Akif Ersoy Üniversitesi, Bucak Teknoloji Fakültesi, Yazılım Mühendisliği Bölümü, Burdur, Türkiye (ORCID: 0000-0003-0734-3869)

(International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT) 2020 – 22-24 Ekim 2020)

(DOI: 10.31590/ejosat.818791)

ATIF/REFERENCE: Atlan, F., Hançer, E. & Pençe, İ. (2020). U-Net ile Çekirdek Segmentasyonunda Hiper Parametre Optimizasyonu Etkisinin Değerlendirilmesi. *Avrupa Bilim ve Teknoloji Dergisi*, (Özel Sayı), 60-69.

Öz

Tıbbi görüntülerin yorumlanarak hasta ve hastalık hakkında önemli veriler elde edilmesi zaman ve emek açısından oldukça maliyetlidir. Tıbbi görüntülerin yapay zekâ yöntemleri ile analiz edilmesi sayesinde hastalık tespitinin yapılması, sınıflandırılması ve bunların otomatikleştirilmesi uzmanların iş yükünü azaltmaktadır. Bu çalışmada, 2018 Data Science Bowl veri setinden elde edilen tıbbi görüntülerdeki çekirdeklerin tespitinin otomatikleştirilmesi yapılmaktadır. 2018 Data Science Bowl, çekirdek tespitinin otomatikleştirilmesi amacıyla bir yarışma için bir araya getirilmiş tıbbi görüntülerden oluşmaktadır. Bu veri seti içerisinde 670 adet eğitim görüntüsü ve bunlara ait maske görüntüleri ile 65 adet test görüntüsü yer almaktadır. Test görüntülerine ait maske görüntüleri olmadığı için çalışma, eğitim görüntülerinin %10'unun doğrulama verisi olarak ayrılarak bu veriler üzerinden yapılmaktadır. Çalışmada, çekirdek tespitinin otomatikleştirilmesinde U-Net Evrişimli Sinir Ağı kullanılıp en yüksek doğruluk ve en düşük hata oranının elde edilmesi amaçlanmıştır. Çekirdek tespitindeki doğruluk oranının artırılması ve hata oranının minimize edilmesi amaçlandığı için, yapılan deneysel çalışmalar iki bölüme ayrılmaktadır. İlk olarak Adam optimizasyonu ve belirli hiper parametreler kullanılmakta, ikinci bölümde ise sonuçların en iyilenmesi için hiper parametre optimizasyonu yapılmaktadır. Hiper parametre optimizasyonu için Izgara Arama yöntemi tercih edilmiştir. Deneysel çalışmaların ilk aşamasında doğrulama verileri için elde edilen doğruluk oranı 0.9675 olarak bulunmuştur. Hiper parametre optimizasyonu yapıldıktan sonra ise doğrulama verileri için en iyi iki sonuç sırası ile 0.9698 ve 0.9739 olarak bulunmuştur. Son olarak, en iyi sonucu veren 0.9739 oranı için belirlenen hiper parametreler Stokastik Gradyan İniş, RMSProp ve AdaDelta optimizasyon yöntemleri için de denenmiştir. Bu sayede optimizasyon yöntemlerin kıyaslaması da yapılmış ve en iyi sonucu veren optimizasyon yönteminin Adam optimizasyonu olduğu belirlenmiştir.

Anahtar Kelimeler: Tıbbi Görüntü Segmentasyonu, Çekirdek Tespiti Otomasyonu, U-Net, Hiper Parametre Optimizasyonu, Izgara Arama.

Evaluation of Hyper Parameter Optimization Effect on Nuclei Segmentation with U-Net

Abstract

Interpreting medical images and obtaining important data about the patient and the disease is very costly in terms of time and effort. Thanks to the analysis of medical images with artificial intelligence methods, disease detection, classification and automating these is reduced the workload of experts. In this study, the process of automating the detection of the nuclei in medical images obtained from the 2018 Data Science Bowl data set is carried out. 2018 Data Science Bowl consists of medical images put together for a competition to automate nuclei detection. This data set includes 670 training images and their mask images and 65 test images. Since there are no mask images belonging to the test images, the study is carried out by separating 10% of the training images as validation data. In the study, it is aimed to achieve the highest accuracy and the lowest error rate by using U-Net Convolutional Neural Network to automate nuclei detection. Since it is aimed to increase the accuracy rate in nuclei detection and to minimize the error rate, the experimental

* Sorumlu Yazar: Burdur Mehmet Akif Ersoy Üniversitesi, Sosyal Bilimler Enstitüsü, Yönetim Bilişim Sistemleri Bölümü, Burdur, Türkiye, ORCID: 0000-0003-1602-1941, furkanatlan@gmail.com

studies are divided into two parts. Firstly, the hyper parameters determined by the trainer have been run with Adam optimization, and in the second part, hyper parameter optimization is performed to optimize the results. Grid Search method is preferred for hyper parameter optimization. In the first stage of experimental studies, the accuracy rate obtained for verification data was found to be 0.9675. After the hyper parameter optimization, the best two results for the validation data were found as 0.9698 and 0.9739, respectively. Finally, the hyper parameters determined for the ratio of 0.9739 that gives the best result are also tested for Stochastic Gradient Descent, RMSProp and AdaDelta optimization methods. Thus, the comparison of optimization methods is also made and it was determined that the optimization method that gave the best results is Adam optimization.

Keywords: Medical Image Segmentation, Nuclei Detection Automation, Hyper Parameter Optimization, Grid Search.

1. Giriş

Hücre ve çekirdek bölütlemesi, dijital hale getirilmiş mikroskopi görüntülerinin yani tıbbi görüntülerin otomatik olarak analiz edilmesine olanak sağlayan çok önemli bir gelişmedir. Çünkü çekirdek bölütlemesinde kullanılan görüntüler, dijitalleştirilmiş mikroskopi görüntüleridir (Veta ve ark., 2013). Dokunun mikroskobik incelenmesi ile ilgilenen temel anabilim dalının adı histopatolojidir. Spesifik olarak, klinik tıpta histopatoloji; dokunun işlendikten ve histopatolojik bölümlerin cam slaytlara yerleştirilmesinden sonra biyopsi örneğinin bir patolog tarafından incelenmesini kapsar. Bu cam slaytlar, tüm slaytları dijitalleştirilen son teknoloji dijital slayt tarayıcıları (WSI) yardımıyla taranarak bilgisayara aktarılabilir (Wang ve ark., 2016). Büyük sıralama yaklaşımları sayesinde kanserin moleküler temelli olarak ele alınışına ciddi bir katkı sağlanmıştır. Dijital patoloji sayesinde hastalıklı doku kısımlarının taranması ya da bunları dijital bir şekilde depolanması, oldukça karmaşık bu verilerin sayısal ve otomatik analizi için birtakım yazılımların geliştirilmesine olanak sağlamaktadır (Naylor ve ark., 2017). Genel olarak görüntü bölütleme algoritmaları, süreksizlik ve benzerlik kategorisi altında toplanmaktadır. Süreksizlik, görüntüdeki ayrıtlarda yeğinlik bilgisinin anlık değişimlerine göre görüntünün parçalara ayrılması; benzerlik ise görüntünün, önceden tanımlanmış belirli özelliklere göre parçalara ayrılmasıdır. Eşik seviyesi, bu alanda en sık kullanılan kriterlerden biridir (Gonzalez ve Woods, 2018). Yapay zekânın sağlık alanında kullanılması ile beraber tıbbi görüntülerin bölütlemesine yönelik yazılımlar ve çalışmalar artmış olsa da çekirdek bölütlemesi oldukça zor bir iştir. Doku ve hücre türleri, hücrelere göre boyama farklılıkları gibi faktörler çekirdek bölütlemesinde kullanılan geleneksel bölütleme algoritmaları tarafından çok zor ayırt edilir ve oldukça yavaş çalışır. Ancak yapay zekâ kullanılarak yapılan çalışmalarda bölütleme işlemi hızlandırılmış (Şişeci ve ark., 2014) ve derin öğrenme kullanılarak geliştirilen algoritmalar ile çekirdek bölütleme çalışmalarında büyük bir başarı elde edilmiştir (Naylor ve ark., 2017; Ciresan ve ark., 2012).

Günümüzde tıbbi görüntülerin yapay zekâ ile işlenmesi, araştırmacılar için oldukça tercih edilen bir konudur. Tıbbi görüntülerin yarı ya da tam teşhisinin otomasyonu oldukça zor bir çalışma alanıdır (Gülay ve İçer, 2020). Bu görüntülerin yorumlanarak bilgisayar destekli tanı sistemleri ile hastalık tespiti yapılması, sınıflandırılması hem uzmanlar (radyolog, klinisyen vb.) hem hastalar hem de bu alanda hizmet veren kurumlar açısından oldukça önemlidir (Öziç ve Özşen, 2020). Tıbbi görüntülerin doğru bir şekilde etiketlenmesi oldukça zahmetli ve maliyetli bir iştir. Üstelik klasik görüntü işleme tabanlı algoritmalar veya modeller için oldukça fazla sayıda veri yani görüntü gerekmektedir.

Literatürde tıbbi görüntülerin bölütlenmesi ile ilgili çalışmalar özellikle son birkaç yılda büyük bir artış göstermektedir. Yapılan çalışmalar patoloji görüntülerindeki hücre çekirdeklerinin bölütlenmesi (Turan ve Bilgin, 2019), kan damarlarının bölütlenmesi (Atlı ve Gedik, 2019), hücre çekirdeğinin bölütlenmesinde farklı mimarilerin kıyaslanması (Vuola ve ark., 2019), hassas çözünürlüğe sahip tıbbi görüntülerin bölütlenmesi için özel modellerin oluşturulması (Alom ve ark., 2018; Punn ve Agarwal, 2019; Chidester ve ark., 2019; Zeng ve ark., 2019) ve Hibrit modellerin kullanılması (Fuat ve ark., 2019) gibi bir çerçevede bir araya getirilebilir.

Çalışmada kullanılan veri seti ve analiz edilen metrik (doğruluk oranı) ile ilgili literatürde yapılmış ve kıyaslanabilecek bir çalışma olarak görüntülerin bölütlenmesinin zar katsayısı açısından doğruluğunu gösteren bir çalışma (Alom ve ark., 2018) bulunmaktadır. Alom ve ark., (2018) R2U-Net modeli kullanarak yaptıkları çalışmada görüntüleri zar katsayısı açısından %92.15 doğruluk oranı ile bölütlemeyi başarmıştır. Bu oran, bu çalışmadaki iki araştırma sonucundan da düşüktür. Bu durum, bu çalışmadaki sonuçların başarısını ortaya koymaktadır. Literatürde, mevcut veri seti ile yapılan çalışmaların az olmasının sebebi olarak veri setinin 2018 yılında duyurulması yani çok yeni olmasıdır. Ayrıca çalışmada kullanılan metriktten elde edilen sonucun optimize edilmesi de bu çalışmanın özgünlüğünü ve güncelliğini açık bir şekilde ortaya koymaktadır.

Literatür taramasında yer alan söz konusu çalışmalarda kullanılan veri setleri, hastanelerden elde edilen tıbbi görüntülerden oluşmaktadır. Bu veri setleri ile ilgili çalışmalar aynı zamanda istatistik bilgileri açısından da oldukça kıymetlidir. Bu veri setlerinde ilgili hastalığın cinsiyetlere, ülke içerisindeki bölgelere göre dağılımı gibi konularda da bilgi sahibi olmak açısından faydalanılabilir. Tıbbi görüntülerin analizi, bölütlenmesi vb. çalışmalarda kullanılan veri setleri genellikle internet ortamındaki halka açık veri tabanlarından veya ilişkili üniversite, devlet veya özel hastanelerden elde edilen verilerden oluşmaktadır.

Bu çalışmada, 2018 Data Science Bowl veri setinde yer alan hücre çekirdeği görüntülerinin bölütlenmesi (segmentasyonu) yapılmaktadır. Çalışmanın amacı, çekirdeklerin tespitinin otomatikleştirilmesi ve bu işlemin büyük bir doğruluk oranı ve en az hata oranı ile gerçekleştirilmesidir. Çalışmanın en yüksek bir doğruluk oranı ile tamamlanması için elde edilen sonuçların hiper parametre optimizasyonu ile en iyilenmesi de amaçlanmaktadır.

2. Materyal ve Metot

2.1. Veri Seti

Çalışmada kullanılan veri setini, 2018 Data Science Bowl (<https://www.kaggle.com/c/>) kaynağından elde edilen çok sayıda bölünmüş çekirdek görüntüsünden oluşturmaktadır. Bu veri seti içerisinde, 670 adet eğitim görüntüsü ve bunlara ait farklı resim dosyalarında tutulan ve bir araya getirildiğinde 670 adet maske (Segmente edilmiş) görüntü verisi bulunmaktadır.

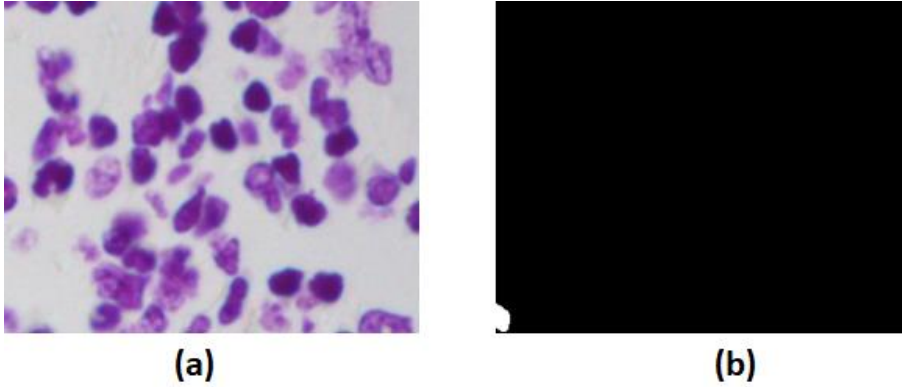
Bu 670 adet eğitim görüntüsünün 607'si eğitim görüntüsü olarak, %10'u olan 67 tanesi ise doğrulama (validation) verisi olarak kullanılmaktadır.

Bu veri seti içerisinde yer alan verilere ait istatistikler Tablo 1'de gösterilmektedir.

Tablo 1. Data Science Bowl 2018 Veri Setine ait İstatistikler

Eğitim Verisi	Test Verisi	Veri Formatı	Toplam Veri
670	65	Portable Network Graphics (PNG)	735

Şekil 1'de orijinal hücre görüntüsü ve bu hücreye ait bölütlenmiş bir maske görüntüsü verilmektedir.



Şekil 1. 2018 Data Science Bowl Veri Setindeki bir eğitim görüntüsü: (a) Orijinal Gösterim, (b) Bölütlenmiş Gösterim

Şekil 1'den anlaşılacağı üzere eğitim verisi içerisindeki çekirdek görüntülerine ait her bir bölüt, maske olarak adlandırılmaktadır. Örneğin; bir eğitim verisi olarak kabul edilen çekirdek görüntüsünde 20 adet çekirdek bir veri olarak kabul edilirken, bu veriye ait 20 maske oluşturulmakta ve her bir bölüt bir görüntü olarak ifade edilmektedir. Test verisi olan 65 görüntüye ait maske görüntüsü bulunmamakta ve bu görüntülerin araştırmacı tarafından oluşturulacak bir model ile en doğru şekilde tahmin edilmesi beklenmektedir.

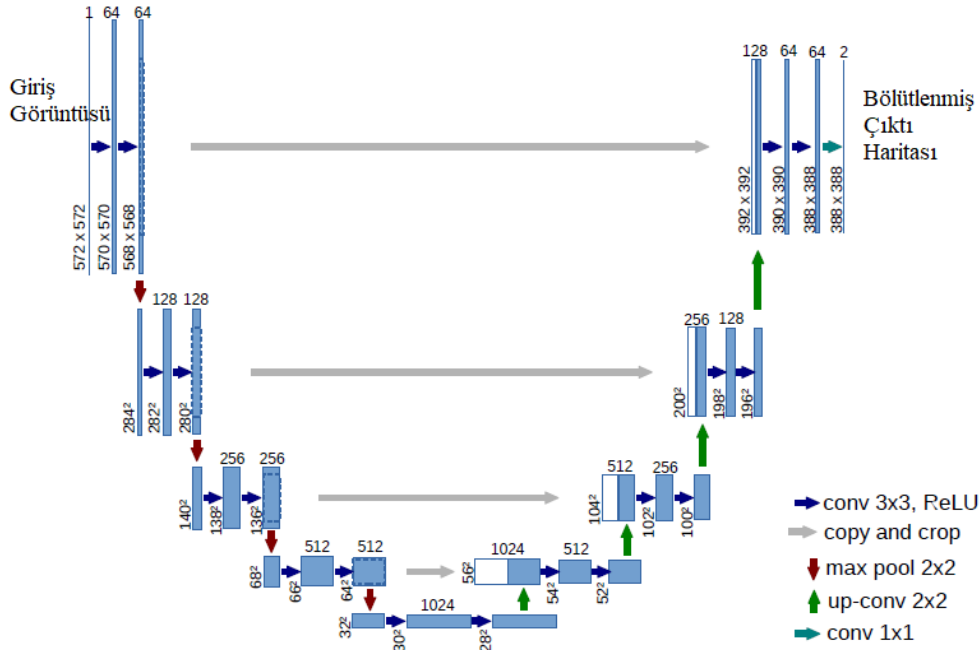
2.2. U-Net Modeli

U-Net, özellikle biyomedikal görüntüler üzerinde daha iyi bir bölütleme yapılması için 2015 yılında Olaf Ronneberger, Philip Fischer ve Thomas Brox tarafından "[U-Net: Convolutional Networks for Biomedical Image Segmentation](#)" isimli bir makale ile tüm dünyaya tanıtılmıştır (Ronneberger ve ark., 2015).

Klasik evrişimli sinir ağı modellerinde eğitim için büyük veri setlerine ihtiyaç duyulmaktadır. Bu veri setleri içerisindeki görüntüler etiketlenerek ağa sunulur ve ağı, görüntüleri bu etiket bilgileri ile tanır. Biyomedikal görüntülerde bu işlem (etiketleme) oldukça güçtür. Çünkü biyomedikal görüntüler piksel temelli yaklaşımlar gerektirmektedir. Böyle bir etiketleme ise oldukça büyük bir insan ve donanım kapasitesi gerektirmekle birlikte çok zaman almaktadır. U-Net ise bu konuda klasik evrişimli sinir ağlarından farklı bir mimari ve piksel bazlı görüntü bölütleme imkânı sunmaktadır.

2.1.1. U-Net Mimarisi

U-Net Mimarisi, Şekil 2'de gösterilmektedir. Çalışmada kullanılan modelin anlaşılması için mimarinin teknik açıdan detaylı ve sade bir şekilde açıklanmasında fayda görülmektedir.

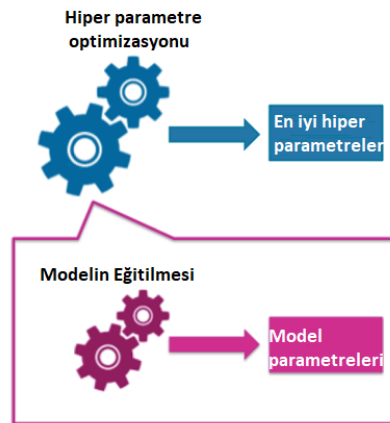


Şekil 2. U-Net Mimarisi (Ronneberger vd., 2015)

Şekil 2’de gösterilen U-net mimarisi (en düşük çözünürlükte 32x32 piksel için örnek). Her mavi kutu, çok kanallı bir özellik haritasına karşılık gelir. Kanal sayısı kutunun üstünde gösterilir. X-y boyutu kutunun sol alt kenarında bulunur. Beyaz kutular kopyalanan özellik haritalarını temsil eder. Oklar farklı işlemleri gösterir. U-Net modeli, bir büzülme yolu (contracting path) ve bir genişleme/açılım yolundan (expansive path) meydana gelmektedir. Şekil 3’te gösterilen ve sol tarafta yer alan büzülme yolu, klasik bir evrişimsel ağın tipik mimarisini izlemektedir. 3 x 3’lük bir evrişim işleminin iki defa tekrarlanmasından oluşur. Daha sonra bu işlemi, aktivasyon olarak geri dönüşümlü doğrusal birim fonksiyonu (ReLU) ve alt örnekleme (downsampling) için 2 kaydırmalı (stride) adım ile 2 x 2’lik bir maksimum ortaklama (max pooling) işlemleri takip eder. Her alt örnekleme işleminde özellik kanallarının (feature channel) sayısı ikiye katlanmaktadır. Şekil 2’de sağ tarafta yer alan genişleme yolundaki her adım yukarı örneklemesinden (upsampling) sonra özellik kanallarının sayısını yarıya indiren 2 x 2’lik bir evrişim işleminden, büzülme yolundan karşılık gelen kırılmış özellik eşlemesiyle birleştirme işleminden ve 3 x 3’lük bir evrişim işleminin iki defa uygulanmasıyla oluşmaktadır. Bu evrişim işlemlerinde aktivasyon olarak ReLU fonksiyonu kullanılmaktadır. Son katmanda ise 64 bileşenli özellik vektörü, istenen sayıda sınıfa eşlenmek için 1 x 1’lik bir evrişim işlemine tâbi tutulmaktadır. Toplamda, bu ağın 23 evrişim katmanı bulunmaktadır (Ronneberger ve ark., 2015).

2.3. Hiper Parametre Optimizasyonu

Hiper parametre, üzerinde ittifak sağlanmış bir tanımlı olmamasıyla birlikte bir soruna çözüm üretmek için kullanılan veri setine ve modele göre farklılık gösteren parametrelerdir. Hiper parametre optimizasyonunun amacı, herhangi bir sinir ağı modelinde, istenilen başarı kriterinden elde edilen sonucun en iyilenmesidir. İyi bir modelin inşa edilmesinde doğru bir modelin seçilmesi ve/veya hiper parametrelerin tercih edilmesi yeterli olmamaktadır. Bu hiper parametreler arasında en iyi sonucu verecek olan kombinasyonun bulunması, bilinmesi de gerekmektedir. Şekil 3’te hiper parametre optimizasyonu basit bir şekilde açıklanmaktadır.



Şekil 3. Model Üzerinde Hiper Parametre Optimizasyonu Gerçekleştirilmesi (<https://medium.com/deep-learning-turkiye>)

2.3.1. Izgara Arama

Izgara Arama (Grid Search), hiper parametre optimizasyonu için en sık kullanılan stratejilerden biridir. Kullanılan farklı modellere göre hiper parametreler bazen sonsuz denebilecek sayıda değer alabilirler. Böyle durumlarda eğitimci, mevcut bilgilerine dayanarak hiper parametreler için bir değer aralığı belirleyebilir. Belirlenen aralıklarda belli başlı noktalar-değerler seçilerek bir hiper parametre dizisi oluşturulabilir. Izgara arama sayesinde bu dizi içerisinde yer alan tüm değerlerin kombinasyonu için tüm ağ eğitilir ve en iyi kombinasyona sahip değerler hiper parametre olarak seçilir (Bergstra ve Bengio, 2012).

2.3.2. Optimize Edilen Hiper Parametreler

Çalışmanın bu bölümünde, U-net modelinin optimizasyonunda kullanılan hiper parametreler verilmektedir. Bu hiper parametrelerin tanıtımına geçilmeden önemli bir bilginin hatırlatılmasında fayda görülmektedir. Evrişimli Sinir Ağlarının hiper parametrelerinin optimizasyonu ile U-net modelinin hiper parametre optimizasyonunda kullanılan hiper parametreler arasında fark bulunmaktadır. Bunun en önemli sebebi U-net modelinde klasik Evrişimli Sinir Ağlarının son katmanı olan Tam/Tüm Bağlantı Katmanının (Fully Connected Layer) bulunmamasıdır.

Özellikle Evrişimli Sinir Ağları tabanlı modellerde hiper parametre optimizasyonunun yapılmasının en önemli sebebi, eğitimci modeline ve dolayısıyla veri setine en uygun yani en yüksek doğruluk oranını verebilecek olan yapay sinir ağları (YSA) katmanlarının (Python'da "Dense" olarak adlandırılmaktadır) ve bu katmanlarda kullanılacak nöron sayılarının bulunmasıdır. Ancak daha önce de belirtildiği gibi U-net modelinin yapısında YSA katmanları bulunmamaktadır. Bu sebeple bu çalışmada yapılan U-net hiper parametre optimizasyonunda aşağıda verilen hiper parametreler üzerinden bir optimizasyon gerçekleştirilmiştir. Bu hiper parametreler:

Parça boyutu (batch_size), yapay zekâ uygulamalarında, çalışılan veri seti içerisinde yer alan tüm verilerin aynı anda işlenmesi ve öğrenmenin bu şekilde gerçekleşmesi özellikle bellek ve zaman açısından epey maliyetlidir. Çünkü, öğrenme işlemi sırasında her bir İterasyon için "geriye yayılım" (backpropagation) ve gradyan hesaplaması yapılır. Bunlar sayesinde ağırlık değerleri güncellenir ve elde edilen hata oranı minimize edilmeye çalışılır. Bir veri seti ne denli büyükse, bu işlemler de o kadar uzun sürer. Bunun için veriler tek tek değil de küçük gruplar halinde işlenir. Bu küçük gruplara "mini-batch" denilmektedir (Gülcü ve Kuş, 2019).

Devir (epoch) sayısı, yapay zekâ algoritmalarında bütün bir öğrenme işlemi gerçekleştirirken yapılan tüm işlemler bittiğinde bu 1 devir (epoch) olarak adlandırılmaktadır (Gülcü ve Kuş, 2019).

Optimizasyon yöntemi (optimizer), doğrusal olmayan sorunların çözümü için en uygun değerlerin bulunmasını sağlamaktadır (Gülcü ve Kuş, 2019).

2.3.2. Hiper Parametre Optimizasyonu için Kullanılan Bazı Modüller

Python programlama dilinde hiper parametre optimizasyonunun sağlanması için kişi, kurum (üniversite, şirket vb.) bazlı oluşturulmuş farklı modüller bulunmaktadır. Bu modüller: hyperopt (<https://github.com/hyperopt>), skopt (<https://pypi.org>), sherpa (<https://parameter-sherpa.readthedocs.io/>), tune (<https://docs.ray.io/>), optuna (<https://optuna.org/>), keras-tuner (<https://github.com/keras-team/>) ve talos (<https://github.com/autonomio>) olarak özetlenebilir. Elbette farklı amaçlara için hazırlanmış ve daha da geliştirilmiş modüllerde mevcuttur. Ancak bu çalışmada bu modüller sadece isim olarak tanıtılmaktadır.

Bu çalışmada hiper parametre optimizasyonunun gerçekleştirilmesi için özellikle tensorflow (<https://www.tensorflow.org/>) modülü ile uyum sorununa çözüm getirdiğinden dolayı talos 1.0 kullanılmıştır. Çalışmada editör olarak, Python programlama dili için PyCharm Community Edition 2019.3.1 kullanılmıştır.

2.4. Optimizasyon Yöntemleri

Bu bölümde, çalışmada literatürde kullanılan popüler optimizasyon yöntemlerinden Stokastik Gradyan İniş (SGD), AdaDelta, RMSProp ve Adam açıklanmaktadır.

Stokastik Gradyan İniş: Gradyan inişi, bir optimizasyon yöntemidir. Gradyan ise bir fonksiyonun eğimi olarak adlandırılabilir. Yani, bir değışkende meydana gelen değışikliklere karşılık olarak bir değışkenin değışim derecesini ölçmektedir. Gradyan iniş-çıkışı dış büyük bir fonksiyondur ve eğimin büyüklüğü ile eğimin dikliğı doğru orantılıdır. SGD optimizasyon yönteminde, veri setinin tamamı yerine o veri setinden rastgele bir parça seçilir ve maliyet fonksiyonu bu parçanın üzerinden hesaplanır. Standart Gradyan inişe göre çok daha az maliyetlidir. Ancak hedef fonksiyonun en azlanması için çok fazla sayıda yineleme gerektirmektedir (Bottou, 2010).

AdaDelta: Uyarlanabilir öğrenme oranına dayalı bir SGD yöntemidir. AdaDelta, eğitim boyunca geçmiş gradyanları depolamak yerine, güncelleme yapıldığında bile öğrenmeye devam edebilmek için öğrenme oranlarını hareketli gradyan güncellemelerine uyarlayan bir optimizasyon yöntemidir (Zeiler, 2012).

RMSProp: Tıpkı AdaDelta gibi çok ciddi bir şekilde azalan öğrenme oranı sorununa çözüm bulmak amacıyla geliştirilmiştir. İki temel işlem ile açıklanabilir: Gradyan karesinin hareketli ortalaması korunur ve gradyan bu ortalamanın köküne bölünür (Hinton vd., 2012).

Adam: Eğitim verileri baz alınarak ağ ağırlıklarını güncellemek amacıyla standart SGD algoritması yerine kullanılabilen ve RMSProp ile momentumu birleştiren bir optimizasyon yöntemidir. Adam, hiper parametre optimizasyonunu sezgisel olarak yorumlayabilir ve bunun için çok az sayıda ayarlama gerektirmektedir (Kingma ve Ba, 2014).

2.5. U-Net Modelinin Hazırlanması

U-Net kullanılarak oluşturulan modelde 4 evrişim ve 4 ters evrişim işlemi ardışık olarak verilere uygulanmıştır. Tablo 2’de modele ait veriler yer almaktadır.

Tablo 2. Çekirdek Tespiti için Oluşturulan U-Net Model Verileri

Küçülme Yolunda Kullanılan Filtre Boyutu	Genişleme Yolunda Kullanılan Filtre Boyutu
16	128
32	64
64	32
128	16
256	1

Şekil 3’te verilen U-Net mimarisine uygun olarak hazırlanan modelde elde edilen çıktı, 1 filtrelik 1 x 1’lik bir evrişim işleminden geçirildikten sonra sigmoid aktivasyon fonksiyonuna tâbi tutulmuştur. U-Net mimarisi ile oluşturulan bu model daha sonra Adam optimizasyonu ile eğitilmiştir.

Evrişimsel sinir ağı modelleri bilgisayarların GPU’larını çok fazla meşgul ettiği ve uzun sürdüğü için *keras* kütüphanesinin sağladığı avantajlardan biri olan Erken Durdurma (Early Stopping) fonksiyonu kullanılmıştır. Bu fonksiyonun amacı, belli bir yitim fonksiyonu ve/veya doğruluk değerinin üzerine çıkılmayacağı anlaşıldığında algoritmanın çalışmayı bitirmesidir. Yani erken bitirmesidir.

Bu çalışmada, deneysel çalışmalar iki bölümde ele alınmaktadır. İlk olarak Adam optimizasyonu ile modelin kurulması ve elde edilen sonuçlar ışığında Adam optimizasyonunun hiper parametre optimizasyonu ile sonuçların en iyilenmesidir. Deneysel çalışmaların ikinci bölümünde ise hiper parametre optimizasyonu sonucunda elde edilen en iyi hiper parametrelerin diğer optimizasyon yöntemlerine uygulanarak optimizasyon yöntemlerinin karşılaştırılmasıdır.

3. Araştırma Sonuçları ve Tartışma

3.1. Çalışmanın Adam Optimizasyonu ile Gerçekleştirilmesi

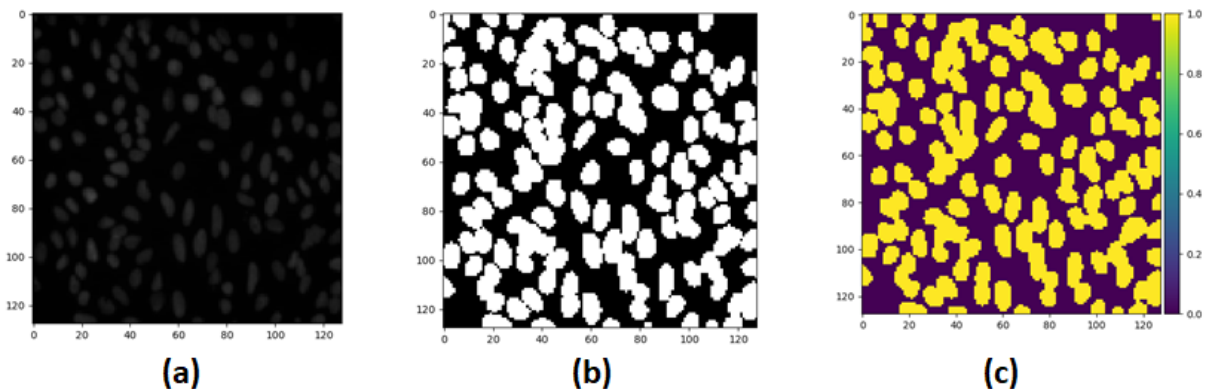
Çekirdek tespiti için oluşturulan modelin ilgili veri seti içerisindeki eğitim verilerinin tamamını çalıştırması bir tur sayısı (epoch) olarak değerlendirildiği için çalışmadaki epoch sayısı 100 ve batch size değeri 16 olarak belirlenmiştir ancak erken durdurma fonksiyonu 21. tur sayısı tamamlandığında bitirilmiştir.

Çalışmanın sonunda elde edilen yitim fonksiyonu ve doğruluk değeri Tablo 3’te verilmektedir.

Tablo 3. Eğitim Verileri için Modele ait Yitim Fonksiyonu ve Doğruluk İstatistikler

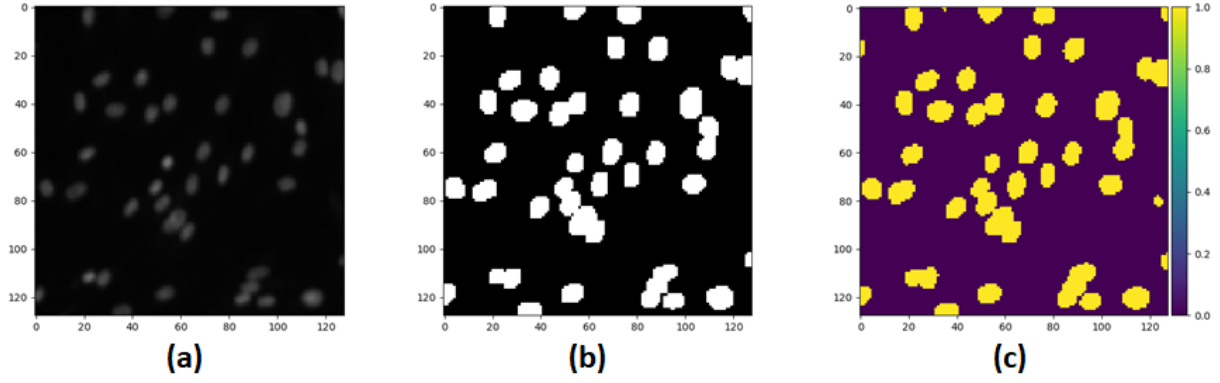
Yitim (Loss) Fonksiyonu	Doğruluk (Accuracy) Değeri
0.0828	0.9677

Çalışmada kullanılan eğitim görüntüsü, eğitim maskesi görüntüsü ve modelin uygulandığı eğitim görüntüsü sırası ile Şekil 4’te gösterilmektedir. Eğitim görüntüsüne ait maske görüntüleri birleştirilerek tek bir PNG dosyasında toplanmıştır.



Şekil 4. (a) Eğitim Verisinin Orijinal Gösterimi, (b) Eğitim Verisine ait Maske Görüntüsü, (c) Modelin Orijinal Görüntüye Uygulanması

Modelde, bir eşik değeri (threshold) oluşturulmuş olup bu değer 0.5 olarak belirlenmiştir. Bu değerinin altında kalan pikseller arka plan rengi olan mor rengine maskelenmiş, bu eşik değerini geçen pikseller ise sarı renk olarak maskelenmiştir. 670 görüntüden oluşan eğitim verisinin %10’u (67 görüntü) doğrulama verisi olarak seçilmiştir. Model, doğrulama verilerine uygulandığında elde edilen sonuçlar sırası ile Şekil 5’te gösterilmektedir.



Şekil 5. (a) Doğrulama Verisine ait Orijinal Görüntü, (b) Doğrulama Verisine ait Maske Görüntüsü, (c) Modelin Doğrulama Verisine Uygulandığında Elde Edilen Görüntü

Tablo 4'te modelin Doğrulama verisine uygulandığında elde edilen sonuçlar verilmektedir.

Tablo 4. Modelin Uygulandığı Doğrulama Verisine ait İstatistikler

Yitim (Loss) Fonksiyonu	Doğruluk (Accuracy) Değeri
0.0836	0.9675

3.1.1. Adam Optimizasyon Yönteminde Hiper Parametre Optimizasyonu

Çalışmanın bu bölümünde, U-net modelinde Adam optimizasyon yönteminde hiper parametre optimizasyonu için belirlenen hiper parametrelere ait değerlerin sonucunda elde edilen doğruluk oranları Tablo 5'te verilmektedir.

Tablo 5. Adam Optimizasyonunda Hiper Parametre Optimizasyonu için Seçilen Hiper Parametrelere ait Değerler

Batch size	Epoch
[16, 32]	[10, 20, 30, 40, 50, 60, 70, 80, 90, 100]

Tablo 5'te verilen değerler ile ilgili olarak; Batch size hiper parametresi için 2 değer belirlenmiştir (16 ve 32). Epoch hiper parametresi için de 10 değer belirlenmiştir (10'dan 100'e kadar 10'ar 10'ar artışla 10 değer).

Hiper parametrelere ait belirlenen bu değerlerin algoritma çalışma esnasında optimizasyonu şu şekilde olacaktır. Örneğin; algoritma da Batch size değeri 16 iken 10 defa çalışacak (10 Epoch) ve bir doğruluk değeri verecektir. Daha sonra 10'ar artışla en son 100 defa çalışacak (100 Epoch) ve yine bir doğruluk değeri verecektir. Sonrasında ise Batch size 32 iken bu işlemler aynen tekrarlanacaktır ve bunlar sonucunda bir doğruluk değeri üretilecektir.

Ayrıca, bu doğruluk oranlarının önceki çalışmada elde edilen doğruluk oranı da verilmektedir. Tablo 6'da Adam optimizasyonu ile batch size değeri 16 alındığında Epoch sayısı değiştirilerek optimize edilen U-net modeline ait sonuçlar verilmektedir.

Tablo 6. Adam Optimizasyonu ile Batch Size Değeri 16 alındığında Yapılan Optimizasyon Sonucu Elde Edilen Yitim (Loss) ve Doğruluk (Accuracy) Sonuçları

Train Loss	Train Accuracy	Val_Loss	Val_Accuracy	Epoch
0.1099	0.9581	0.1278	0.9508	10
0.0796	0.9689	0.1050	0.9572	20
0.0535	0.9782	0.0892	0.9652	30
0.0677	0.9728	0.1006	0.9639	40
0.0661	0.9734	0.0874	0.9698	50
0.0476	0.9802	0.0999	0.9697	60
0.1230	0.9525	0.1149	0.9553	70
0.1022	0.9606	0.0973	0.9627	80
0.0978	0.9622	0.0865	0.9681	90
0.0879	0.9656	0.0796	0.9689	100

Tablo 6'da verilen; **Train Loss**: Eğitim verilerine ait yitim fonksiyonu değerine (Loss Function), **Train Accuracy**: Modelin eğitim verilerini doğru tahmin etme oranına, **Val_Loss**: Doğrulama (validation) verilerine ait yitim fonksiyonu değerine, **Val_Accuracy**: Modelin doğrulama verilerinin doğru tahmin etme oranına karşılık gelmektedir.

Tablo 6'da verilen doğrulama verilerininin doğru tahmin edilmesinde en büyük doğruluk oranını veren Epoch sayısı Tablo 7'de verilmektedir.

Tablo 7. Adam Optimizasyonu ile Batch Size Değeri 16 alındığında Yapılan Optimizasyon Sonucu Elde Edilen En Yüksek Doğruluk Oranını Veren Epoch Değeri

Val_Accuracy	Epoch
--------------	-------

0.9698	50
--------	----

Tablo 8’de ise Adam optimizasyonu ile batch size değeri 32 alındığında Epoch sayısı değiştirilerek optimize edilen U-net modeline ait sonuçlar verilmektedir.

Tablo 8. Adam Optimizasyonu ile Batch Size Değeri 32 alındığında Yapılan Optimizasyon Sonucu Elde Edilen Yitim (Loss) ve Doğruluk (Accuracy) Sonuçları

Train Loss	Train Accuracy	Val_Loss	Val_Accuracy	Epoch
0.0444	0.9816	0.0913	0.9660	10
0.0413	0.9829	0.0983	0.9656	20
0.0375	0.9844	0.1034	0.9651	30
0.0435	0.9819	0.1020	0.9701	40
0.0406	0.9830	0.1077	0.9703	50
0.0368	0.9847	0.1132	0.9700	60
0.0817	0.9677	0.0667	0.9737	70
0.0795	0.9684	0.0651	0.9734	80
0.0767	0.9693	0.0790	0.9677	90
0.0774	0.9692	0.0676	0.9739	100

Batch size değeri 16 alındığı zaman %97 ve üzeri bir orana hiç çıkılamamıştı. Ancak Batch size değeri 32 alındığında birçok yerde bunun aşılabildiği görülmektedir. Tablo 8’de verilen sonuçlar içerisinde doğrulama verilerinin doğru tahmin edilmesinde en büyük doğruluk oranını veren Epoch sayısı Tablo 9’da verilmektedir.

Tablo 9. Adam Optimizasyonu ile Batch Size Değeri 32 alındığında Yapılan Optimizasyon Sonucu Elde Edilen En Yüksek Doğruluk Oranını Veren Epoch Değeri

Val_Accuracy	Epoch
0.9739	100

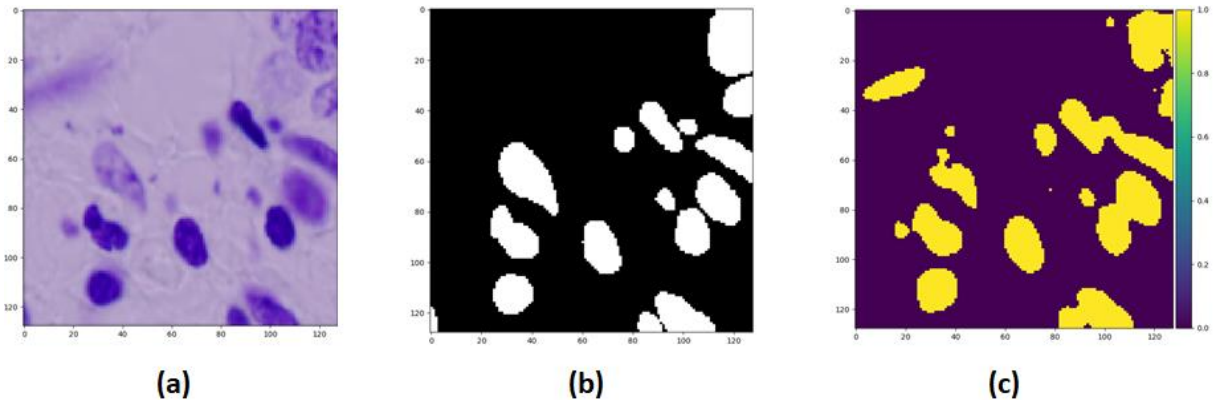
Daha önceki çalışmada Adam optimizasyonu kullanılarak elde edilen doğruluk oranı ile bu çalışmada Adam optimizasyonu kullanılarak gerçekleştirilen hiper parametre optimizasyonlarında batch size ve Epoch değerine göre elde edilen en yüksek doğruluk oranlarının karşılaştırılması Tablo 10’da verilmektedir.

Tablo 10. Adam Optimizasyon Yönteminde Elde Edilen Doğruluk Oranı ve Buna ait Hiper Parametreler ile Optimize Edilen Hiper Parametrelere ait Doğruluk Oranlarının Karşılaştırılması

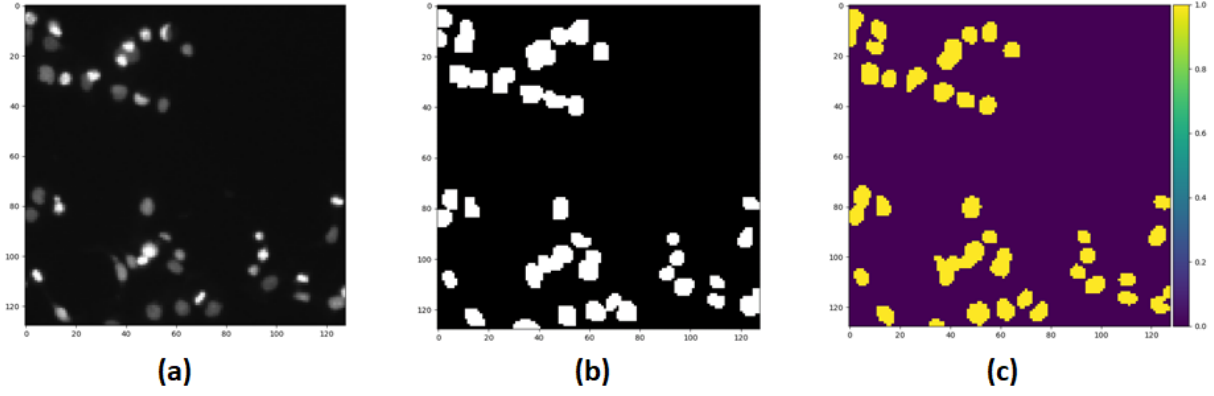
Sıralama	Batch Size	Epoch	Val_Accuracy
1	16	100	0.9675
2	16	50	0.9698
3	32	100	0.9739

Tablo 10’da ilk çalışmadan şu ana dek elde edilen sonuçlar sıralı bir şekilde verilmektedir. Bu göre ilk satırda verilen değerler önceki çalışmaya ait, 2. ve 3. Satırlarda verilen değerler ise bu çalışmada elde edilen verilerdir. Tablo 10’da da görüldüğü gibi Adam optimizasyonunda en yüksek doğruluk oranı batch size 32 değerinde iken 100 Epoch ile çalıştırıldığında elde edilmektedir. Çalışmanın sonraki bölümünde ise, optimizasyon yöntemlerinin karşılaştırılması verilmektedir.

Şekil 6’da hiper parametre optimizasyonu sonucunda elde edilen hiper parametre değerleri, Şekil 7’de ise modelin görüntü verilerine uygulandığı zaman elde edilen sonuçlar gösterilmektedir. 2018 Data Science Bowl veri setinden alınan görüntülere ait şekillerin boyutu 128 x 128 olarak ayarlanmıştır.



Şekil 6. (a) Eğitim Verisine ait Orijinal Görüntü, (b) Eğitim Verisine ait Maske Görüntüsü, (c) U-net Modeli ile Tahmin Edilen Eğitim Görüntüsü



Şekil 7. (a) Doğrulama Verisine ait Orijinal Görüntü, (b) Doğrulama Verisine ait Maske Görüntüsü, (c) U-net Modeli ile Tahmin Edilen Doğrulama Görüntüsü

3.2. Diğer Optimizasyon Yöntemlerine ait Sonuçların Karşılaştırılması

Çalışmanın bu bölümünde, Adam optimizasyonu kullanılarak, doğrulama verilerinde en yüksek doğruluk oranını veren hiper parametre değerleri (batch size = 32, Epoch = 100) diğer optimizasyon yöntemleri (SGD, RMSprop, Adadelta) ile çalıştırılmıştır. Her ne kadar bu çalışma özellikle doğrulama verilerine ait doğruluk oranlarının iyileştirilmesi için optimize edilmek amacıyla hazırlanmış olsa da çalışmanın bu bölümünde elde edilen sonuçlar Tablo 11’de verilmektedir.

Tablo 11. 4 Optimizasyon Yönteminin 100 Epoch ve 32 Batch Size Hiper Parametreleri ile Çalıştırılmasında Doğrulama Verilerine ait Sonuçlar

Optimizasyon	Val_Accuracy	Val_Loss
Adam	0.9739	0.0676
SGD	0.9461	0.1348
RMSProp	0.9137	0.2152
AdaDelta	0.9484	0.1298

Tablo 11’de verilen optimizasyon yöntemlerinin karşılaştırılmasında yine en başarılı sonucu Adam optimizasyonu vermektedir. Diğer üç optimizasyon yöntemi de aynı hiper parametreler ile çalıştırılmıştır.

Araştırma sonuçları incelendiğinde Adam optimizasyon yönteminin mevcut hiper parametreler ile çalıştırıldığında ve bunların optimizasyonu sonucunda diğer optimizasyon yöntemlerine göre en yüksek doğruluk oranını verdiği görülmektedir. Hiper parametre optimizasyonunun görüntü bölütlemesinde doğruluk oranını arttırdığı görülmektedir.

4. Sonuç

Bu çalışmada, U-Net modeli kullanılarak yüksek bir doğruluk oranı ve düşük bir hata oranı ile çekirdek tespitinin otomatikleştirilmesi amaçlanmaktadır. Bu amacı gerçekleştirmek için deneysel çalışmaların iki aşama olarak belirlendiği bu çalışmada, ilk olarak Adam optimizasyonu kullanıldıktan sonra hiper parametre optimizasyonu ile hem mevcut optimizasyon yönteminin en iyilenmesi hem de diğer optimizasyon yöntemlerinin aynı hiper parametreler ile çalıştırılarak bir kıyaslama yapılmaktadır.

Hiper parametre optimizasyonu sonucunda batch size = 32 ve Epoch = 100 iken 0.9739 oranı ile en yüksek doğruluk değeri elde edilmiştir. Daha sonra diğer optimizasyon yöntemleri olan ve tensorflow modülü ile erişim sağlanan SGD, RMSprop ve Adadelta optimizasyon yöntemleri ile daha önce belirlenen en iyi hiper parametre değerleri kullanılarak aynı model çalıştırılmıştır. Bu işlemin sonucunda ise yani ikinci bir optimizasyon işleminin sonucunda diğer optimizasyon yöntemleri aynı hiper parametreler ile çalıştırıldığında elde ettikleri doğruluk oranı Adam optimizasyonuna göre daha düşük kalmıştır. Bu sonuç, ilk çalışmada optimizasyon yöntemi olarak belirlenen Adam optimizasyonunun tercih edilmesindeki başarıyı ortaya koymaktadır. Ayrıca, bu çalışma için en iyi optimizasyon yöntemi olan Adam optimizasyonunun da en yüksek doğruluk sonucunu üretmesini sağlayan en iyi hiper parametreler de hiper parametre optimizasyonu ile bulunmuştur. Özellikle görüntü işleme de belirli bir süreden (Epoch sayısından) sonra öğrenme işleminde doğruluk oranının artması oldukça zordur. Bu sebeple, çalışmada özellikle %97 den sonra doğruluk oranının artmasının çok uzun bir süre ve dolayısı ile birkaç Epoch’tan sonra olması gibi sebeplerden dolayı önceki çalışmaya (0.9675) nazaran elde edilen bu doğruluk oranı (0.9739) kısmen de olsa anlamlı bulunmaktadır. Ayrıca diğer optimizasyon yöntemlerinin elde ettiği doğruluk oranlarına göre Adam optimizasyonunun ve batch size ile Epoch gibi hiper parametrelerin tercih edilmesi de isabetli bulunmaktadır.

U-net modelinin diğer Evrişimli Sinir Ağları modellerinden en önemli farklılığı olan son katmanda YSA katmanlarının yer almaması ve sadece evrişim (convolution) işlemi ile ortaklama (pooling) işlemlerinin yapılması, U-net hiper parametre optimizasyonunu oldukça güçlendirmektedir. Çünkü görüntü işleme de özellikle Evrişimli Sinir Ağları modellerinde yapılan hiper parametre optimizasyonunun amacı, özellikle YSA katmanları ile bağlantıyı sağlayan Fully Connected Layer katmanında kaç adet YSA katmanı ve bu katmanlarda kaç adet nöron kullanılacağı, bunların hangi aktivasyon fonksiyonu ile en iyi sonucu vereceği gibi

sorunlara çözüm bulmaktır. Ancak U-net modelinde YSA katmanlarının olmaması ve kullanılan aktivasyon fonksiyonunun da model itibari ile zaten tanımlanmış olmasından dolayı U-net modelinde hiper parametre optimizasyonu oldukça kısıtlıdır.

Çalışmada kullanılan mevcut veri setinin hem sezgisel algoritmalar hem de U-net dışında farklı bir model ile çalıştırılması ve elde edilen sonuçların karşılaştırılmasının tıbbi görüntü alanında hastalık tespiti ve bunun otomatikleştirilmesine katkı sağlaması açısından oldukça önemli olduğu düşünülmektedir.

Kaynakça

- Alom, M. Z., Yakopcic, C., Taha, T. M., & Asari, V. K. (2018). Nuclei segmentation with recurrent residual convolutional neural networks-based U-Net (R2U-Net). In *NAECON 2018-IEEE National Aerospace and Electronics Conference* (pp. 228-233). IEEE.
- Atli, İ., & Gedik, O. S. (2019). VESUNETDeep: A Fully Convolutional Deep Learning Architecture for Automated Vessel Segmentation. In *2019 27th Signal Processing and Communications Applications Conference (SIU)* (pp. 1-4). IEEE.
- Bergstra, J., & Bengio, Y. (2012). Random search for hyper-parameter optimization. *The Journal of Machine Learning Research*, 13(1), 281-305.
- Bottou, L. (2010). Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010* (pp. 177-186). Physica-Verlag HD.
- Chidester, B., Ton, T. V., Tran, M. T., Ma, J., & Do, M. N. (2019). Enhanced rotation-equivariant U-Net for nuclear segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops* (pp. 0-0).
- Ciresan, D., Giusti, A., Gambardella, L. M., & Schmidhuber, J. (2012). Deep neural networks segment neuronal membranes in electron microscopy images. In *Advances in neural information processing systems* (pp. 2843-2851).
- Data Science Bowl 2018 Veri Seti. <https://www.kaggle.com/c/data-science-bowl-2018> (Erişim Tarihi: 05.06.2020)
- Derin Öğrenmede Kullanılan Hiper Parametreler. <https://medium.com/deep-learning-turkiye/derin-ogrenme-uygulamalarinda-en-sik-kullanilan-hiper-parametreler-ece8e9125c4> (Erişim Tarihi: 05.06.2020)
- Fuat, T. Ü. R. K., Murat, L. Ü. Y., & BARIŞCI, N. (2019). Böbrek Tümör Segmentasyonu İçin Unet ve Unet-ResNet Modellerinin Karşılaştırılması. In *2019 3rd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)* (pp. 1-5). IEEE.
- Gonzalez, R. C., Woods, R. E., Telatar, Z., Tora, H., Arı, F., Kalaycıoğlu, E., & Somyürek, H. I. (2014). *Sayısal görüntü işleme: Üçüncü baskıdan çeviri*. Ankara: Palme Yayıncılık.
- Gülay, E., & Semra, İ. Ç. E. R. (2020). Evaluation of Lung Size in Patients with Pneumonia and Healthy Individuals. *Avrupa Bilim ve Teknoloji Dergisi*, 304-309.
- Gülcü, A., & Zeki, K. U. Ş. (2019). Konvülsiyonel sinir ağlarında hiper-parametre optimizasyonu yöntemlerinin incelenmesi. *Gazi Üniversitesi Fen Bilimleri Dergisi Part C: Tasarım ve Teknoloji*, 7(2), 503-522.
- Hinton, G., Srivastava, N., & Swersky, K. (2012). Neural networks for machine learning lecture 6a overview of mini-batch gradient descent. *Cited on*, 14(8).
- Hyperopt Hiper Optimizasyon Modülü. <https://github.com/hyperopt/hyperopt> (Erişim Tarihi: 04.06.2020)
- Keras-Tuner Hiper Optimizasyon Modülü. <https://github.com/keras-team/keras-tuner> (Erişim Tarihi: 04.06.2020)
- Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- Naylor, P., Laé, M., Reyat, F., & Walter, T. (2017). Nuclei segmentation in histopathology images using deep neural networks. In *2017 IEEE 14th international symposium on biomedical imaging (ISBI 2017)* (pp. 933-936). IEEE.
- Optuna Hiper Optimizasyon Modülü. <https://optuna.org/> (Erişim Tarihi: 04.06.2020)
- ÖZİÇ, M. Ü., & ÖZŞEN, S. (2020). Üç Boyutlu T1 Ağırlıklı Manyetik Rezonans Görüntülerinde Ön İşleme Yöntemleri. *Avrupa Bilim ve Teknoloji Dergisi*, (19), 227-240.
- Punn, N. S., & Agarwal, S. (2020). Inception U-Net Architecture for Semantic Segmentation to Identify Nuclei in Microscopy Cell Images. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 16(1), 1-15.
- Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention* (pp. 234-241). Springer, Cham. <https://arxiv.org/abs/1505.04597>
- Scikit-Learn Optimizasyon Yöntemi. <https://pypi.org/project/scikit-optimize/> (Erişim Tarihi: 04.06.2020)
- Sherpa Hiper Optimizasyon Modülü. <https://parameter-sherpa.readthedocs.io/en/latest/> (Erişim Tarihi: 04.06.2020)
- Şişeci, M., Metlek, S., & Cetişli, B. (2014). Alt-Bloklar Tekniği ve Kümeleme Yöntemleri İle Görüntü Bölütlemenin Hızlandırılması. *Journal of the Faculty of Engineering & Architecture of Gazi University*, 29(4).
- Talos Hiper Optimizasyon Modülü. <https://github.com/autonomio/talos/issues/451> (Erişim Tarihi: 04.06.2020)
- Tensorflow Kütüphanesi. <https://www.tensorflow.org/> (Erişim Tarihi: 04.06.2020)
- Tune Hiper Optimizasyon Modülü. <https://docs.ray.io/en/latest/tune.html> (Erişim Tarihi: 04.06.2020)
- Turan, S., & Bilgin, G. (2019). Meme Patolojisi Görüntüleri Üzerinde Derin Öğrenme ile Semantik Çekirdek Bölütleme Semantic Nuclei Segmentation with Deep Learning on Breast Pathology Images.
- Veta, M., Van Diest, P. J., Kornegoor, R., Huisman, A., Viergever, M. A., & Pluim, J. P. (2013). Automatic nuclei segmentation in H&E stained breast cancer histopathology images. *PLoS one*, 8(7), e70221.
- Vuola, A. O., Akram, S. U., & Kannala, J. (2019). Mask-RCNN and U-net ensembled for nuclei segmentation. In *2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019)* (pp. 208-212). IEEE.
- Wang, P., Hu, X., Li, Y., Liu, Q., & Zhu, X. (2016). Automatic cell nuclei segmentation and classification of breast cancer histopathology images. *Signal Processing*, 122, 1-13.
- Zeiler, M. D. (2012). ADADELTA: An Adaptive Learning Rate Method. Retrieved from <http://arxiv.org/abs/1212.5701>
- Zeng, Z., Xie, W., Zhang, Y., & Lu, Y. (2019). RIC-Unet: An improved neural network based on Unet for nuclei segmentation in histology images. *Ieee Access*, 7, 21420-21428.