

Retina Damar Segmentasyonunda Sinir Mimarisi Arama İçin Farklı Kodlama Yöntemlerinin Karşılaştırılması

Comparison of Different Encoding Methods for Neural Architecture Search in Retinal Vessel Segmentation

Zeki KUŞ¹ , Berna KİRAZ¹ 

Fatih Sultan Mehmet Vakıf Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, İstanbul, Türkiye

Öz

Retinal kan damar segmentasyonu diyabetik retinopati ve yaşa bağlı makula dejenerasyonu gibi göz hastalıklarının tespiti ve incelemesi açısından kritik bir görevdir. U-şekilli derin sinir ağlarının bu görev için başarılı sonuçlar verdiği bilinmektedir; fakat bu ağların optimize edilmesi gereken bir çok hiper-parametresi bulunmaktadır. Bu ağların otomatik bir şekilde optimizasyonu ve aranması için birçok sinir mimarisi arama (SMA) çalışması gerçekleştirilmiştir. SMA çalışmaları incelendiğinde seçilen kodlama şemalarının üretilen ağların karmaşıklığını ve performansını doğrudan etkilediği görülmüştür. Bu çalışmada, retinal kan damar segmentasyonu için sunduğumuz iki SMA çalışmasında (UNAS-Net ve MedUNAS) önerilen kodlama şemalarının performansları herkese açık olarak yayınlanan iki farklı retinal kan damar segmentasyonu veri kümesi üzerinde karşılaştırılmıştır. Elde edilen sonuçlara bakıldığında, önerilen her iki yöntemin temel alınan U-Net'ten 25 kata kadar daha az parametre ile tüm ölçütler açısından daha iyi performans gösterdiği görülmüştür. Ayrıca, SMA çalışmaları ile kıyaslandığında UNAS-Net'in 54,19 kata kadar daha az parametreyle ACC, SEN ve F1 değerleri açısından daha iyi sonuçlar ürettiği, benzer şekilde MedUNAS'ın 13,22 kata kadar daha az parametreye sahip model ile aynı metrikler açısından daha iyi sonuçlar elde edebildiği gösterilmiştir.

Anahtar Kelimeler: Retina Damar Segmentasyonu, Sinir Mimarisi Arama, Kodlama, Karşıtlık Temelli Diferansiyel Gelişim

Abstract

Retinal blood vessel segmentation is critical for detecting and investigating eye diseases such as diabetic retinopathy and age-related macular degeneration. U-shaped deep neural networks are known to give good results for this task, but these networks have many hyper-parameters that need to be optimized. Many neural architecture search (NAS) studies have been conducted to optimize and search these networks automatically. These studies have shown that the coding schemes used affect the complexity and performance of the generated networks. In this study, we compare the performance of the coding schemes proposed in two NAS for retinal blood vessel segmentation (UNAS-Net and MedUNAS) on two publicly released retinal blood vessel segmentation datasets. The results show that both methods outperform the baseline U-Net in all metrics with up to 25 times fewer parameters. Compared to NAS studies, UNAS-Net has shown better performance in terms of ACC, SEN, and F1 values with up to 54,19 times fewer parameters. Similarly, MedUNAS has achieved better results in terms of the same metrics with up to 13,22 times fewer parameters.

Keywords: Retinal Vessel Segmentation, Neural Architecture Search, Encoding, Opposition Based Differential Evolution

I. GİRİŞ

Medikal görüntü analizinde önemli bir görev olan retinal kan damarı segmentasyonu, fundus görüntülerinden kan damarlarının tespit edilmesini içermektedir. Retinal vasküler hastalıkların teşhis ve takibinde kritik bir rol oynadığı için, bu işlem özellikle diyabetik retinopati ve yaşa bağlı makula dejenerasyonu gibi hastalıkların teşhisi ve izlenmesi için kritik öneme sahiptir [1]. Ancak, kan damarlarının karmaşık yapısı ve değişken görünümü sahip olmasından dolayı retinal damar segmentasyonu zor bir işlemdir [2]. Bu sebeple, gürbüz ve etkili algoritmalar geliştirme ihtiyacı, retinal damar segmentasyonu problemini sıcak ve çok çalışılan bir konu haline getirmiştir.

Literatürde, U-Net [3] gibi U-şekilli derin sinir ağı modellerinin retinal damar segmentasyonu problemi için başarılı sonuçlar verdiği gözlemlenmiştir [4, 5, 6]. Bu nedenle U-Net ve Attention U-Net gibi varyasyonlar, yapılan segmentasyon çalışmalarında temel alınan yöntem olarak seçilmektedir. Ancak, U-Net optimize edilmesi gereken birçok hiper-parametreye sahiptir ve elle tasarlanan bir ağ olduğu için statik bir yapıya sahiptir.

Bu tasarım sorunları, retinal kan damarlarının karmaşık yapısı ve değişken görünümü nedeniyle farklı kalınlıklara sahip retinal damarların segmentasyonu için dezavantajlara neden olabilir [7]. Aynı zamanda, eğitilebilir parametre sayısının fazla olması eğitim ve çıkarım sürelerini arttırmaktadır. Bu durumlar sebebiyle, otomatik retinal damar segmentasyonu gerçekleştirecek sinir mimarilerinin araştırılmasına ihtiyaç duyulmaktadır.

Sinir mimarisi arama (SMA) çalışmaları, otomatik yapay öğrenme (AutoML) çalışmalarının bir alt kümesidir. Verilen bir problem için tanımlanan arama uzayı sınırları dahilinde seçilen arama yöntemi ile otomatik bir şekilde derin sinir ağı mimarisinin araştırılması hedeflenmektedir. SMA için pekiştirmeli öğrenme [8, 9], gradyan tabanlı [10] ve evrimsel yöntemlere [11, 12] dayalı yaklaşımlar önerilmiş ve arama maliyeti ve başarısı açısından evrimsel yöntemlerin daha iyi sonuçlar verdiği gözlemlenmiştir. Retinal kan damar segmentasyonu için birçok sinir mimarisi arama çalışması gerçekleştirilmiştir [13 - 15]. Bu çalışmalarda en iyi sinir ağı mimarilerini aramak için genetik algoritma gibi meta-sezgisel yöntemler kullanılmış ve başarılı sonuçlar verdiği gözlemlenmiştir [18].

Chen vd. (2021) [4] tarafından gerçekleştirilen çalışma U-Net [3] gibi U-şekilli derin sinir ağı modellerinin retinal damar segmentasyonu problemi için başarılı sonuçlar verdiğini göstermektedir. Zhou vd. tarafından yapılan çalışmada “*Çalışma Grubu Öğrenimi (SGL)*”[5] isimli yeni bir metod önerilmiş ve bu metodun gürültü içeren referans görüntüleri için temel alınan U-Net’in segmentasyon performansını hassasiyet, dice skoru, doğruluk ve AUC metrikleri açısından sırasıyla 8,5; 2,7; 0,2 ve 0,5 puan iyileştirdiği ve daha gürbüz hale getirdiği görülmüştür. Wentao vd. (2022) [6] tarafından yapılan çalışmada ise ince damar görüntüleri ve düşük ışıkta görüntüleme gibi önemli problemler için etkili bir şekilde segmentasyon gerçekleştirebilen yeni bir teknik tanıtılmış ve “*FR-UNet*” olarak isimlendirilmiştir. *FR-UNet*, yatay ve dikey yönde ağı genişletmek için çok çözünürlüklü bir evrişim mekanizması içerir ve piksel düzeyinde doğru bir temsil sağlamak için özellik toplama modülüne ve değiştirilmiş artık bloklara sahiptir.

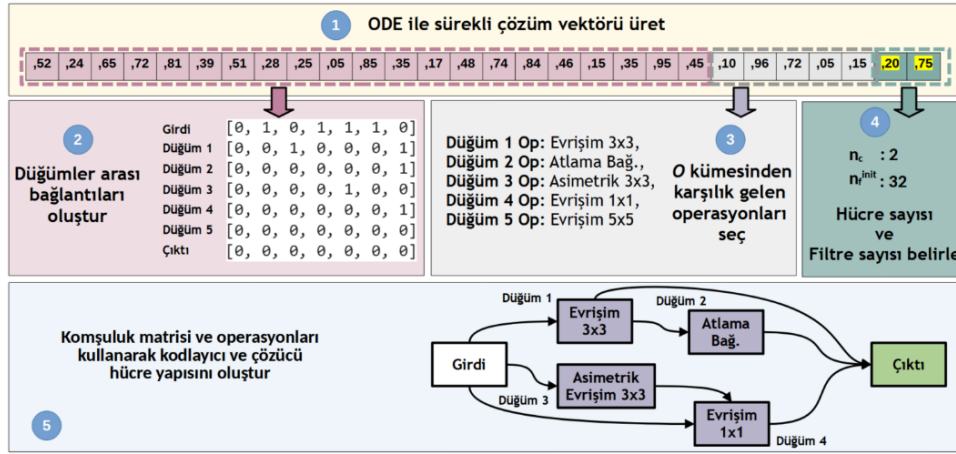
Bu çalışmalar yanında retinal damar segmentasyonu için üst-sezgiseller ile sinir mimarisi arama işlemi gerçekleştiren çalışmalar da yapılmaktadır. Houreh vd. (2021) [13] tarafından sunulan “*Hiper Sinirsel Mimari Arama (HNAS)*” yöntemi, retinal kan damarı segmentasyonu için genetik algoritma yöntemini kullanarak U-şekilli sinir ağları için SMA işlemi gerçekleştirmiştir. *HNAS*, *DRIVE* veri kümesi üzerinde test edilmiş ve daha az karmaşık ağlar ile

literatürdeki çalışmalara kıyasla yüksek rekabetçi sonuçlar elde edilmiştir. Wei vd. (2022) [14] tarafından gerçekleştirilen çalışmada ise retinal damar segmentasyonu için U-şeklinde sinir ağları üreten “*Genetik U-Net*” adında bir SMA yöntemi önerilmiştir. Önerilen yöntem, daha az parametre ile segmentasyon başarısını arttırmayı hedeflemiştir. Yüksek segmentasyon başarısına sahip mimarileri oluşturabilmek için, önerilen yöntem, genetik algoritmayı geliştirmiş ve daha az kısıtlı bir arama uzayı kullanmıştır. Deney sonuçları, Genetik U-Net kullanılarak elde edilen mimarilerin, diğer çalışmalara göre daha az karmaşık modeller ile daha yüksek segmentasyon başarısı elde edebildiğini göstermektedir. “*BTLBO*”[15], Rajesh vd. tarafından geliştirilen bir optimizasyon yöntemi olarak, evrimsel algoritmalarla birlikte arama algoritmaları arasında yer almaktadır. Önerilen çalışmada, SMA adımlarını hızlandırmak için arama uzayı, dikkat mekanizması ve önbellek sistemi ile U-şeklindeki mimarilerin otomatik optimizasyonu gerçekleştirilmiştir. “*BTU-Net*” olarak isimlendirilen bu yaklaşım farklı retinal kan damar veri kümeleri üzerinde değerlendirilmiş ve diğer çalışmalara kıyasla daha iyi segmentasyon performansı göstermiştir. Önerilen teknik, çeşitli retinal damar segmentasyonu veri kümelerinde test edilmiş ve daha az parametre ile literatürdeki çalışmalardan daha iyi performans göstermiştir.

Bu çalışmada, U-şekilli mimariler için SMA işlemi gerçekleştirilmiş ve önerdiğimiz bu mimarilerin oluşturulması için kullanılan farklı kodlama stratejilerinin performansı retinal kan damar segmentasyonu veri kümeleri üzerinde karşılaştırılmıştır. Farklı mimarilerin arama uzayında araştırılması için meta-sezgisel yöntem olarak “*Karşıtlık Temelli Diferansiyel Gelişim*” yöntemi kullanılmıştır. UNAS-Net [16] ve MedUNAS [17] yöntemlerinde kullanılan kodlama şemaları karşılaştırıldığında UNAS-Net çalışmasında önerilen stratejinin en az %50 daha az parametre ile rekabetçi sonuçlar elde ettiği gözlemlenmiştir.

II. MATERYAL VE METOD

Yapılan çalışmada, U-şekilli mimarilerin eniyilemesi için önerdiğimiz UNAS-Net [16] ve MedUNAS [17] yöntemleri karşılaştırılmıştır. Her iki yöntem de hücre tabanlı kodlayıcı ve kod çözücü hücrelere sahip U-şekilli bir yapıdır. Bu yapılar, hücre tabanlı mikro arama uzayı aracılığıyla hücrenin yapısını belirlerken, makro arama uzayı aracılığıyla toplam hücre sayısı ve ilk hücredeki özellik haritalarının sayısını belirlemektedir. Karşılaştırılan yapılar, kodlayıcı ve kod çözücü hücrelerden oluşur. Kodlayıcı hücreler özellikleri çıkarıp görüntü boyutunu azaltırken, kod çözücü hücreler çıkarılan özelliklerden görüntüyü



Şekil 1. UNAS-Net kodlama şeması adımları

yeniden inşa edip görüntü boyutunu artırır. Girdi görüntüsü bir dizi kodlayıcı ve kod çözücü hücreden geçirildikten sonra 2-boyutlu (2B) evrişim ve sigmoid aktivasyonu uygulanarak segmente edilmiş görüntü oluşturulur. Her iki yöntemde de kodlayıcı ve kod çözücü hücreler döngü içermeyen yönlü çizgeler (*DAG - directed acyclic graph* $G = (N, E)$) ile temsil edilmektedir. UNAS-Net ve MedUNAS, sinir mimarisi yapısı açısından benzer olsalar da kodlama şeması ve hücre yapısı açısından birbirinden farklılaşmaktadır. Ayrıca farklı mimarilerin arama uzayında araştırılması için Karşıtlık Temelli Diferansiyel Gelişim (Opposition-based differential evolution - ODE) algoritması kullanılmıştır. Bu bölümde bu iki yöntem için kullanılan arama uzayı ve kodlama şemalarından ve ODE algoritmasının genel yapısından bahsedilmiştir.

2.1. UNAS-Net

2.1.1 Arama uzayı

UNAS-Net [16] çalışmasında, arama uzayını iki kısma ayırdık. Birinci kısımda, kodlayıcı ve kod çözücü hücreleri oluşturan yapı için arama işlemi gerçekleştirilir ve mikro arama uzayı olarak isimlendirilir. Makro arama uzayı olarak isimlendirilen ikinci kısımda ise UNAS-Net yapısında kullanılacak hücre sayısı ve özellik haritalarının sayısı optimize edilir.

UNAS-Net yapısında hücreler, N adet düğüm ve E adet kenardan oluşur. Her bir hücre bir girdi, bir çıktı ve en fazla 5 adet ara düğümden (Şekil 1 mor renkli düğümler) oluşur. Bu yapıda, ara düğümler gelen girdiye uygulanacak işlemi/operasyonu temsil ederken, kenarlar ise düğümler arasındaki bilgi akışını göstermektedir. Her bir ara düğüm uygulanacak işlemi belirten bir $l \in L$ etiketine sahiptir ve bu kümedeki işlemler DARTS [12] makalesi referans alınarak seçilmiştir. L kümesindeki elemanlar Tablo 1'de gösterilmiştir.

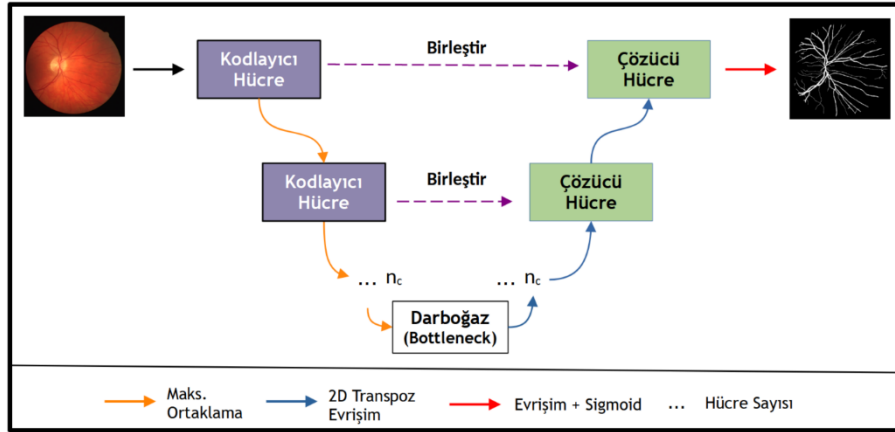
Tablo 1. Ara düğümler için seçebilecek olası işlemler

Operasyon Adı
1x1 2B Evrişim (1x1 conv)
3x3 2B Evrişim (3x3 conv)
5x5 2B Evrişim (5x5 conv)
7x7 2B Evrişim (7x7 conv)
3x3 2B Derinlemesine Ayrılabilir Evrişim (3x3 depconv)
5x5 2B Derinlemesine Ayrılabilir Evrişim (5x5 depconv)
7x7 2B Derinlemesine Ayrılabilir Evrişim (7x7 depconv)
3x3 2B Açılan Evrişim (3x3 dilconv)
5x5 2B Açılan Evrişim (5x5 dilconv)
7x7 2B Açılan Evrişim (7x7 dilconv)
3x3 2B Asimetrik Evrişim (3x3 asymconv)
5x5 2B Asimetrik Evrişim (5x5 asymconv)
7x7 2B Asimetrik Evrişim (7x7 asymconv)
Atlama bağlantısı (Skip-Connection - skip)

Arama uzayının küçültülmesi amacıyla düğüm ($|N| \leq 7$), kenar ($|E| \leq 9$) ve olası işlem sayısı ($|L| \leq 14$) sınırlandırılmıştır. N , E ve L kümeleri hücre tabanlı mikro arama uzayını oluştururken hücre sayısı ($n_c \in \{2, 3, 4, 5\}$) ve ilk hücredeki özellik haritalarının sayısını ($n_f^{init} \in \{8, 16, 32\}$) belirleyen iki karar değişkeni de makro arama uzayını oluşturur. UNAS-Net çalışmasından farklı olarak bu çalışmada daha iyi sonuç gösterdiği için girdi düğümünün çıktı düğümüne bağlı olması zorunlu tutulmuştur.

2.1.2 Kodlama şeması

Her bir çözüm vektörü toplamda 28 elemandan oluşan gerçek değerli vektörler ile temsil edilir. Bu çözüm vektöründeki ilk 21 eleman hücre yapısı içerisindeki düğümlerin birbirine nasıl bağlandığını, sonraki 5 eleman ara düğümler için seçilen operasyonları ve son iki eleman da sırasıyla makro arama uzayındaki n_c ve n_f^{init} karar değişkenlerini



Şekil 2. UNAS-Net ve MedUNAS için kullanılan U-şekilli mimarinin genel yapısı

temsil eder. Hücre yapısı DAG ile temsil edildiği için, DAG'ı tanımlamak amacıyla komşuluk matrisi kullanılmıştır. Maksimum 7 düğümden oluşan hücre yapısındaki komşuluklar, 7×7 'lik komşuluk matrisinin üst-üçgensel alanı ile temsil edilmiştir ve bu matriste 0 ve 1 değerleri iki düğümün birbirine bağlı olup olmadığını göstermek için kullanılır.

Gerçek değerlerden oluşan çözüm vektöründen UNAS-Net mimarisini oluşturmak için eşleme işlemi gerekmektedir. Şekil 1'de gösterilen bu eşleme adımları 5 adımda gerçekleştirilmektedir.

1. İlk adımda her biri $[0, 1]$ aralığında rastgele değerlerden oluşan 28 elemanlı çözüm vektörü oluşturulur.
2. Vektördeki ilk 21 eleman komşuluk matrisinin oluşturulması için kullanılır ve düğümler arasında bağlantı olup olmadığını belirtir. Bağlantı olup olmadığını ikili bir değer (0 ise bağlantı yok, 1 ise var) ile temsil edilebileceğinden $[0, 1]$ uzayı iki eşit parçaya bölünür: Çözüm vektöründe karşılık gelen gerçek değer $[0, 0,5)$ aralığında ise iki düğüm arasında bağlantı yok; $[0,5, 1)$ ise bağlantı vardır. Örneğin: Şekil 1 'de çözüm vektöründeki ilk eleman Girdi ve Düğüm 1 arasında bağlantı olup olmadığını temsil eder. Bu vektördeki ilk değer 0,52 olduğu için komşuluk matrisinin üst-üçgensel alanındaki ilk değer (birinci satır, ikinci sütun) 1 olur. Benzer şekilde, vektördeki ikinci değer 0,24 olup Girdi ile Düğüm 2 arasında bağlantı olmadığını temsil eder ve komşuluk matrisinin birinci satır, üçüncü sütunundaki değer 0 olarak işaretlenir.
3. Çözüm vektöründe bulunan sonraki 5 eleman her bir ara düğüm için seçilen işlemleri temsil eder ve bu değerler için yine benzer şekilde eşleme işlemi gerçekleştirilir. L kümesinde 14 olası işlem olduğu için $[0, 1]$ aralığı 14 eşit parçaya bölünür: $[0, 0,07)$ aralığı 1×1

filtre boyutuna sahip evrilim işlemine; $[0,07, 0,14)$ aralığı 3×3 filtre boyutuna sahip evrilim işlemine karşılık gelmektedir. Örneğin: Çözüm vektöründeki 22. eleman 0,10 olduğu için $[0,07, 0,14)$ aralığına düşer ve bu aralığa karşılık gelen 3×3 filtre boyutuna sahip evrilim işlemi Düğüm 1 için seçilen işlem olur. Diğer tüm düğümler için benzer işlemler gerçekleştirilir.

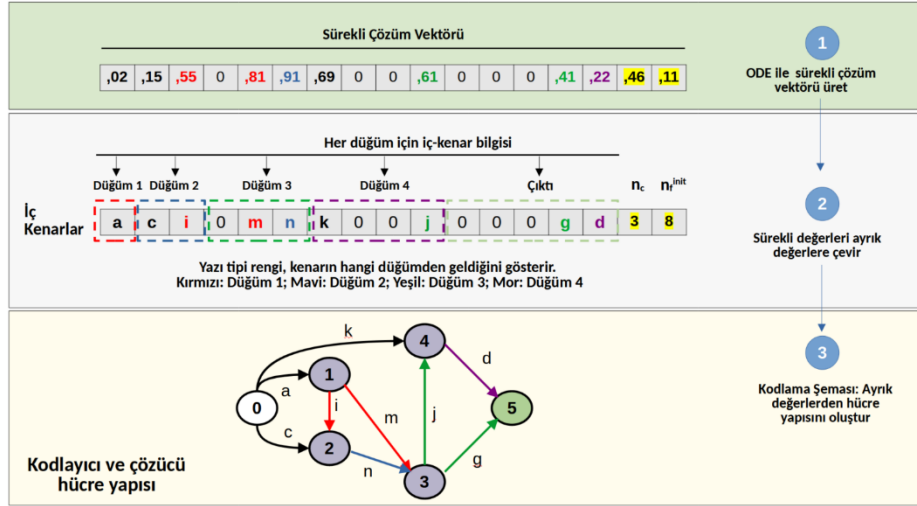
4. Çözüm vektöründeki son iki eleman ise n_c ve n_f^{init} karar değişkenlerine karşılık gelmektedir. n_c için seçilebilecek 4 eleman vardır. Bu nedenle $[0, 1]$ aralığı 4 eşit parçaya bölünür ($[0, 0,25)$ aralığı 2 değerine, $[0,25, 0,50)$ aralığı 3 değerine karşılık gelir vb.). Örneğin: Vektördeki 27. Eleman 0,20 olduğu için $[0, 0,25)$ aralığına düşer ve n_c için 2 değeri seçilir. Benzer şekilde n_f^{init} için ise seçilebilecek 3 değer vardır ve aralık 3 eşit parçaya bölünür ($[0, 0,33)$ aralığı 8; $[0,33, 0,66)$ aralığı 16 değerine karşılık gelir vb.). Benzer şekilde, çözüm vektöründeki 28. eleman 0,75 olduğu için $[0,66, 1)$ aralığına düşer ve n_f^{init} için 32 değeri seçilir.
5. Adım 2 ve 3'te oluşturulan komşuluk matrisi ve seçilen operasyonlar kullanılarak kodlayıcı ve kod çözücü hücrenin yapısı oluşturulur.

Şekil 1'de gösterilen tüm adımlar (1-5) tamamlandıktan sonra oluşturulan hücre yapısı kullanılarak Şekil 2'de gösterilen UNAS-Net mimarisi oluşturulur.

2.2. MedUNAS

2.2.1. Arama uzayı

MedUNAS yapısında, kodlayıcı ve kod çözücü hücreler, N adet düğüm, E adet kenar ve O adet operasyondan oluşur. Her hücre, bir girdi düğümü,



Şekil 3. MedUNAS kodlama şeması adımları

4 ara düğüm ve bir çıktı düğümü içermektedir. Ara düğümler farklı düğümlerden gelen çıktıların toplama işlemi ile birleştirildiği noktalar ve kenarlar ise ara düğümden gelen bilgiye uygulanan operasyonu belirtir. Her ara düğüm ($n \in N \setminus \{n_1\}$), kendinden önce gelen düğümlerden gelebilecek 2 iç-kenara (in-edge) sahiptir. Bu kenarlar için seçilebilecek 15 olası işlem (O) vardır ve bu işlemler $a-o$ arasındaki karakterler ile temsil edilir. O kümesindeki işlemler Tablo 1'de gösterilen L kümesinde aynıdır. L kümesinden farklı olarak o işlemi eklenmiştir ve bu işlem ise bir bağlantı olmadığını (2 iç kenar kısıtlamasını sağlarken karmaşıklığı azaltmak için kullanılır) temsil etmek için kullanılır.

MedUNAS mimarisi, kodlayıcı ve kod çözücü hücrelerin eşit sayıda bulunduğu bir mimaridir. Bu mimaride, hücre sayısı ve ilk hücredeki özellik haritalarının sayısı SMA'ya dahil edilen iki temel karar değişkenidir. Hücre sayısı ($n_c \in \{2, 3, 4, 5\}$), kodlayıcı ve kod çözücü hücre sayısını belirlemek için kullanılan bir değişkendir. İlk hücredeki özellik haritalarının sayısı ($n_f^{mit} \in \{8, 16, 32\}$) ise, ağın performansını etkileyen bir diğer değişkendir ve her maksimum havuzlama işleminden sonra iki katına çıkarılır.

2.2.2. Kodlama şeması

Dong vd. (2022) [18], işlem ve bağlantı bilgilerini temsil etmek için hem alfabe hem de "0" dizelerini kullanan bir çözüm kodlaması önermiştir. Yapılan çalışmada, önerilen bu kodlama şeması ODE için modifiye edilmiştir. Her bir çözüm 17 elemandan oluşan gerçek değerlere sahip bir vektör ($\in [0, 1]$) kullanılarak temsil edilir. Bu vektördeki ilk 15 değer, her bir düğüm arasındaki bilgi akışını ve her bir kenar için seçilen işlemleri ifade eder. Son iki değişken ise kodlayıcı/çözücü hücre sayısını (n_c) ve ilk hücredeki özellik haritası sayısını (n_f^{mit}) belirtir. Üretilen çözüm vektöründen, hücre yapısının nasıl oluşturulduğu Şekil 3'te gösterilmiştir.

Bu çözüm kodlamasında kenarlar işlemleri, düğümler ise farklı işlemlerden elde edilen çıktıları göstermektedir. Şekil 3'te " $a-ci-0mn-k00j-000gd$ " kodlaması kullanılarak oluşturulmuş örnek bir hücre yapısı gösterilmiştir. Çözüm kodlamasının ilk biti Düğüm 1'e gelen iç-kenar (in-edge) bilgisini ve bu iç-kenar için seçilen işlemi belirtir. Sonraki iki bit Düğüm 2'ye gelen iç-kenar bilgisi ve seçilen işlemleri belirtir ve bu şekilde devam eder. Son olarak, son beş bit Düğüm 5 için iç-kenar ve seçilen işlem bilgisini içerir. Örneğin, " $k00j$ " bitleri Düğüm 4 için dört olası iç-kenar bağlantısını (Düğüm 0, Düğüm 1, Düğüm 2, Düğüm 3) ve bu olası iç-kenar bağlantıları için seçilen işlemleri gösterir. " $k00j$ " bitlerine göre şu düğümler arasında bir bağlantı yoktur: Düğüm 1 – Düğüm 4; Düğüm 2 – Düğüm 4 (Şekil 3'e bakınız). Yine bu kodlama için; Düğüm 4, Düğüm 0 ve Düğüm 3'ten bilgi alır ve sırasıyla alınan bu bilgilere k ve j işlemlerini uygular. Uygulanan işlemler sonucunda elde edilen bilgiler ise toplama operatörü ile birleştirilip Düğüm 4'te toplanır.

MedUNAS yapısı için ODE tarafından üretilen çözüm vektöründen hücre yapısının oluşturulması amacıyla eşleme aşaması gerekmektedir. Bu çalışmada, Awad vd. (2020) [19] tarafından önerilen eşleme yöntemi kullanılmıştır. İlk adımda, sıfırlarla dolu 17 elemandan oluşan bir çözüm vektörü oluşturulur. Bu vektördeki, ilk 15 eleman maksimum 2 iç-kenar kısıtlaması ve her düğüm için olası bağlantı sayısı dikkate alınarak rastgele üretilen gerçek değerlerle değiştirilir. Her düğüm için olası iç kenar sayısı farklı renkli kesikli çizgilerle gösterilmiştir (Şekil 3). Örneğin, kırmızı kesikli çizgi Düğüm 1 için olası iç-kenarları gösterirken, yeşil kesikli çizgi Düğüm 3 için 3 olası iç-kenarı göstermektedir. Düğüm 1 için yalnızca bir olası iç-kenar vardır. Bu sebeple, $[0, 1]$ aralığında bir tane gerçek değerli rastgele sayı üretilir (Şekil 3'te bu değer $0,02$ 'dir). Bu iç-kenar için on beş olası işlem olduğundan, 0 ile 1 arasındaki değerler, her biri belirli bir işleme karşılık gelen on beş eşdeğer bölmeyle ayrılır. İlk bölme $[0, 0,067)$; " $a : 1 \times 1$ 2B Evrişim"

işlemini gösterirken, ikinci bölme $[0,067, 0,133]$; {"b : 3×3 2B Evrişim"} işlemine karşılık gelir ve bu şekilde devam eder. Şekil 3'te gösterilen $0,02$ değeri ilk bölmedeki işleme karşılık geldiğinden a işlemi ilk eleman için seçilmiştir. Benzer şekilde, aynı şekil üzerindeki çözüm vektöründe gösterilen 4 ile 6 arasındaki elemanlar Düğüm 3 için olası iç-kenarları gösterir. Düğüm 3'ün üç olası iç-kenarı vardır ve bunlardan ikisi rastgele seçilmelidir (iki iç kenar kısıtı). Bu nedenle, $[0, 1]$ aralığında iki sayı rastgele üretilir (Şekil 3'te $0,81$ ve $0,91$) ve bu iki iç-kenar için sırasıyla m ve n işlemleri seçilir. Son iki eleman ise; n_c ve n_f^{init} karar değişkenlerine karşılık gelir. n_c için dört değer ve n_f^{init} için üç değer vardır, bu nedenle $[0, 1]$ aralığı sırasıyla eşit büyüklükte dört ve üç parçaya bölünür. Ardından, Şekil 3'te gösterildiği gibi (sarı vurgulu) bu karar değişkenleri için karşılık gelen değerler seçilir. Hücre yapısının ardından Şekil 2'de gösterilen MedUNAS mimarisi oluşturulur.

2.3. Karşıtlık Temelli Diferansiyel Gelişim (Opposition-based Differential Evolution - ODE)

Rahnamayan vd. (2008) [20] tarafından geliştirilen ODE yöntemi, popülasyon başlatma ve oluşturma aşamalarında karşıtlığa dayalı öğrenme (Karşıtlık Temelli Öğrenme - Opposition Based Learning - OBL) yöntemi kullanılmaktadır. Yapılan çalışmada, ODE için $DE/rand/1/bin$ kullanılmıştır. Dört adımdan oluşan bir yöntem olan karşıtlık temelli başlatma sırasıyla şu adımları gerçekleştirilir: Birinci adımda, rastgele bir şekilde P başlangıç popülasyonu oluşturulur. İkinci adımda ise her çözümün karşıtını hesaplamak için başlangıç popülasyonunun karşıtı olan OP oluşturulur. Verilen bir çözüm için karşıt çözümün üretilmesi denklem 1'e göre gerçekleştirilir. Denklem 1'de, $P_{i,G}^k$ ve $OP_{i,G}^k$, G jenerasyonu için mevcut popülasyon ve karşıt popülasyondaki i . çözümün k . karar değişkenini; l^k ve u^k , k . karar değişkeninin alt ve üst sınırlarını; D karar değişkenlerinin sayısını ve NP popülasyon büyüklüğünü temsil etmektedir.

$$OP_{i,G}^k = l^k + u^k - P_{i,G}^k \quad (1)$$

$for i: \{1, \dots, NP\}; k: \{1, \dots, D\}$

Bu problem için, üretilen her birey $[0, 1]$ aralığında sürekli değerlerden oluşur. Bu nedenle l^k ve u^k sırasıyla 0 ve 1 değerlerini alır. Sonuç olarak karşıt çözümler tüm karar değişkenlerinden bir değerinin çıkarılmasıyla oluşturulur. Üçüncü adımda P ve OP birleştirilir ve dördüncü adımda birleştirilen kümeden en iyi NP adet çözüm seçilir. Ayrıca ODE yönteminde belirli adımlarda, JR olasılığına bağlı olarak nesil atlama adımı gerçekleştirilir. Bu nesil atlama adımında, P 'nin karşıtı olan OP hesaplanır. Ancak burada, k . karar değişkeni için sabit değerler kullanmak yerine P popülasyonunda üretilmiş olan çözümlerin ilgili k karar değişkeni için almış olduğu maksimum ve minimum değerler kullanılır. Nesil atlama işlemi denklem 2'de gösterilmiştir.

$$OP_{i,G}^k = Min_G^k + Max_G^k - P_{i,G}^k \quad (2)$$

$for i: \{1, \dots, NP\}; k: \{1, \dots, D\}$

Denklem 2'de, Min_G^k ve Max_G^k sırasıyla k . karar değişkeninin G jenerasyonu P popülasyonunda aldığı minimum ve maksimum değerleri belirtir. Karşıt popülasyon hesaplandıktan sonra karşıtlık temelli başlatma işleminde olduğu gibi P ve OP birleştirilir ve NP adet en iyi çözüm seçilir. ODE algoritmasının sözde kodu Algoritma 1'de gösterilmiştir. Yapılan çalışmada, ODE için seçilen parametreler şu şekildedir: NP : 20, J_r : 0.3. Bu parametreler belirlenirken NP için Awad vd. (2020)[19] tarafından gerçekleştirilen çalışma referans alınırken, J_r için Rahnamayan vd. (2008)[20]'nin yayınladığı çalışmada önerilen değer kullanılmıştır. Üretilen çözümlerin uygunluk değerlerinin belirlenmesi için de "F1 skoru" metriği kullanılmıştır.

Algoritma 1 şu şekilde özetlenebilir: (Satır 2); $[0 - 1]$ aralığındaki gerçek değerlerden oluşan NP adet çözüm vektörü ile başlangıç popülasyonu oluşturulur. (Satır 4-6); Başlangıç popülasyonundaki her bir çözüm için karşıt olan çözümler hesaplanır. (Satır 7); Başlangıç popülasyonu ve karşıt popülasyondaki tüm çözümlerin uygunluk değerleri hesaplanır ve en iyi NP adet çözüm başlangıç popülasyonu için seçilir. (Satır 12-

Algoritma 1: Karşıtlık Temelli Diferansiyel Gelişim Algoritması Sözde Kodu

```

1  $G \leftarrow 0$ 
2  $P_G \leftarrow$  Başlangıç popülasyonunu oluştur
3 /* Karşıtlık Temelli Popülasyon Başlatma */
4 for  $i = 0; i < NP; i++$  do
5   for  $k = 0; k < D; k++$  do
6      $OP_{i,G}^k = l^k + u^k - P_{i,G}^k$ 
7  $P_G \leftarrow \{P_G, OP_G\}$  kümesinden  $NP$  adet en iyi çözümü seç
8  $G \leftarrow G + 1$ 
9 /* Karşıtlık Temelli Popülasyon Başlatma */
10 while Sonlandırma kriteri gerçekleşene kadar devam et do
11   for  $P_{G-1}$  içerisindeki tüm modeller, model do
12     hedef = model.vektor
13     mutant = Mutasyon(hedef)
14     mutant = Çaprazlama(hedef, mutant, CR)
15     mutant.uygunluk  $\leftarrow$  Değerlendir(mutant)
16     if mutant.uygunluk < hedef.uygunluk then
17        $P_G \leftarrow$  mutant // Bir sonraki jenerasyona
18       bu çözümü ekle
19     else
20        $P_G \leftarrow$  hedef // Bir sonraki jenerasyona
21       bu çözümü ekle
22 /* Karşıtlık Temelli Jenerasyon Atlaması */
23 if  $rand(0, 1) < J_r$  then
24   for  $i = 0; i < NP; i++$  do
25     for  $k = 0; k < D; k++$  do
26        $OP_{i,G}^k = Min_G^k + Max_G^k - P_{i,G}^k$ 
27    $P_G \leftarrow \{P_G, OP_G\}$  kümelerinden  $NP$  adet en iyi
28   çözümü seç
29 /* Karşıtlık Temelli Jenerasyon Atlaması */
30  $G \leftarrow G + 1$ 

```

15); Başlangıç popülasyonundaki her bir hedef çözüm için $DE/rand/1/bin$ operatörü ile mutasyon ve çaprazlama işlemleri uygulanarak mutant çözümler üretilir. Bu çözümlerin uygunluk değerleri hesaplanır. (Satır 16-19); Hedef ve mutant çözümler uygunluk değerlerine göre karşılaştırılır ve uygunluk değeri daha iyi olan çözüm bir sonraki popülasyona aktarılır. (Satır 21-25); Belirli bir olasılıkla "Karşıtlık Temelli Jenerasyon Atlaması" işlemi gerçekleştirilir ve ilgili popülasyondaki her bir çözüm için karşıt çözümler üretilir, uygunluk değerleri hesaplanır. İlgili çözümler ve karşıtları arasından en iyi NP adet çözüm seçilerek bir sonraki jenerasyona aktarılır. ODE adımları 500

adet çözümün üretilip, ekran kartı üzerinde eğitilmesi sonucunda sonlandırılır.

III. DENEYSSEL ÇALIŞMALAR

3.1. Veri Kümesi

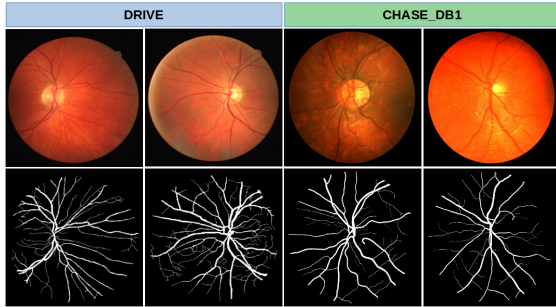
Yapılan çalışmada üretilen modellerin test edilmesi için herkese açık olarak paylaşılan iki farklı retinal kan damar segmentasyonu veri kümesi kullanılmıştır: DRIVE [21] ve CHASE_DB1 [22]. Her iki veri kümesine ait örnek görüntüler ve bunlara karşılık gelen gerçek referans görüntüleri Şekil 4'te gösterilmiştir.

3.1.1. DRIVE

565 × 584 boyutunda renkli görüntüler içeren retinal kan damar segmentasyonu veri kümesidir. Toplamda 40 adet retinal kan damar görüntüsü içeren bu veri kümesindeki görüntülerin 20 tanesi eğitim ve 20 tanesi test için kullanılmıştır. Her bir retina görüntüsüne ait gerçek referans görüntüsünün oluşturulması için iki farklı gözlemci kullanılmış ve yapılan çalışmada birinci gözlemci tarafından oluşturulan gerçek referans görüntüleri kullanılmıştır [6, 23].

3.1.2. CHASE_DB1

28 adet renkli retina görüntüsünden oluşan bir veri kümesidir ve Carballal vd. (2018) [22] tarafından oluşturulmuştur. Bu görüntüler, 999 × 960 piksel çözünürlüğe sahiptir ve iki bağımsız uzman tarafından elle işaretlenmiştir. Yapılan çalışmada, ilk uzmanın işaretlemeleri gerçek referans görüntüsü olarak kabul edilmiştir. Eğitim için ilk 20 görüntü ve son sekiz görüntü ise test için kullanılmıştır [6, 23].



Şekil 4. DRIVE ve CHASE_DB1 veri kümeleri için örnek görüntüler. Üst satır; gerçek görüntüleri, alt satır; gerçek görüntülere karşılık gelen gerçek referans görüntülerini temsil etmektedir.

3.2. Geliştirme Detayları

Derin öğrenme ağlarının gerçek performansı, ağı tüm eğitim veri kümesi üzerinde uzun eğitim dönemleri boyunca eğitilmesi ve eğitime dahil edilmeyen test kümesi üzerinde test edilmesi sonucunda hesaplanır. Ancak SMA çalışmaları sırasında birçok aday sinir ağı üretilmekte ve bu sinir ağlarının hepsinin uzun eğitim dönemleri boyunca tüm eğitim veri kümesi ile eğitilmesi uygulanabilir değildir. Bu sorunu çözebilmek için literatürde, üretilen ağların seyretilmiş bir eğitim kümesi üzerinde daha az eğitim adımı ile

değerlendirilmesi fikri önerilmiştir [24]. Yapılan çalışmada eğitim ve değerlendirme adımları iki kısma ayrılmıştır: SMA dönemi ve SMA sonrası dönem.

SMA döneminde meta-sezgisel ile SMA işlemi sırasında üretilen ağların eğitilmesi ve değerlendirilmesi gerçekleştirilir. Yapılan çalışmada, SMA dönemi sırasında üretilen ağların eğitilmesi için tüm eğitim veri kümesinin yarısı kullanılır. Ağın kalitesinin değerlendirilmesi için ise eğitim veri kümesinin %10'u kullanılarak oluşturulan doğrulama kümesi kullanılır. Üretilen ağlar, maksimum 120 eğitim adımı boyunca eğitilir ve eğitim sırasında ağı en iyi doğrulama kaybı 10 adım boyunca iyileşmezse bu adımda eğitim sonlandırılır (Erken Durdurma). 500 adet çözümün ekran kartı üzerinde eğitilmesi sonucunda ise meta-sezgiselin çalıştırılması sonlandırılır SMA dönemi bitilir.

Meta-sezgisel ile yapılan SMA işlemi sonucunda (SMA sonrası dönem) en iyi kaliteye sahip 5 adet çözüm seçilir ve bu çözümler eğitim veri kümesinin tamamıyla 200 eğitim adımı boyunca eğitilir. Ardından bu çözümlerin gerçek performansı, eğitime hiç dahil edilmemiş olan test kümesi üzerinde değerlendirilir ve bu test kümesi üzerinde elde edilen sonuçlar raporlanır. Yapılan çalışmaları boyunca üretilen derin öğrenme ağlarının oluşturulması için Pytorch kütüphanesi kullanılmıştır. SMA sonrası dönem için eğitilen UNAS-Net ve MedUNAS ağları için ortak olarak seçilen hiper-parametreler şu şekildedir: Optimize edici: Adam, Kayıp fonksiyonu: Dice kaybı + Jaccard kaybı + ikili çapraz entropi kaybı [25], Öğrenme oranı: 1e-3 ve Yığın boyutu: 128.

3.3. Ön-işleme

Yapılan çalışmada kullanılan DRIVE ve CHASE_DB1 veri kümeleri az eğitim örneğine sahiptir. Bu durum üretilen modellerin eğitimi ve değerlendirilmesini zorlaştırmakta, veri kümesindeki kompleks yapıların ortaya çıkarılmasını mümkün kılmamaktadır. Bu nedenle, gerçekleştirilen çalışmada Wentao vd. (2022) [6] tarafından önerilen ön-işleme adımları uygulanmıştır. Seçilen veri kümelerindeki eğitim örneklerinin sayısını arttırmak amacıyla görüntüler 48 × 48 boyutunda yamalara bölünmüş ve aşırı-öğrenmeyi engellemek amacıyla şu veri çoğaltma işlemleri uygulanmıştır: Dikey ve yatay rastgele çevirme ve [90, 180, 270] derece görüntülerin döndürülmesi. Anlatılan ön-işleme adımları sadece eğitim örnekleri üzerinde uygulanmıştır. Eğitilen modelin gerçek performansının hesaplandığı test veri kümelerinde ise herhangi bir ön-işleme veya veri çoğaltma işlemi uygulanmamıştır. Adil bir değerlendirme gerçekleştirilmesi için test setindeki orijinal görüntüler kullanılmıştır.

3.4. Performans Ölçütleri

Önerilen yöntemlerin segmentasyon performansı doğruluk (ACC), duyarlılık (SEN), özgüllük (SPE), F1 puanı (F1) ve birleşimlerin kesişimi (IOU) kullanılarak değerlendirilmiştir [6]. Bu metrikler dört farklı terminoloji kullanılarak hesaplanmaktadır: Doğru Pozitif (TP): doğru şekilde damar olarak sınıflandırılan pikseller (pozitif örnekler); Doğru Negatif (TN): doğru şekilde damar yok olarak sınıflandırılan pikseller (negatif örnekler); Yanlış Pozitif (FP): yanlışlıkla damar olarak sınıflandırılan ancak damar olmayan pikselleri ifade eder; Yanlış Negatif (FN): yanlışlıkla damar olmayan olarak sınıflandırılan ancak gerçekte damar olan pikselleri ifade eder. Bu ölçütlerin her biri önerilen yöntemin segmentasyon performansını farklı şekillerde ölçmektedir ve ölçütlerin hesaplanması için kullanılan formüller Tablo 2'de gösterilmiştir. Doğruluğu hesaplamak için doğru şekilde damar/damar olmayan olarak sınıflandırılan piksel sayısı görüntüdeki toplam piksel sayısına bölünürken, özgüllüğü hesaplamak için doğru şekilde damar olmayan olarak sınıflandırılan piksel sayısı görüntüdeki toplam damar olmayan piksel sayısına bölünmektedir. Bu çalışmada, segmente edilmiş görüntüler oluşturmak için eşik değeri 0,5 olarak belirlenmiştir.

Tablo 2. Performans Ölçütleri

Performans Ölçütü	Formül
ACC	$(TP + TN) / (TP + TN + FP + FN)$
SEN	$(TP) / (TP + FN)$
SPE	$(TN) / (TN + FP)$
F1	$(2TP) / (2TP + FP + FN)$
IOU	$(TP) / (TP + FP + FN)$

IV. BULGULAR VE TARTIŞMA

UNAS-Net ve MedUNAS yöntemleri için elde edilen sonuçlar farklı SMA çalışmaları ile DRIVE ve CHASE_DB1 veri kümeleri üzerinde segmentasyon performansı (Doğruluk-ACC, Duyarlılık-SEN, Özgüllük-SEP, F1 skoru-F1, kesişim/birleşim-IOU) ve model karmaşıklığı açısından karşılaştırılmıştır. Elde edilen sonuçlar Tablo 3'te verilmiştir. Ayrıca bu tabloda son zamanlarda literatürde önerilen SMA çalışmalarının (HNAS [13], Genetic U-Net [14],

BTU-Net [15] ve Popat vd. (2020) [26]) sonuçları da listelenmiştir. Adil bir kıyaslama yapılması amacıyla tabloda verilen çalışmalar tarafından raporlanan sonuçlar kullanılmıştır. Tablo 3'te verilen sonuçlar incelendiğinde UNAS-Net ve MedUNAS yöntemlerinin her iki veri kümesi için de en az parametreye (Genetic U-Net hariç) sahip ağları ürettiği görülmektedir. DRIVE veri kümesi için MedUNAS en iyi SEN değerini elde ederken ACC ve F1 değerleri açısından en iyi ikinci değerleri elde etmiştir. MedUNAS, Popat vd. (2020)'den 6,37, HNAS çalışmasından 2,28 ve BTU-Net çalışmasından 13,22 kat daha az parametreye sahip model ile ACC, SEN ve F1 değerleri açısından daha iyi performans göstermiştir. Benzer şekilde, UNAS-Net'de bu çalışmalardan sırasıyla 26,12, 9,35 ve 54,19 kat daha az parametreyle yine ACC, SEN ve F1 değerleri açısından daha iyi segmentasyon performansı göstermiştir. MedUNAS ve UNAS-Net çalışmaları kendi aralarında karşılaştırıldığında ise beklenildiği gibi UNAS-Net'in daha az parametreye sahip ağlar ürettiği görülmüş (MedUNAS iki iç-kenar kısıtı sebebiyle daha yüksek parametrelili ağlar üretmeye daha yatkın) ve yine de MedUNAS ile tüm ölçütler açısından yüksek rekabetçi sonuçlar elde ettiği gösterilmiştir. CHASE_DB1 veri kümesi için de benzer sonuçların elde edildiği görülmektedir. UNAS-Net, MedUNAS'a göre 1,76 kat daha az parametreye sahip sinir ağı ile tüm ölçütler açısından rekabetçi sonuçlar elde etmiştir (Maksimum 0,26 puan fark).

UNAS-Net ve MedUNAS yöntemleri U-şekilli ağlar öneren son-teknoloji çalışmalar ile de karşılaştırılmıştır ve sonuçlar Tablo 4'te verilmiştir. Tablo 4'te verilen sonuçlar doğru karşılaştırma yapılması için Wentao vd. (2022)[6] tarafından yapılan çalışmadan alınmıştır (MedUNAS ve UNAS-Net hariç). Elde edilen sonuçlar incelendiğinde UNAS-Net ve MedUNAS yöntemlerinin DRIVE veri kümesi için en az parametreye sahip modelleri ürettiği görülmektedir. MedUNAS yönteminin ACC için en iyi ve F1 için en iyi ikinci sonucu elde ettiği görülmektedir ve diğer ölçütler için de 12,22 kata kadar daha az parametre ile yüksek rekabetçi sonuçlar elde edilmiştir. UNAS-Net yöntemi MedUNAS yönteminden 4,09 kat daha az parametreye sahip model üretmiş ancak performans ölçütleri açısından daha kötü sonuçlar vermiştir.

Tablo 3. SMA yöntemlerinin segmentasyon performansı ve karmaşıklık açısından karşılaştırılması. Kırmızı ve mavi renkler sırasıyla en iyi iki sonucu göstermektedir.

Yöntemler	Param (M)	DRIVE				
		ACC	SEN	SPE	F1	IOU
HNAS [13]	2,90	95,46	77,44	98,09	81,29	-
Genetic U-Net [14]	0,27	97,07	83,00	98,43	83,14	-
BTU-Net [15]	16,80	96,89	-	98,44	81,78	-
Popat vd. [26]	8,10	95,34	75,01	98,31	-	-
UNAS-Net [16]	0,31	97,02	82,00	98,42	81,80	69,23
MedUNAS [17]	1,27	97,06	83,41	98,36	82,18	69,77

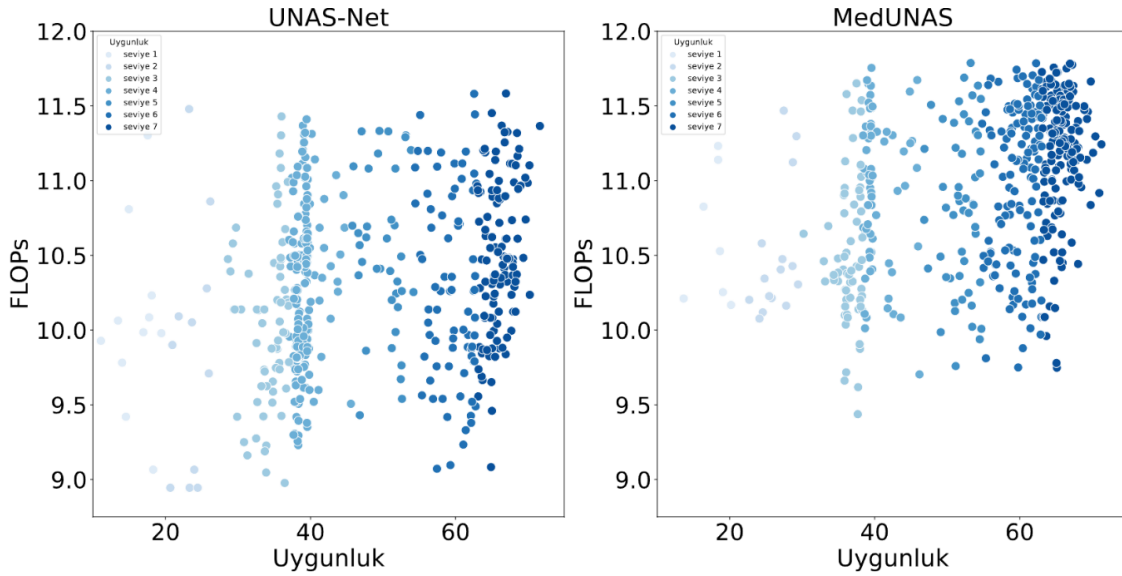
Yöntemler	Param (M)	CHASE_DB1				
		ACC	SEN	SPE	F1	IOU
Genetic U-Net [14]	0,27	97,69	84,63	98,57	82,23	-
BTU-Net [15]	16,80	97,87	-	98,64	81,09	-
UNAS-Net [16]	3,41	97,55	87,50	98,34	80,52	67,43
MedUNAS [17]	6,01	97,56	84,50	98,60	80,22	66,51

Tablo 4. Önerilen yöntemlerin son-teknoloji yöntemler ile segmentasyon performansı ve parametre sayısı açısından karşılaştırılması. Kırmızı ve mavi renkler sırasıyla en iyi iki sonucu göstermektedir.

Yöntemler	Yıl	Param (M)	DRIVE					CHASE_DB1				
			ACC	SEN	SPE	F1	IOU	ACC	SEN	SPE	F1	IOU
U-Net [3]	2015	7,76	96,78	80,57	98,33	81,41	68,64	97,43	76,50	98,84	78,98	65,26
U-Net++[27]	2018	9,05	96,79	78,91	98,50	81,14	68,27	97,39	83,57	98,32	80,15	66,88
Attention U-Net [28]	2018	8,73	96,62	79,06	98,31	80,39	67,21	97,30	83,84	98,20	79,64	66,17
CS-Net [29]	2019	8,40	96,32	81,70	98,54	80,39	70,17	97,42	84,00	98,32	80,42	67,25
AG-Net [30]	2019	-	96,92	81,00	98,48	-	69,65	97,43	81,86	98,48	-	66,69
RVSegNet [31]	2020	5,20	96,81	81,07	98,45	-	-	97,26	80,69	98,36	-	-
SCS-Net [32]	2021	-	96,97	82,89	98,38	-	-	97,44	83,65	98,39	-	-
SGL [5]	2021	15,53	97,05	83,80	98,34	83,16	-	97,71	86,90	98,43	82,71	-
FR-UNet [6]	2022	5,72	97,05	83,56	98,37	83,16	71,20	97,48	87,98	98,14	81,51	68,82
UNAS-Net[16]	2023	0,31-3,41	97,02	82,00	98,42	81,80	69,23	97,55	87,50	98,34	80,52	67,43
MedUNAS[17]	2023	1,27-6,01	97,06	83,41	98,36	82,18	69,77	97,56	84,50	98,60	80,22	66,51

CHASE_DB1 için elde edilen sonuçlara bakıldığında UNAS-Net yönteminin en az karmaşıklığa sahip sinir ağını ürettiği görülmektedir. MedUNAS, en az karmaşıklığa sahip üçüncü ağ ile ACC ve SPE ölçütleri açısından en iyi ikinci sonuçları elde etmiştir. UNAS-Net ise en az karmaşıklığa sahip ağ yapısı ile SEN ve IOU ölçütleri açısından en iyi ikinci sonuçları elde etmiştir. UNAS-Net ve MedUNAS yöntemleri karşılaştırıldığında ise UNAS-Net'in 1,76 kat daha parametre ile SEN, F1 ve IOU ölçütleri açısından MedUNAS'tan daha iyi sonuçlar verdiği görülmektedir. Diğer ölçütler açısından incelendiğinde her iki yöntemin de diğer çalışmalar ile rekabetçi sonuçlar elde ettiği görülmektedir. UNAS-Net ve MedUNAS yöntemleri U-şekilli ağlar temel alınarak

üretildiği için U-Net ile de karşılaştırılmıştır. Elde edilen sonuçlara bakıldığında her iki veri kümesi için de UNAS-Net ve MedUNAS'ın daha az parametre sayısı ile tüm ölçütler açısından U-Net'ten daha iyi performans gösterdiği görülmektedir. DRIVE veri kümesi için UNAS-Net 25 kat, MedUNAS ise 6,11 kat daha az parametre sayısı ile U-Net'ten tüm performans ölçütleri açısından daha iyi sonuçlar verirken, CHASE_DB1 için UNAS-Net 2,27, MedUNAS ise 1,29 kat daha az parametre ile tüm ölçütler açısından daha iyi sonuçlar vermiştir. Sonuç olarak UNAS-Net ve MedUNAS'ın hem segmentasyon performansı hem de model karmaşıklığı açısından temel alınan U-Net'i iyileştirdiği görülmüştür.



Şekil 5. UNAS-Net ve MedUNAS yöntemlerinin arama performansının model karmaşıklığı ve segmentasyon performansı açısından karşılaştırılması. Seviyenin yükselmesi üretilen çözümlerin kalitesinin arttığını göstermektedir.

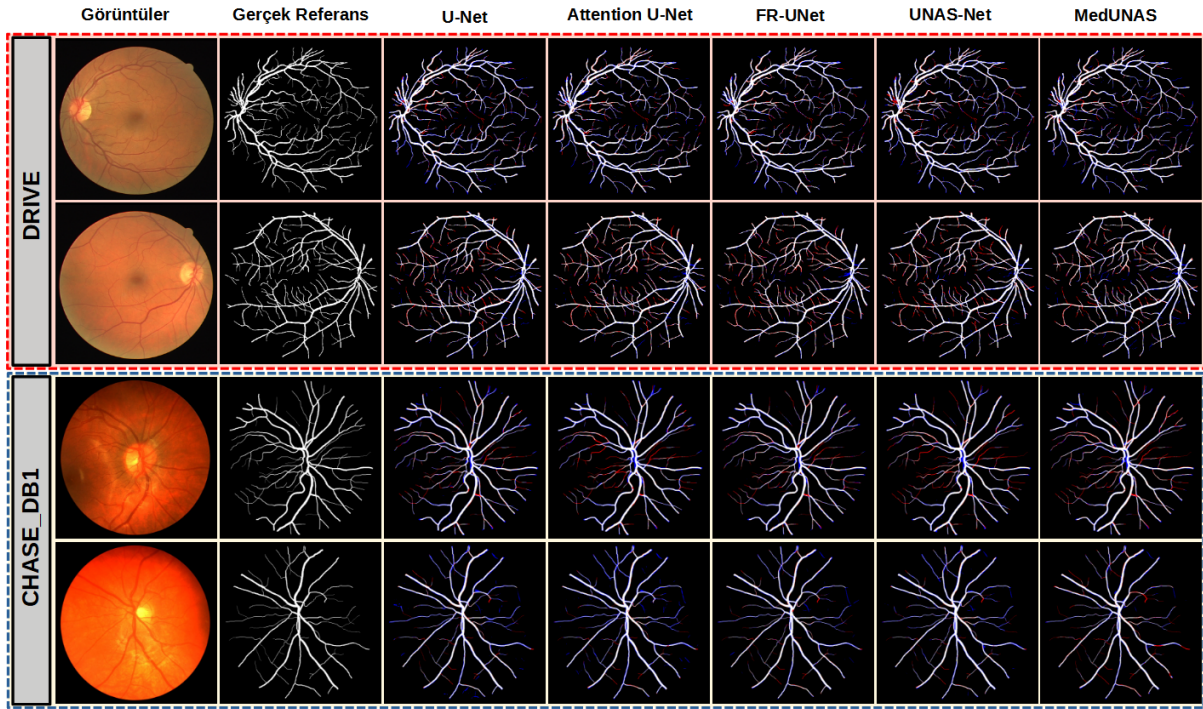
Önerilen yöntemler tarafından elde edilen sonuçlar görsel olarak da karşılaştırılmıştır. Elde edilen sonuçlar Şekil 6'da verilmiştir. U-Net ve Attention U-Net modelleri kendi çalışmaları referans alınarak tekrar kodlanabildiği için ve FR-UNet çalışması da farklı veri kümeleri için eğitilmiş ağırlıkları hazır olarak sunduğu için karşılaştırmaya dahil edilmiştir. Şekil 6'da gösterilen yöntemler FP, FN ve TP açısından karşılaştırılmıştır. Bölüm 3.4'te belirtildiği gibi Doğru Pozitif (TP): doğru şekilde damar olarak sınıflandırılan pikseller (pozitif örnekler); Yanlış Pozitif (FP): yanlışlıkla damar olarak sınıflandırılan ancak damar olmayan pikselleri ifade eder; Yanlış Negatif (FN): yanlışlıkla damar olmayan olarak sınıflandırılan ancak gerçekte damar olan pikselleri ifade eder. Bu nedenle elde edilen görüntülerde, FP ve FN piksellerin sayısının mümkün olduğunca az olması beklenir. Elde edilen sonuçlara bakıldığında, DRIVE veri kümesi için tüm yöntemlerin birbirine benzer sonuçlar elde ettiği ancak FR-UNet, UNAS-Net ve MedUNAS yöntemlerinin daha az FP ve FN piksel içerdiği görülmektedir. Benzer sonuçlar CHASE_DB1 veri kümesi için de geçerlidir. FR-UNet ve önerilen yöntemler kıyaslandığında ise önerilen yöntemler daha fazla FN piksele sahipken, FR-UNet'in daha fazla FP piksele sahip olduğu görülmektedir.

UNAS-Net ve MedUNAS için arama yöntemi olarak kullanılan ODE'nin arama uzayındaki performansı iki farklı kodlama şeması üzerinden karşılaştırılmış ve Şekil 5'te gösterilmiştir. Her iki yöntem de, üretilen çözümlerin uygunluk değerleri ve model karmaşıklığı (Kayan Noktalı Operasyonlar - Floating Point Operations - FLOPs) açısından karşılaştırılmıştır. Yapılan çalışmalarda, üretilen modellerin karmaşıklığının karşılaştırılabilmesi için parametre

sayısı ve FLOPs gibi değerler sıklıkla kullanılmaktadır [20]. Şekil 5'te y ekseninde gösterilen değerler \log_{10} tabanında temsil edilmiş ve model karmaşıklığının bir de FLOPs açısından karşılaştırılması hedeflenmiştir. Elde edilen sonuçlara bakıldığında, UNAS-Net yönteminin yüksek uygunluk değerine (F1 skoru) sahip çözümleri daha geniş bir FLOPs aralığında (y ekseninde 9-11,5 değerleri arası) yakalayabildiği görülmektedir. Bu durum, UNAS-Net'in yüksek kaliteli çözümleri daha az karmaşıklığa sahip modeller ile de yakalayabildiğini göstermektedir. MedUNAS yönteminde ise iyi çözümlerin genel olarak daha yüksek FLOPs değerlerine sahip alanlarda (y ekseninde 11-12 değerleri arası) yakalanabildiği görülmektedir.

V. SONUÇLAR

Yapılan çalışmada, SMA için önerdiğimiz UNAS-Net ve MedUNAS yöntemlerinde kullanılan kodlama şemaları iki farklı retinal kan damar segmentasyonu veri kümesi üzerinde karşılaştırılmıştır. Bu yöntemler, farklı SMA çalışmaları ve U-şekilli ağlar öneren son teknoloji çalışmalar ile parametre sayısı ve segmentasyon performansı açısından karşılaştırılmıştır. Elde edilen sonuçlara bakıldığında, UNAS-Net ve MedUNAS yöntemlerinin tüm veri kümeleri için en az parametreye (Genetic U-Net hariç) sahip ağları ürettiği görülmektedir. MedUNAS, BTU-Net'ten 13,22 kat daha az parametre ile ACC, SEN ve FI değerleri açısından daha iyi performans gösterirken, UNAS-Net 54,19 kat daha az parametreyle ACC, SEN ve FI değerleri açısından daha iyi segmentasyon performansı göstermiştir. Ayrıca, bu yöntemler U-şekilli son teknoloji çalışmalar ile de



Şekil 6. Farklı segmentasyon modelleri tarafından elde edilen görüntülerin görsel olarak karşılaştırılması.

Kırmızı renk yanlış segmente edilen negatif pikselleri (FN), mavi renk yanlış segmente edilen pozitif pikselleri (FP) ve beyaz renk ise doğru segmente edilen pikselleri (TP) temsil etmektedir.

karşılaştırılmış ve DRIVE veri kümesi için en az parametreye sahip modeller ile yüksek rekabetçi sonuçlar elde edilmiştir. CHASE_DB1 için UNAS-Net 1,76 kat daha az parametre ile *SEN*, *F1* ve *IOU* ölçütleri açısından MedUNAS'tan daha iyi sonuçlar vermiş diğer ölçütler açısından da her iki yöntemin de diğer çalışmalar ile rekabetçi sonuçlar elde ettiği görülmüştür. UNAS-Net ve MedUNAS yöntemleri temel alınan U-Net yöntemi ile de karşılaştırılmıştır. DRIVE veri kümesi için UNAS-Net 25, MedUNAS ise 6,11 kat daha az parametre sayısı ile temel alınan U-Net'ten tüm performans ölçütleri açısından daha iyi sonuçlar verirken, CHASE_DB1 için UNAS-Net 2,27, MedUNAS ise 1,29 kat daha az parametre ile tüm ölçütler açısından daha iyi sonuçlar vermiştir. Sonuç olarak UNAS-Net ve MedUNAS'ın hem segmentasyon performansı hem de model karmaşıklığı açısından temel alınan U-Net'i iyileştirdiği görülmüştür. Ayrıca, UNAS-Net için seçilen kodlama şeması ile MedUNAS'a kıyasla daha az karmaşık modeller ile yüksek rekabetçi sonuçlar elde edilebildiği gösterilmiştir. Önerilen yöntemler kalın damarların segmentasyonu için başarılı sonuçlar verse de ince damarlar için aynı başarıyı elde edememektedir. Gelecek çalışmalarda, gerçek referans görüntüsündeki ince ve kalın damarların ayrıştırılması ve bu görüntüler ile iki farklı derin sinir ağı mimarisinin ayrı ayrı eğitilmesi amaçlanmaktadır. Ayrıca, Karşıtlık Temelli Diferansiyel Gelişim Algoritmasının başarısı seçilen parametreler ile doğrudan ilişkilidir. Bu nedenle bu algoritma üzerinde parametre optimizasyonu gerçekleştirilebilir. Benzer şekilde, öğrenme oranı, kayıp fonksiyonu ve yığın

boyutu gibi parametreler için de parametre ayarlama işlemi gerçekleştirilebilir. Önerilen yöntemler, 3B damar segmentasyonu için test edilebilir ve farklı medikal görüntü segmentasyonu problemleri üzerinde değerlendirilebilir. Son olarak, segmente edilen görüntü üzerinde genişletme ve daraltma gibi işlemler uygulanarak son-işlemenin sonuca olan etkisi incelenebilir.

KAYNAKLAR

- [1] Abràmoff, M. D., Garvin, M. K., & Sonka, M. (2010). Retinal imaging and image analysis. *IEEE reviews in biomedical engineering*, 3, 169-208.
- [2] Wang, B., Qiu, S., He, H. (2019). Dual Encoding U-Net for Retinal Vessel Segmentation. In: Shen, D., et al. *Medical Image Computing and Computer Assisted Intervention – MICCAI 2019*. MICCAI 2019. Lecture Notes in Computer Science(), vol 11764. Springer, Cham.
- [3] Ronneberger, O., Fischer, P., Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. In: Navab, N., Hornegger, J., Wells, W., Frangi, A. (eds) *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*. MICCAI 2015. Lecture Notes in Computer Science(), vol 9351. Springer, Cham.
- [4] Chen, C., Chuah, J. H., Ali, R., & Wang, Y. (2021). Retinal vessel segmentation using deep learning: a review. *IEEE Access*, 9, 111985-112004.

- [5] Zhou, Y., Yu, H., & Shi, H. (2021). Study group learning: Improving retinal vessel segmentation trained with noisy labels. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2021: 24th International Conference, Strasbourg, France, September 27–October 1, 2021, Proceedings, Part I 24* (pp. 57-67). Springer International Publishing.
- [6] Liu, W., Yang, H., Tian, T., Cao, Z., Pan, X., Xu, W., ... & Gao, F. (2022). Full-resolution network and dual-threshold iteration for retinal vessel and coronary angiograph segmentation. *IEEE Journal of Biomedical and Health Informatics*, 26(9), 4623-4634.
- [7] Rong, Y., Xiong, Y., Li, C., Chen, Y., Wei, P., Wei, C., & Fan, Z. (2023). Segmentation of retinal vessels in fundus images based on U-Net with self-calibrated convolutions and spatial attention modules. *Medical & Biological Engineering & Computing*, 1-11.
- [8] Baker, B., Gupta, O., Naik, N., & Raskar, R. (2016). Designing neural network architectures using reinforcement learning. *arXiv preprint arXiv:1611.02167*.
- [9] Zoph, B., & Le, Q. V. (2016). Neural architecture search with reinforcement learning. *arXiv preprint arXiv:1611.01578*.
- [10] Grathwohl, W., Creager, E., Ghasemipour, S. K. S., & Zemel, R. (2018). Gradient-based optimization of neural network architecture.
- [11] Liu, H., Simonyan, K., & Yang, Y. (2018). Darts: Differentiable architecture search. *arXiv preprint arXiv:1806.09055*.
- [12] Liu, X., Song, L., Liu, S., & Zhang, Y. (2021). A review of deep-learning-based medical image segmentation methods. *Sustainability*, 13(3), 1224.
- [13] Houreh, Y., Mahdinejad, M., Naredo, E., Dias, D. M., & Ryan, C. (2021). HNAS: Hyper Neural Architecture Search for Image Segmentation. In *ICAART (2)* (pp. 246-256).
- [14] Wei, J., Zhu, G., Fan, Z., Liu, J., Rong, Y., Mo, J., ... & Chen, X. (2021). Genetic U-Net: automatically designed deep networks for retinal vessel segmentation using a genetic algorithm. *IEEE Transactions on Medical Imaging*, 41(2), 292-307.
- [15] Rajesh, C., Sadam, R., & Kumar, S. (2023). An evolutionary U-shaped network for Retinal Vessel Segmentation using Binary Teaching–Learning–Based Optimization. *Biomedical Signal Processing and Control*, 83, 104669.
- [16] Kuş, Z., Aydın, M., Kiraz, B., & Can, B. (2022). Neural Architecture Search Using Metaheuristics for Automated Cell Segmentation. In *Metaheuristics International Conference* (pp. 158-171). Cham: Springer International Publishing.
- [17] Kuş, Z., Kiraz, B. (2023). Evolutionary Architecture Optimization for Retinal Vessel Segmentation. *IEEE Journal of Biomedical and Health Informatics*.
- [18] Dong, J., Hou, B., Feng, L., Tang, H., Tan, K. C., & Ong, Y. S. (2022). A cell-based fast memetic algorithm for automated convolutional neural architecture design. *IEEE Transactions on Neural Networks and Learning Systems*.
- [19] Awad, N., Mallik, N., & Hutter, F. (2020). Differential evolution for neural architecture search. *arXiv preprint arXiv:2012.06400*.
- [20] Rahnamayan, S., Tizhoosh, H. R., & Salama, M. M. (2008). Opposition-based differential evolution. *IEEE Transactions on Evolutionary Computation*, 12(1), 64-79.
- [21] Staal, J., Abramoff, M. D., Niemeijer, M., Viergever, M. A., & Van Ginneken, B. (2004). Ridge-based vessel segmentation in color images of the retina. *IEEE transactions on medical imaging*, 23(4), 501-509.
- [22] Carballal, A., Novoa, F. J., Fernandez-Lozano, C., García-Guimaraes, M., Aldama-López, G., Calviño-Santos, R., ... & Pazos, A. (2018). Automatic multiscale vascular image segmentation algorithm for coronary angiography. *Biomedical Signal Processing and Control*, 46, 1-9.
- [23] Li, J., Gao, G., Liu, Y., & Yang, L. (2023). MAGF-Net: A multiscale attention-guided fusion network for retinal vessel segmentation. *Measurement*, 206, 112316.
- [24] Xie, X., Song, X., Lv, Z., Yen, G. G., Ding, W., & Sun, Y. (2023). Efficient evaluation methods for neural architecture search: A survey. *arXiv preprint arXiv:2301.05919*.
- [25] Kuş, Z., Kiraz, B., Göksu, T. K., Aydın, M., Özkan, E., Vural, A., Kiraz, A., & Can, B. (2023). Differential evolution-based neural architecture search for brain vessel segmentation. In *Engineering Science and Technology, an International Journal* (Vol. 46, p. 101502).
- [26] Papat, V., Mahdinejad, M., Cedeño, O. D., Naredo, E., & Ryan, C. (2020). GA-based U-Net Architecture Optimization Applied to Retina Blood Vessel Segmentation. In *IJCCI* (pp. 192-199).
- [27] Zhou, Z., Rahman Siddiquee, M. M., Tajbakhsh, N., & Liang, J. (2018). Unet++: A nested u-net architecture for medical image segmentation. In *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support: 4th International Workshop, DLMIA*

- 2018, and 8th International Workshop, ML-CDS 2018, Held in Conjunction with MICCAI 2018, Granada, Spain, September 20, 2018, Proceedings 4 (pp. 3-11). Springer International Publishing.
- [28] Oktay, O., Schlemper, J., Folgoc, L. L., Lee, M., Heinrich, M., Misawa, K., ... & Rueckert, D. (2018). Attention u-net: Learning where to look for the pancreas. arXiv 2018. arXiv preprint arXiv:1804.03999.
- [29] Mou, L., Zhao, Y., Chen, L., Cheng, J., Gu, Z., Hao, H., ... & Liu, J. (2019). CS-Net: Channel and spatial attention network for curvilinear structure segmentation. In Medical Image Computing and Computer Assisted Intervention–MICCAI 2019: 22nd International Conference, Shenzhen, China, October 13–17, 2019, Proceedings, Part I 22 (pp. 721-730). Springer International Publishing.
- [30] Zhang, S., Fu, H., Yan, Y., Zhang, Y., Wu, Q., Yang, M., ... & Xu, Y. (2019). Attention guided network for retinal image segmentation. In Medical Image Computing and Computer Assisted Intervention–MICCAI 2019: 22nd International Conference, Shenzhen, China, October 13–17, 2019, Proceedings, Part I 22 (pp. 797-805). Springer International Publishing.
- [31] Wang, W., Zhong, J., Wu, H., Wen, Z., & Qin, J. (2020). Rvseg-net: An efficient feature pyramid cascade network for retinal vessel segmentation. In Medical Image Computing and Computer Assisted Intervention–MICCAI 2020: 23rd International Conference, Lima, Peru, October 4–8, 2020, Proceedings, Part V 23 (pp. 796-805). Springer International Publishing.
- [32] Wu, H., Wang, W., Zhong, J., Lei, B., Wen, Z., & Qin, J. (2021). Scs-net: A scale and context sensitive network for retinal vessel segmentation. *Medical Image Analysis*, 70, 102025.