

HSA ile Sayısal Görüntülerde Kenar Çıkarımı için Parçacık Sürüsü Optimizasyon Algoritması Tabanlı Şablon Katsayısı Öğrenimi

Particle Swarm Optimization Algorithm Based Cloning Template Learning for Edge Detection in Digital Images with CNN

Enis Günay¹, Alper Baştürk², Hilal Benli³

¹Elektrik-Elektronik Müh. Böl., ²Bilgisayar Müh. Böl., Erciyes Üniversitesi, KAYSERİ
{egunay,ab}@erciyes.edu.tr

³Bilişim Müdürlüğü, TÜBİTAK, ANKARA
hilal.benli@tubitak.gov.tr

Özet

Bu çalışmada, bir hücreli sinir ağı (HSA) şablon katsayıları sayısal imgelede kenar çıkarımı amacıyla parçacık sürüsü optimizasyon algoritması kullanılarak belirlenmektedir. Sunulan HSA'nın şablon katsayıları basit yapay eğitim imgeleri kullanılarak belirlenmiştir. Sunulan kenar çıkarıcının başarımı birinci dereceden farklı kenar çıkarıcılarla ve literatürdeki diğer HSA tabanlı kenar çıkarıcılarla karşılaştırılmış, tüm operatörlerin başarımları hem ikili hem de gri seviyeli test imgeleri üzerinde test edilmiştir. Benzetim sonuçları sunulan kenar çıkarıcının başarımının diğer kenar çıkarıcılara kıyasla daha iyi olduğunu göstermiştir.

Abstract

A cellular neural network (CNN) based edge detector whose cloning template is tuned by particle swarm optimization (PSO) algorithm is presented. Cloning template of the proposed CNN is adaptively tuned by using simple training images. The performance of the proposed edge detector is evaluated on different test images and compared with well known first-order edge detectors and other CNN based edge detectors from the literature. Simulation results indicate that the proposed CNN edge detector outperforms competing edge detectors and offers superior performance in edge detection in digital images.

1. Giriş

Hücreli sinir ağı (HSA), yapay zeka, görüntü işleme ve enformasyon teorisi gibi araştırma ve uygulama alanlarında kendisine yer bulmuş işlemsel geniş ölçekli doğrusal olmayan devreleri temsil eder [1-8]. HSA, yerel bağlantılılık özelliği sayesinde imge işleme alanında çok sayıda uygulamada kullanılmıştır [9-14].

HSA'nın dinamik davranışı, şablon katsayıları (cloning template) olarak adlandırılan 19 parametre tarafından belirlenmektedir [14,15]. HSA tabanlı görüntü işleme çalışmaları içerisinde, kararlı şablon katsayılarının belirlenmesi işlemi en önemli problemlerden birisini oluşturmaktadır [15,16]. Şablon katsayılarının belirlenmesi üzerine literatürde yer alan çalışmalar, kendi içerisinde üç kısma ayrılmaktadır.

Şablon katsayılarının belirlenmesinde tercih edilen yollardan birincisi deneme yanılma yöntemidir ve tasarımcının bu konudaki kişisel tecrübesine dayanmaktadır. Bu tasarım yöntemi ile nadiren olumlu sonuçlara ulaşılsa da arzu edilen başarımlar çoğunlukla elde edilememektedir. Ayrıca başarılı sonuçlara ulaşabilmek adına şablon katsayıları ve ağ yapısı üzerinde oldukça fazla uygulama yapmayı gerektirmektedir. İkinci tercih edilen yol ise probleme en yakın literatürdeki HSA şablon katsayılarından yola çıkılarak kararlı katsayılarla ulaşılmaya çalışılmaktadır. Bu yöntemin dezavantajı olarak yaklaşık çözümlere dayanması gösterilebilir [17].

Üçüncü ve en çok tercih edilen yöntem ise şablon katsayılarının optimizasyonudur. Şablon katsayılarının optimizasyon yöntemleri de iki kısma ayrılabilir. Bunlardan birincisi yapay sinir ağı kullanılarak şablon katsayılarının optimizasyonu, diğeri ise bu optimizasyonun genetik algoritma, ısıtma işlem algoritması, parçacık sürüsü optimizasyonu, benzetimli tavlama gibi sezgisel yaklaşımlar ile gerçekleştirilmesidir [17]. Her iki metotta da arzu edilen çıkış görüntüsü ile giriş görüntüsü kullanılarak kararlı şablon katsayılarına ulaşılmaya çalışılmaktadır.

Bir sayısal imgedeki kenarlar imgede yer alan nesnelere birbirinden ayıran sınırlar oldukları için nesnelere hakkında vermiş oldukları bilgiler açısından önem arz ederler. Kenar çıkarımı, imge bölütleme, sınır sezimi, nesne tanıma ve sınıflandırma vb. gibi imge işleme operasyonları öncesinde yapılan ilk operasyonlardan birisi olması nedeniyle imge işleme uygulamalarında sıklıkla gerçekleştirilen bir operasyondur. Sonuç olarak, ardı sıra gelecek olan imge işleme uygulamalarının başarımı kenar çıkarma işleminin başarımına doğrudan bağlılık gösterir [18,19].

İmge yoğunluğu, kenarlarda ani değişimler gösterir. Bu nedenle, kenar çıkarımı herhangi bir imge elemanı yerleşimindeki imge yoğunluk fonksiyonunun türevinin hesaplanması gerektirir. İmge elemanı yerleşimindeki imge yoğunluk fonksiyonunun türevinin değeri yüksek olması, bu yerleşimdeki imge elemanının bir kenar elemanı olarak sınıflandırılabilirliği manasına gelir [18,19].

Literatürde sayısal imgelede kenar çıkarımına yönelik farklı birçok yöntem bulunmaktadır. Bu yöntemlere istatistiksel metotlar [20-24], fark metotları [25-28] ve eğri uydurma metotları [29-33] örnek verilebilir. Sobel [25], Prewitt [26] ve Kirsch [27] kenar çıkarıcıları gibi klasik yöntemler kenar yerleşimlerini belirlemek için birinci dereceden yönlü türevleri hesaplarlar. Sıfır geçişli kenar çıkarıcılar [34, 35] Laplacian

operatörü ile birlikte ikinci türevi kullanırlar. Bir gaussian kenar çıkarıcı olan Canny operatörü [36], literatürde oldukça yoğun kullanılan popüler bir kenar sezicidir [37-39].

Son zamanlarda, gelişimsel bir hesaplama tekniği olan parçacık sürüsü optimizasyon algoritması (PSOA) birçok farklı optimizasyon problemine uygulanmış ve başarılı sonuçlar elde edilmiştir. İlk olarak [40]'ta duyurulan PSO, kuş sürülerindeki ve balık topluluklarındaki sosyal dayanışmayı ve sürü teorisini temel alarak geliştirilmiş bir algoritmadır. PSOA çözüme rasgele bir çözüm popülasyonu ile başlar. Parçacık olarak isimlendirilen popülasyondaki her bir birey belirli bir hız ve konum değerine sahiptir ve bu değerler hem bireyin kendi hız ve konum bilgisinden hem de popülasyondaki diğer bireylerin hız ve konum bilgisinden yararlanılarak hesaplanır. Parçacıklar çok boyutlu uzayda konum değiştirirler ve global optimaya yakınsarlar. Basit mantığı, kolay gerçekleştirilmesi ve hızlı yakınsaması nedeniyle bu algoritma günümüzde yoğun bir şekilde, özellikle de sınırlamasız optimizasyon problemlerinde [41,42] sıklıkla kullanılmaktadır.

Bu çalışmada, sayısal imgelerdeki kenarların çıkarımı amacıyla PSOA kullanılarak bir HSA'nın şablon katsayılarının nasıl belirlenebileceği konu alınmaktadır. Bildirinin geri kalanı şu şekilde düzenlenmiştir: 2. bölümde HSA yapısı; 3. bölümde PSOA hakkında bilgi verilmiştir. 4. bölümde PSOA kullanılarak HSA şablon katsayılarının optimizasyonu tartışılmıştır. Deneysel sonuçlar ve Sobel [25], Prewitt [26], Canny [36] gibi klasik tekniklerin yanı sıra Sheu [43], Xavier [44] ve Firpi [45] gibi diğer HSA tabanlı kenar çıkarıcılar ile yapılan karşılaştırmalar 5. bölümde verilmiş olup, çıkarım 6. bölümde sunulmuştur.

2. Hüresel Sinir Ağı Yapısı

HSA bilginin giriş, eşik ve başlangıç durumu olarak adlandırılabilir üç adet bağımsız değişken kullanılarak hücre adı verilen doğrusal-olmayan dinamik sistemler tarafından işlendiği bir ağ yapısıdır [1-8]. Aynı zamanda HSA, belirli kurallara göre işlem yapan, bölgesel olarak birbirine bağlı (locally-coupled) hücreler topluluğudur. Bu topluluk içerisindeki her bir dinamik sistem, bir giriş, bir çıkış ve bir durum (state) değerine sahiptir [1-8]. Genelleştirilmiş HSA eşitlikleri aşağıdaki gibi verilebilir.

$$\dot{x}_{ij} = -x_{ij} + \sum_{kl \in S_{ij}(r)} a_{kl} y_{kl} + \sum_{kl \in S_{ij}(r)} b_{kl} u_{kl} + z_{ij} \quad (1)$$

$$y_{ij} = \frac{1}{2} (|x_{ij} + 1| - |x_{ij} - 1|); \quad i = 1, 2, \dots, M, \quad j = 1, 2, \dots, N \quad (2)$$

$$S_{ij}(r) = \{C_{kl} : \max(|k - i|, |l - j|) \leq r, 1 \leq k \leq M, 1 \leq l \leq N\} \quad (3)$$

Yukarıdaki eşitliklerde görülen x_{ij} , y_{ij} , u_{ij} ve z_{ij} sırasıyla bir hücrenin durum, çıkış, giriş ve eşik ifadelerini, $S_{ij}(r)$ hücrenin r yarıçapı komşuluk kümesini oluşturmaktadır. Eşitlik (1)'deki a katsayısı, hücrenin komşu hücre çıkışlarıyla olan bağlantı ağırlıklarını veren geri besleme şablonunu, b katsayısı ise hücrenin komşu girişleriyle olan bağlantı ağırlıklarını veren ileri besleme şablonunu oluşturmaktadır. Bir HSA hücresinin etkinlik alanı ya da etki bölgesi eşitlik (3)'te verilen komşuluk ifadesi ile belirlenmektedir [1-8]. Eşitlik (3)'teki r pozitif bir tam sayıdır ve $r=1$ komşuluğu "3x3 komşuluğu", $r=2$ komşuluğu "5x5 komşuluğu" ve $r=3$ komşuluğu da bir "7x7 komşuluğu" olarak tanımlanır. $r=1$ yarıçap komşuluğunda yani 3x3 boyutlu bir HSA hücresi için ileri besleme ve geri besleme şablon katsayıları aşağıda matris formunda

görülmektedir. Matrislerden de görülebileceği üzere genelleştirilmiş bir HSA hücresi, bir adet eşik $z_{kl}=z$ değeri, 9 adet geri besleme şablon katsayısı a_{kl} ve 9 adet ileri yönde besleme şablon katsayısı b_{kl} ile toplam 19 adet reel sayı tarafından tanımlanabilmektedir.

$$A = \begin{bmatrix} a_{i,j;i-1,j-1} & a_{i,j;i-1,j} & a_{i,j;i-1,j+1} \\ a_{i,j;i,j-1} & a_{i,j;i,j} & a_{i,j;i,j+1} \\ a_{i,j;i+1,j-1} & a_{i,j;i+1,j} & a_{i,j;i+1,j+1} \end{bmatrix} \equiv \begin{bmatrix} a_1 & a_2 & a_3 \\ a_4 & a_5 & a_6 \\ a_7 & a_8 & a_9 \end{bmatrix}$$

$$B = \begin{bmatrix} b_{i,j;i-1,j-1} & b_{i,j;i-1,j} & b_{i,j;i-1,j+1} \\ b_{i,j;i,j-1} & b_{i,j;i,j} & b_{i,j;i,j+1} \\ b_{i,j;i+1,j-1} & b_{i,j;i+1,j} & b_{i,j;i+1,j+1} \end{bmatrix} \equiv \begin{bmatrix} b_1 & b_2 & b_3 \\ b_4 & b_5 & b_6 \\ b_7 & b_8 & b_9 \end{bmatrix}, \quad Z = [z]$$

Yukarıda sayılan 19 reel sayı kullanılarak bir HSA hücresi için şablon diyagramı vektörel olarak eşitlik (4)'teki gibi tanımlanabilir.

$$\mu = [a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, b_1, b_2, b_3, b_4, b_5, b_6, b_7, b_8, b_9, z] \quad (4)$$

HSA üzerine yapılan çalışmalarda, konumsal-değişmezlik varsayımı altında, geri besleme şablon katsayılarının eşitlik (5)'te verildiği gibi simetrik seçilmesi durumunda tasarlanan HSA'nın tümüyle kararlı olduğu belirtilmektedir [1-8].

$$|x_{ij}(0)| \leq 1, |u_{ij}| \leq 1, A(i, j; k, l) = A(k, l; i, j) \quad 1 \leq i \leq M, \quad 1 \leq j \leq N \quad (5)$$

Eşitlik (5)'te verilen simetri kriterine benzer şekilde, sistemin benzetim süresini kısaltmak üzere, ileri besleme şablon katsayıları için de benzer bir simetri öngörülmektedir [46-48]. Buna göre, eşitlik (5)'te verilen kararlılık kriteri ışığında, simetrik şablon katsayıları eşitlik (6)'daki gibi oluşmaktadır.

$$a_i = a_{10-i}, \quad b_i = b_{10-i}, \quad i = 1, 2, 3, 4 \quad (6)$$

Eşitlik (6)'da verilen simetrik şablon katsayılarından yola çıkılarak eşitlik (4) aşağıdaki gibi yeniden yazılabilir.

$$\mu = [a_1, a_2, a_3, a_4, a_5, b_1, b_2, b_3, b_4, b_5, z] \quad (7)$$

3. Parçacık Sürüsü Optimizasyon Algoritması

PSOA, kuş ve balık sürülerinin sosyal dayanışmalarından esinlenilerek geliştirilmiş gelişimsel bir hesaplama tekniğidir [40]. Genetik ve farksal gelişim algoritmasına benzer biçimde, PSOA da popülasyon tabanlı bir optimizasyon algoritmasıdır. Algoritmadaki her bir birey parçacık olarak adlandırılır. Her parçacık, değeri diğer parçacıklardan ve kendisinden edindiği deneyimlere bağlı olarak sürekli olarak güncellenen bir hız değeri ile çok boyutlu uzayda konum değiştirir. Diğer gelişimsel tekniklerle karşılaştırıldığında PSOA gerçekleşmesi kolay ve ayarlanması gereken sadece birkaç parametreye sahip bir algoritma olarak farklılık gösterir.

Bu çalışmada Clerc'in büzmeli PSOA'sı [49] kullanılmıştır. x ve v parçacığın çok boyutlu uzaydaki konum ve hız değerlerini gösterebilir. Algoritmada parçacığın araştırma uzayındaki daha önce ulaşılmış olduğu en iyi konumu kaydedilir ve p^b ile gösterilir. Benzer şekilde tüm parçacıklar arasındaki en iyi konum bilgisi de kaydedilir ve p^g ile gösterilir. Algoritma ile parçacığın hız ve konum değerleri aşağıdaki eşitlikler ile hesaplanır:

$$v_{t+1} = \zeta (v_t + c_1 r_1 (p_t^b - x_t) + c_2 r_2 (p_t^g - x_t)) \quad (8)$$

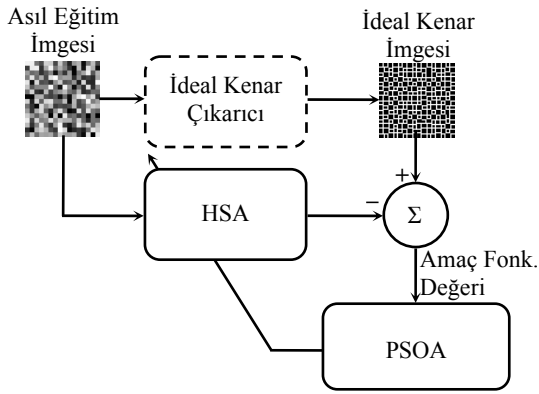
$$x_{t+1} = x_t + v_{t+1} \quad (9)$$

Burada ζ büzme çarpanını, c_1 ve c_2 sabit pozitif hızlanma katsayılarını, r_1 ve r_2 [0,1] aralığından eşdağılımlı bir şekilde seçilen rasgele sayıları, x ve v daha önce de değinildiği gibi parçacığın konum ve hız değerini, t ve $t+1$ ise iterasyon değerlerini göstermektedir. Konum ve hız değerlerine ek olarak algoritmanın büzme çarpanı ζ eşitlik (10) yardımıyla hesaplanır [49].

$$\zeta = \frac{2}{2 - (c_1 + c_2) - \sqrt{(c_1 + c_2)^2 - 4(c_1 + c_2)}} \quad (10)$$

4. PSOA ile HSA Şablon Katsayılarının Belirlenmesi

Şekil 1, HSA şablon katsayılarının belirlenmesine ait düzeneği göstermektedir. Burada HSA'nın şablon katsayıları, HSA'nın çıkış işareti arzu edilen çıkış işaretine yakınsayınca dek optimize edilmektedir. Düzenekteki ideal kenar çıkarıcı blok gerçekte olmayıp, bu bloğun çıkış işareti kenarların ideal bir şekilde belirlenmiş olduğu ideal kenar eğitim imgesini temsil etmektedir.



Şekil 1: HSA şablon katsayılarının optimizasyon düzeneği.

Şekil-2 HSA'nın şablon katsayılarının optimize edilmesi esnasında kullanılan eğitim imgelerini göstermektedir. Şekil-2a kişisel bilgisayar aracılığıyla üretilmiş 64x64 boyutundaki yapay bir imge olan asıl eğitim imgesini göstermektedir. Bu imgedeki her bir kutucuk 4x4 boyutunda olup değerleri 0-255 arasında düzgün dağılıma göre rasgele bir şekilde seçilen aynı parlaklık seviyesine sahip 16 elemandan oluşmaktadır. Şekil-2b, gri renk seviye farklılıklarını temel alarak asıl eğitim imgesindeki kenarların ideal yerleşimlerini temsil eden ideal kenar imgesidir.



Şekil 2: Eğitim için kullanılan imge çifti.

Bu çalışmada kullanılan amaç fonksiyonu, ideal durumda HSA çıkışında elde edilmek istenen ideal kenar imgesi ve şablon katsayıları optimize edilen HSA ile elde edilen kenar imgesi arasındaki ortalama karesel hata değeridir. Bu durum aşağıdaki gibi ifade edilebilir:

$$\text{Amaç Fonk. Değeri} = \text{MSE}(y_{ideal} - \hat{y}) \quad (11)$$

Burada y_{ideal} idealde HSA'nın çıkışından elde edilmek istenen ideal kenar imgesini, \hat{y} ise şablon katsayıları PSOA ile belirlenen HSA'nın çıkışında elde edilen kenar imgesini temsil etmektedir. Bu amaç fonksiyonunun değeri optimizasyon süresince azalan bir seyir izlemektedir. HSA şablon katsayılarının optimizasyonu tamamlandığında, şablon katsayılarına ait değerler sabitlenir.

5. Sonuçlar

Sunulan HSA tabanlı kenar çıkarıcının başarımını karşılaştırmak için üç farklı klasik kenar çıkarıcısı (Sobel, Prewitt, Canny) [25,26,36] ve ayrıca üç farklı HSA tabanlı kenar çıkarıcısı (Sheu, Xavier, Firpi) [43-45] kullanılmıştır. [43]'de HSA'nın optimal şablon katsayıları donanımsal işlem algoritması ile elde edilmiştir. [43]'de kullanılan şablon katsayıları [50]'deki Matlab MatCNN InstantVision araç kutusunda kullanılan şablon katsayıları ile aynıdır. Diğer taraftan adaptif ısı işlem algoritması da [51,52] kenar çıkarımı amacıyla HSA şablon katsayılarının belirlenmesi amacıyla kullanılmıştır [44]. Karşılaştırma amacıyla kullanılan üçüncü HSA tabanlı çalışmada [45] ise bu çalışmaya benzer bir şekilde PSOA kullanılmıştır.

Bu çalışmalardan [43,45]'te şablon katsayıları eşitlik (4)'ü sağlayacak şekilde aşağıdaki gibi seçilmiştir:

$$\begin{aligned} a_1 = a_2 = a_3 = a_4 = a_6 = a_7 = a_8 = a_9 = 0; & \quad a_5 = a \\ b_1 = b_2 = b_3 = b_4 = b_6 = b_7 = b_8 = b_9 = b; & \quad b_5 = b_0 \\ z = z & \end{aligned} \quad (12)$$

Buradan hareketle eşitlik (4)'teki şablon katsayısı optimizasyon vektörü aşağıdaki gibi olmaktadır:

$$\mu = [a b b_0 z] \quad (13)$$

[42]'deki şablon katsayısı optimizasyon vektörü formu bu çalışmada korunmakla birlikte bu vektöre zaman adımı (timestep) ve iterasyon (iter) değerleri de eklenerek aşağıdaki gibi seçilmiştir:

$$\mu = [a_1 a_2 a_3 a_4 a_5 b_1 b_2 b_3 b_4 b_5 z \text{ timestep iter}] \quad (14)$$

Tablo 1'de karşılaştırmada kullanılan ve bu çalışmada sunulan HSA tabanlı kenar çıkarıcılara ait şablon katsayıları verilmiştir. Tablo 1'de verilen şablon katsayılarına ek olarak bu çalışmada ulaşılan optimal zaman adımı değeri ve iterasyon değeri sırasıyla 1.3592 ve 1 olarak bulunmuştur. HSA yapısı ile ilgili benzetimler için Matlab MatCNN InstantVision araç kutusu [50] kullanılmıştır.

HSA tabanlı kenar çıkarıcının başarımını değerlendirmek amacıyla hem eğitim imgesi ve bir ikili imge, hem de iki adet gri seviyeli imge üzerinde detaylı kenar çıkarma benzetimleri yapılmıştır. Benzetimlerde kullanılan giriş imgeleri Şekil 3'te ilk hücrelerde verilmiştir. Bu giriş imgeleri optimizasyon esnasında kullanılan yapay orjinal eğitim imgesi, yapay bir ikili imge, popüler texmos3.s512 [53] ve Aluminum gri seviyeli test imgeleridir. Karşılaştırmada kullanılan kenar çıkarıcılara ve sunulan kenar çıkarıcıya ait çıktı imgeleri Şekil 3'te verilmektedir. Bu şekilden görülebileceği gibi birinci dereceden türev tabanlı kenar çıkarıcılar ve (Sobel, Prewitt), gaussian tabanlı kenar çıkarıcı (Canny) benzer kenar çıkarma başarımlarına sahiptirler. Bu kenar çıkarıcılar arasından Sobel ve Prewitt operatörlerine ait çıktı imgeleri birtakım düzensizliklere ve devamsızlıklara (kenarlarda kesintilere)

sahiptir. Canny kenar çıkarıcısı bu operatörlere kıyasla daha iyi bir başarıma sahiptir. Sheu ve Firpi HSA tabanlı kenar çıkarıcıları texmos3.s512 ve Aluminum test imgelerinde daha düşük başarımla sergilemektedir. Xavier HSA tabanlı kenar çıkarıcı, Firpi ve Sheu'nun operatörlerine göre daha iyi bir başarıma sahiptir. Şekil 3'ten görüldüğü gibi, en iyi başarıma sahip kenar çıkarıcı sunulan HSA tabanlı kenar çıkarıcıdır.

| | | |
|---|---|---|
| | [43,50] | [44] |
| A | $\begin{bmatrix} 0 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix}$ | $\begin{bmatrix} -0.2367 & -0.2416 & -0.2207 \\ -0.1710 & 1.8889 & -0.1710 \\ -0.2207 & -0.2416 & -0.2367 \end{bmatrix}$ |
| B | $\begin{bmatrix} -0.25 & -0.25 & -0.25 \\ -0.25 & 2 & -0.25 \\ -0.25 & -0.25 & -0.25 \end{bmatrix}$ | $\begin{bmatrix} -0.6522 & 0.0306 & -0.8290 \\ -0.0722 & 2.8511 & -0.0722 \\ -0.8290 & 0.0306 & -0.6522 \end{bmatrix}$ |
| Z | -1.5 | -1.3631 |
| | [45] | Sunulan |
| A | $\begin{bmatrix} 0 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix}$ | $\begin{bmatrix} -5.9669 & -2.3573 & -5.9926 \\ -0.5309 & 14.8274 & -0.5309 \\ -5.9926 & -2.3573 & -5.9669 \end{bmatrix}$ |
| B | $\begin{bmatrix} -0.968 & -0.968 & -0.968 \\ -0.968 & 7.670 & -0.968 \\ -0.968 & -0.968 & -0.968 \end{bmatrix}$ | $\begin{bmatrix} 0.0432 & -0.0312 & 0.0009 \\ -0.0544 & 15.0000 & -0.0544 \\ 0.0009 & -0.0312 & 0.0432 \end{bmatrix}$ |
| Z | -2.617 | 1.1962 |

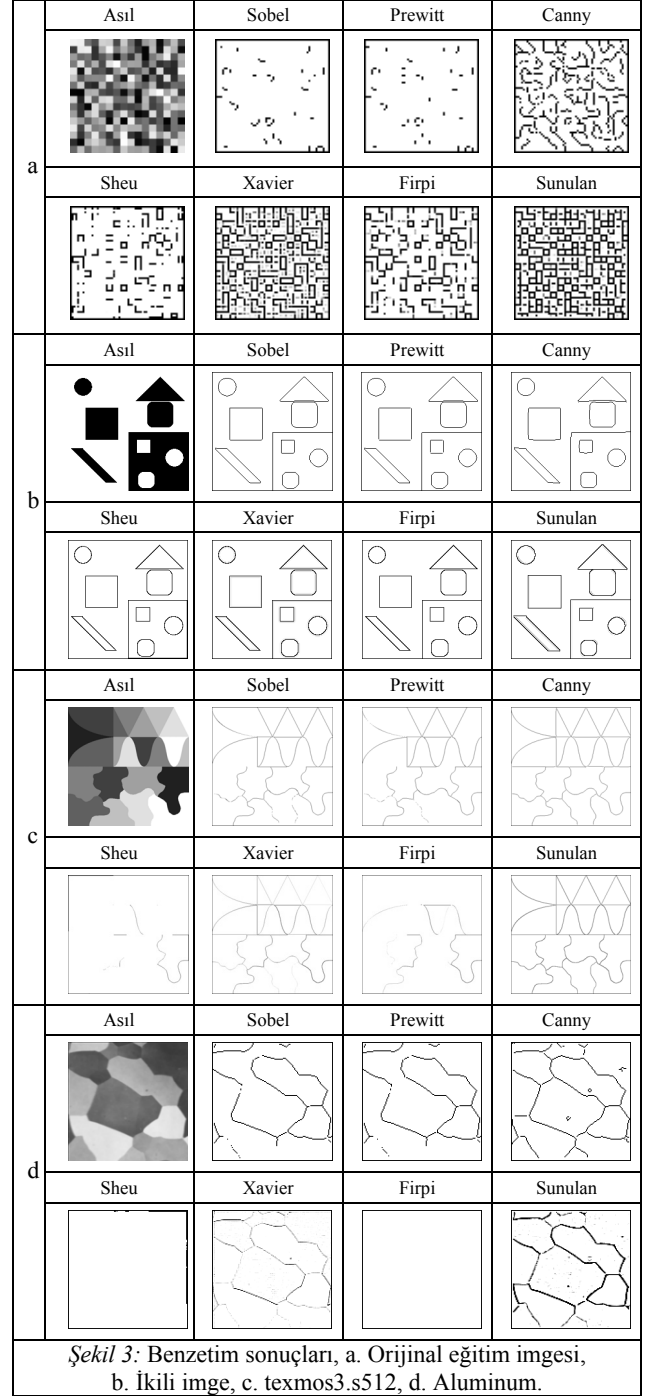
6. Çıkarım

Bu çalışmada, sayısal imgelerde kenar çıkarımı amacıyla PSOA tabanlı bir HSA şablon katsayısı belirleme yöntemi sunulmuştur. Sunulan HSA tabanlı kenar çıkarıcının başarımları hem bir ikili imgede hem de gri seviyeli popüler test imgeleri üzerinde denenmiştir. Sunulan kenar çıkarıcının temel avantajı literatürdeki kenar çıkarıcılara göre kenarları daha etkin bir şekilde elde etmesidir. HSA tabanlı kenar çıkarıcının avantajları şöyle özetlenebilir: i) HSA basit bir yapıya sahiptir. Davranışı birinci dereceden bir diferansiyel denklem ile karakterize edilebilmektedir; ii) Sadece kenar çıkarma amacıyla değil, farklı imge işleme işlevleri için de şablon katsayıları optimize edilerek HSA tabanlı operatörler kullanılabilir.

Sunulan HSA tabanlı kenar çıkarıcı sayısal imgelerde kenarların etkin bir şekilde çıkarımı amacıyla basit ancak güçlü bir araç olarak kullanılabilir.

7. Kaynaklar

- [1] Chua, L. O.; Yang, L.: Cellular neural networks: Theory. IEEE Transactions on Circuits and Systems-I: Fundamental Theory And Applications 35 (1988), 1257-1272.
- [2] Chua, L. O.; Yang, L.: Cellular neural networks: Applications. IEEE Transactions on Circuits and Systems-I: Fundamental Theory And Applications 35 (1988), 1273-1290.
- [3] Chua, L. O.: A paradigm for complexity. Vol. 31. World Scientific Publishing Co. Pte. Ltd., 1998. - (World Scientific Series on NonLinear Science, Series A).
- [4] Dogaru, R.: Universality and emergent computation in cellular neural networks. Vol. 43. World Scientific Publishing Co. Pte. Ltd., 2003. - (World Scientific Series on NonLinear Science, Series A).



Şekil 3: Benzetim sonuçları, a. Orijinal eğitim imgesi, b. İkili imge, c. texmos3.s512, d. Aluminum.

- [5] Frasca, M.; Arena, P.; Fortuna, L.: Bio-inspired emergent control of locomotion systems. Vol. 48. World Scientific Publishing Co. Pte. Ltd., 2004. - (World Scientific Series on NonLinear Science, Series A).
- [6] Arena, P.; Fortuna, L.: Analog cellular locomotion control of hexapod robots. IEEE Control Systems Magazine 22 (2002), 21-36.
- [7] Arena, P.; Fortuna, L.; Frasca, M.; Patane, L.: A cnn-based chip for robot locomotion control. IEEE Transactions on Circuits and Systems-I: Fundamental Theory And Applications 52 (2005), 1862-1871.
- [8] Arena, P.; Bonomo, C.; Fortuna, L.; Frasca, M.; Graziani, S.: Design and control of an ipmc wormlike robot. IEEE

- Transactions on Systems, Man and Cybernetics, Part B 36 (2006), 1044–1052.
- [9] Arena, P.; Bucolo, M.; Fortuna, L.; Occhipinti, L.: Cellular neural networks for real-time dna microarray analysis. *IEEE Engineering in Medicine and Biology Magazine* 21 (2002), 17–25.
 - [10] Arena, P.; Fortuna, L.; Occhipinti, L.: A cnn algorithm for real time analysis of dna microarrays. *IEEE Transactions on Circuits and Systems–I: Fundamental Theory And Applications* 49 (2002), 335–340.
 - [11] Matsumoto, T.; Chua, L. O.; Suzuki, H.: Cnn cloning template: Connected component detector. *IEEE Transactions on Circuits and Systems–I: Fundamental Theory And Applications* 37 (1990), 633–635.
 - [12] Matsumoto, T.; Chua, L. O.; Furukawa, R.: Cnn cloning template: Hole filler. *IEEE Transactions on Circuits and Systems–I: Fundamental Theory And Applications* 37 (1990), 635–638.
 - [13] Matsumoto, T.; Chua, L. O.; Yokohoma, T.: Image thinning with a cellular neural network. *IEEE Transactions on Circuits and Systems–I: Fundamental Theory And Application* 37 (1990), 638–640.
 - [14] Crouse, K. R.; Chua, L. O.: Methods for image processing and pattern formation in cellular neural networks: A tutorial. *IEEE Transactions on Circuits and Systems–I: Fundamental Theory And Applications* 42 (1995), 583–601.
 - [15] Chua, L. O.; Roska, T.; Kozek, T.; Zarandy, A.: *The cnn paradigm*. Baffins Lane, Chichester, West Sussex PO191UD England: John Wiley & Sons Ltd., 1993. – (Cellular Neural Networks).
 - [16] Chua, L. O.; Roska, T.: The cnn paradigm. *IEEE Transactions on Circuits and Systems–I: Fundamental Theory And Application* 40 (1993), 147–156.
 - [17] Zarandy, A.: The art of cnn template design. *Int. Journal of Circuit Theory and Applications* 27 (1999), 5–23.
 - [18] Yükksek, M. E. (2007). Edge detection in noisy images by neuro-fuzzy processing. *International Journal of Electronics and Communications (AEÜ)*, 61(2), 82–89.
 - [19] Yükksek, M. E., & Yildirim, M. T. (2004). A Simple Neuro-Fuzzy Edge Detector for Digital Images Corrupted by Impulse Noise. *International Journal of Electronics and Communications (AEÜ)*, 58(1), 72–75.
 - [20] Stern, D.; Kurz, L.: Edge detection in correlated noise using latin squares models. *Pattern Recognition* 21 (1988), 119–129.
 - [21] Haberstroh, J.; Kurz, L.: Line detection in noisy and structured background using graco–latin squares. *CVGIP: Graphical Models Image Process* 55 (1993), 161–179.
 - [22] Nahi, N. E.; Assefi, T.: Bayesian recursive image estimation. *IEEE Trans. Comput.* 7 (1972), 734–738.
 - [23] Hansen, F. R.; Elliot, H.: Image segmentation using simple markov field models. *Comput. Graphics Image Process* 20 (1982), 101–132.
 - [24] Huang, J. S.; Tseng, D. H.: Statistical theory of edge detection. *Comput. Vision Graphics Image Process.* 43 (1988), 337–346.
 - [25] Sobel, I.: Neighbourhood coding of binary images fast contour following and general array binary processing. *Computer Graphics and Image Processing* 8 (1978), 127–135.
 - [26] Prewitt, J. M. S.: *Object enhancement and extraction*. New York: Academic Press, 1970. – (Picture Processing and Psychopictorics).
 - [27] Kirsch, R. A.: Computer determination of the constituent structure of biological images. *Comput. Biomed. Res.* 4 (1971), 314–328.
 - [28] Marr, D.; Hildreth, E.: Theory of edge detection. *Proceedings of the Royal Society London*, 1980. 187–217.
 - [29] Haralick, R. M.: Digital step edges from zero crossing second directional derivatives. *IEEE Trans. Pattern Anal. Mach. Intell.* PAMI–6 (1984), 58–68.
 - [30] Huechel, M. H.: An operator which locates edges in digitized pictures. *J. Assoc. Comput. Mach.* 18 (1971), 113–125.
 - [31] Haralick, R. M.; Watson, L.: A facet model for image data. *Comput. Graphics Image Process.* 15 (1981), 113–129.
 - [32] Nalwa, V.; Binford, T. O.: On detecting edges. *IEEE Trans. Pattern Anal. Mach. Intell.* PAMI–8 (1986), 699–714.
 - [33] Ji, Q.; Haralick, R. M.: Efficient facet edge detection and quantitative performance evaluation. *Pattern Recognition* 35 (2002), 689–700.
 - [34] Umbaugh, S. E.: *Computer vision and image processing*. Englewood Cliffs, NJ: Prentice–Hall International Inc., 1998.
 - [35] Bovik, A.C.: *Handbook of image and video processing*. New York: Academic Press, 1998.
 - [36] Canny, J.: A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI–8 (1986), 679–698.
 - [37] Linares, I.; Maersereau, R. M.; Smith, M. J. T.: Jpeg estimated spectrum adaptive post filter using image–adaptive q–tables and canny edge detectors. *ISCAS–96, IEEE International Symposium on Circuits and Systems*, Atlanta, GA, USA, 1996. 722–725.
 - [38] Ali, M.; Clausi, D.: Using the canny edge detector for feature extraction and enhancement of remote sensing images. *IGARSS–01, IEEE International Symposium on Geoscience and Remote Sensing*, Sydney, Australia, 2001. 2298–2300.
 - [39] Hongjian, S.; Ward, R.: Canny edge based image expansion. *ISCAS–2002, IEEE International Symposium on Circuits and Systems*, Scottsdale, AZ, USA, 2002. 785–788.
 - [40] Kennedy, J.; Eberhart, R.: Particle swarm optimization. *Int. Conf. Neural Networks*, Perth, Australia, 1995. 1942–1948.
 - [41] Liu, B.; Wang, L.; Jin, Y.-H.; Tang, F.; Huang, D.-X.: Improved particle swarm optimization combined with chaos. *Chaos, Solitons & Fractals* 25 (2005), 1261–1271.
 - [42] Selleri, S.; Mussetta, M.; Pirinoli, P.; Zich, R. E.; Matekovits, L.: Some insight over new variations of the particle swarm optimization method. *Antennas and Wireless Propagation Letters* 5 (2006), 235–238.
 - [43] Sheu, B. J.; Chang, R. C.; Wu, T. H.; Bang, S. H.: Vlsi-compatible cellular neural networks with optimal solution capability for optimization. *IEEE International Symposium on Circuits and Systems*, ISCAS’95, 1995. 1165–1168.
 - [44] Xavier-de Souza, S.; Yalcin, M. E.; Suykens, J. A. K.; Vandewalle, J.: Toward cnn chip-specific robustness. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications* 51 (2004), 892–902.
 - [45] Firpi, H. A.; Goodman, E. D.: Designing templates for cellular neural networks using particle swarm optimization. In *proc. 33rd Applied Imagery Pattern Recognition Workshop*, 2004. 119–123.
 - [46] Guzelis, C.: Supervised learning of the steady-state outputs in generalized cellular networks. *CNNA-92 Second International Workshop on Cellular Neural Networks and their Applications*, 1992. 74–79.
 - [47] Guzelis, C.; Karamahmut, S.: Recurrent perceptron learning algorithm for completely stable cellular neural networks. *CNNA-94 Third International Workshop on Cellular Neural Networks and their Applications*, 1994. 177–182.
 - [48] Guzelis, C.; Karamahmut, S.: Recurrent perceptron learning algorithm for cnns with application to edge detection. *IEEE International Conference on Neural Networks*, 1995. 1134–1139.
 - [49] Clerc, M.; Kennedy, J.: The particle swarm: explosion, stability, and convergence in a multi-dimensional complex space. *IEEE Transactions on Evolutionary Computation* 6 (2002), 58–73.
 - [50] *Matcnn InstantVision Toolbox for Matlab*, <http://www.eutecus.com/Downloads/IVToolboxes.html>.
 - [51] Ingber, L.: Very fast simulated re-annealing. *J. Math. Comput. Model.* 12 (1989), 967–973.
 - [52] Ingber, L.: Adaptive simulated annealing (asa) version 24.1. <http://www.ingber.com>.
 - [53] *Usc-sipi texture database*, university of southern california, signal & image processing institute. <http://sipi.usc.edu/database/database.cgi?volume=textures>.