

Model Tabanlı Yapay Sinir Ağı Kullanılarak Geliştirilmiş Bir Kenar Çıkarma Metodu

F. Talu, Y. Tatar
Firat Üniversitesi Bilgisayar Müh. Bölümü, Elazığ
fatihtalu@firat.edu.tr, ytatar@firat.edu.tr

Özetçe

Bu makalede, Model Tabanlı Yapay Sinir Ağı (MYSA) kullanarak sayısal görüntülerde kenar çıkarma problemine etkili bir çözüm önerisi getirilir. Bu çalışmanın amacı, kenar çıkarma problemi gibi lineer olmayan problemleri öğrenmek için, MYSA'nın adaptasyon yeteneğini kullanmanın yararlarını sunmaktır. Kullanılan MYSA sayesinde gri seviyeli görüntüler iki seviyeli kenar görüntülerine dönüştürülür. Elde edilen kenar sonuçları popüler kenar çıkarıcı olarak bilinen Canny [1] algoritmasının kenar sonuçları ile karşılaştırılır.

Absract

A Developed Edge Detection Method Using Model-Based Artificial Neural Network; In this paper, we present an effective method for edge detection problem using model-based neural network (MBNN). The main motivation of this work is to show the advantages of using the adaptation capacity of MBNN to learn non-linear problems such as edge detection. Edge detection process carried out using MYSA has two steps. Firstly, a gri level image is transformed into two-level in any way. Secondly, each window of two level image is decided if it is an edge or not. The obtained results are compared to the results of Canny [1] algorithm that is known as popular edge detection algorithm.

1. Giriş

Kenar çıkarma, nesne tanımda çok önemli rol oynar. İnsan görme sisteminde, nesnelerin geometrik özelliklerini tanıma süreci, nesnelerin dış çizgilerine göz gezdirilmesi ile başlar. Bu yaklaşım yapay görme sistemlerine uyarlandığında, şayet nesnenin sınırları başarılı bir şekilde izlenebilirse nesne tanımda çok daha iyi sonuçlar alınabileceği açıktır. Buna göre, nesne tanıma işleminde kenar çıkarma işlemi önemli bir rol oynar. Çoğu görüntüler somut nesnelere içermez ve bu görüntüleri anlamak onların yapısal özelliklerine bağlıdır. Yapısal özelliklerin çıkarımı ise kenar çıkarma ile ilgilidir [2].

Kenar çıkarma, özellik çıkarma işleminde önemli bir alt problemdir. Amacı, görüntü içerisinde komşuları ile farklı yoğunluklara sahip piksellerin tespit edilmesidir. Bu işlemin genellikle iki aşamadan oluşur. Birinci aşamada, güncel pikselin yoğunluğu, komşu pikseller ile karşılaştırılarak yoğunluk değişimleri tespit edilir. Geleneksel kenar çıkarıcılar olarak bilinen Sobel operatörü, Prewitt operatörü ve Robert operatörü [3, 4] bu aşamaya ihtiyaç duyarlar. İkinci aşamada ise, elde edilen değişimlerin önceden belirlenmiş bir eşik değeri ile karşılaştırılarak "önemli" bir değişim olup olmadığı tespit edilir ve bu tespite göre güncel pikselin kenar olup olmadığı hakkında karar verilir.

Kenar çıkarma algoritmalarında iki önemli özelliğin olması gerekmektedir [5]. Bunlardan birincisi, kenar çıkarma işleminin sonucu, kenar yönlerinden bağımsız olmalıdır. Yani çıkarılan kenar bilgisi, kenarı çıkarılan nesne yön değiştirdiğinde de değişime uğramamalıdır. İkincisi ise, çıkarılan kenar çizgisi bir piksel (nokta) kalınlığında olmalıdır. Böylece elde edilen kenar bilgisi hafızada daha az yer kaplayacaktır.

Geleneksel kenar çıkarma algoritmaları, istenilen bu özelliklere çoğu zaman doyurucu bir sonuç vermezler. Bu algoritmalarda sık sık yanlış kenar yeri, köşeleri tam çıkarmama ve bulanık kenar çıkarma gibi problemler ile karşılaşılır [6]. Geleneksel kenar çıkarma algoritmaları içerisinde en iyilerden biri olarak bilinen Canny kenar çıkarma algoritmasına göre:

- İyi kenar çıkarma: Yüksek oranda kenar olma olasılığına sahip noktaları "kenar", düşük oranda kenar olma olasılığına sahip noktaları ise "kenar değil" olarak sınıflandırmak.
- İyi kenar yeri belirleme: Geçerli kenar yerine yakın olan noktaları kenar noktası olarak işaretlemek.
- Tek nokta kalınlığında kenar çıkarma: Tek boyutlu bir işarette kenar çizgisini tek nokta kalınlığında göstermek.

Canny kenar çıkartma algoritması bu kriterlere doyurucu bir yanıt verebilmek için kenar çıkartma işlemine başlamadan önce görüntüyü Gaussian fonksiyonunun ilk türevine karşılık gelen bir filtreden geçirir. Görüntü içerisindeki gürültü elendikten sonra filtrelenmiş görüntüye Canny operatörü uygulanır. Genellikle Canny operatörü uygulandıktan sonra elde edilen kenar görüntüsü istenmeyen kenar bilgileri içermektedir. Bu bilgiler nesnelerin yanlış algılanmasına yol açar. İstenmeyen bu bilgileri elemek için non-maxima [6] baskısı uygulanır.

Bu çalışmadaki amacımız MYSA kullanılarak geliştirilen kenar çıkarma yöntemlerinin geleneksel kenar çıkarma yöntemlerinden daha iyi sonuç verdiğini gösteren bazı deney sonuçları sunmaktır. MYSA'nın iyi öğrenme yeteneği kullanılarak sayısal görüntülerde bu dezavantajları ortadan kaldıran veya azaltan makul bir yaklaşım

sunulur. Bunun nedeni, lineer olmayan problemlerin çözümünde YSA'nın adaptasyon kabiliyetinin ispatlanmış oluşudur. Bununla birlikte görüntü işleme ile alakalı diğer uygulamalarda da YSA kullanımı iyi bir performans gösterir. Örneğin görüntü iyileştirme, düzenleme, bölümlenme gibi alanlarında YSA'nın sıklıkla kullanıldığı görülür [6, 7, 8].

2. Kenar çıkarma için kullanılan MYSA mimarisi

MYSA mimarisi alt ağların sınıflanmasını içeren hiyerarşik bir yapıdan oluşur. Her bir alt ağ giriş örneğinin bir kısmını öğrenme görevini üstlenir [9]. Öğrenme aşamasında, her bir alt ağ öğrenme kümesindeki farklı yapıya sahip örnekleri kodlayacaktır. Tanıma aşamasında ise her bir örnek için "kenar" veya "kenar değil" kararını verebilmek, değişik alt ağların çıkışlarını bir kenar çıkarma algoritmasından geçirmekle mümkündür.

Bu mimarinin benimsenmesindeki temel maksat kenar çıkarma işlemindeki gözlemlerimizdir. Kenar çıkarma işleminde tek eşik değeri kullanmak yerine farklı lokal bilgilere karşılık gelen çoklu eşik değeri kümesini benimsemek daha etkilidir [10].

Kullanılan mimari ile adaptif karar verme işleminin üstesinden gelebilmek için her bir alt ağ farklı bir arka plan parlaklık değerine sahiptir. Alt ağlar içerisindeki her bir birim ise alt ağın parlaklık değeri altındaki farklı kenar örneklerine karşılık gelir.

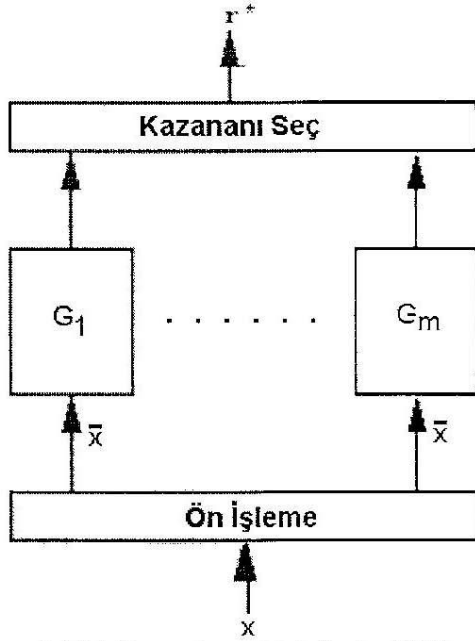
2.1. G_r Alt Ağı

Her bir G_r alt ağı, 2 tane giriş hücresi ve 9 tane orta katman hücresine sahip bir yapay sinir ağıdır. Görüntü içerisindeki farklı yapıya sahip alanlar bu alt ağlarda kodlanır. $r=1, \dots, m$ olmak üzere her bir G_r alt ağına, g_r arka plan parlaklık değeri verilir. Görüntüdeki her bir piksel 4 komşuluğundaki piksel değerleri ve bu değerlerin ortalaması ile beraber düşünülür. Buna göre her bir giriş örneği, $x=[x_1, \dots, x_5]$ ve \bar{x} ortalama değerlerinden oluşur. Bu giriş örneklerinin alt ağlara yönlendirme işlemi ise; giriş örneğinin \bar{x} ortalama değerine bakılarak g_r parlaklık değerine en yakın olan G_r alt ağına yönlendirilmesi şeklindedir. Farklı parlaklık seviyelerine ilişkin adaptif öğrenme görevini

başarabilmek için \bar{x} giriş örneklerinin ortalaması ile alt ağırlık parlaklık değeri arasında aşağıdaki gibi bir ilişki olmalıdır.

$$|\bar{x} - g_{r^*}| < |\bar{x} - g_r| \quad r = 1, \dots, m, r \neq r^* \quad (1)$$

Bu sayede görüntü içerisindeki tüm 3×3 'lük pencereler farklı parlaklık değerlerine sahip alt ağlara yönlendirilir. Böylece görüntüyü tek bir bütün olarak ele almak yerine, benzer özelliklere sahip alanları bir küme altında toplayarak modeller bir yapay sinir ağı eğitimi amaçlanmıştır. Bu kümeleme işlemi Şekil-2 de gösterilir.



Şekil-2: Kenar çıkarma için kullanılan MYSA mimarisi

2.2. G_r Alt Ağındaki N_{rs} hücresi

Her bir G_r alt ağı $s=1, \dots, n$ olmak üzere n tane N_{rs} hücre içerir. Her bir hücre, g_r genel parlaklık değeri altındaki farklı kenar örneklerini gösteren kenar şablonlarını kodlar. Her bir hücre bir $w_{r,s} = [w_{r,s,1}, w_{r,s,2}]$ ağırlık vektörü ile ilişkilendirilir. $w_{r,s}$ ağırlık vektörü 3×3 boyutundaki pencereler içerisindeki iki temel gri seviye değerini belirler.

Her bir pencere için $m = [m_1, m_2]^T$ vektörü tanımlanır. Eğer aşağıdaki şart gerçekleşirse m vektörüne sahip her bir pencere r ağı altındaki N_{rs} hücresi ile ilişkilendirilebilir.

$$\|m - w_{r,s^*}\| < \|m - w_{r,s}\| \quad s = 1, \dots, S \quad s \neq s^* \quad (2)$$

Bu çalışmada $s=2$ olduğu kabul edilir. Böylece her alt ağda iki tane N_{rs} hücresi olacaktır. Hücrelerden biri giriş örneğinin zayıf kenar olma olasılığını gösterir, diğeri ise güçlü kenar olma olasılığını gösterir. Buna karşılık $s=1$, olmak üzere w_{r,s^*} ağırlık vektörlerinden w_{r,s^*} zayıf kenar örneğinin ağırlığını, w_{r,s^*} ise kuvvetli kenar örneğinin ağırlığını gösterir. w_{r,s^*} zayıf kenar örneği geleneksel kenar çıkarma algoritmalarında görülebilir en düşük limit seviyesi olan eşik değeri ile benzer bir rol oynar. Bahsedilen alt ağı yapısı Şekil-3 de gösterilir.

2.3. İkili kenar bilgisi

Kenar çıkarma işlemi güncel pikselin $N \times N$ komşuluğunda gerçekleşir. Gri seviye değerlerine karşılık gelen $x = [x_1, x_2, \dots, x_{N^2}]^T \in R^N$ vektörünün ortalama değeri aşağıdaki gibidir.

$$\bar{x} = \frac{1}{N^2} \sum_{n=1}^{N^2} x_n \quad (3)$$

Verilen bu ortalama değeri, yerel penceredeki gri seviye bilgisini, pencere içerisindeki iki temel gri seviye değerine tekabül eden $m = [m_1, m_2]^T \in R^2$ vektörünü kullanarak karakterize edebilir. Ve aşağıdaki gibi tanımlanır.

$$m_1 = \frac{\sum_{i=1}^{N^2} I(x_i < \bar{x}) x_i}{\sum_{i=1}^{N^2} I(x_i < \bar{x})} \quad (4)$$

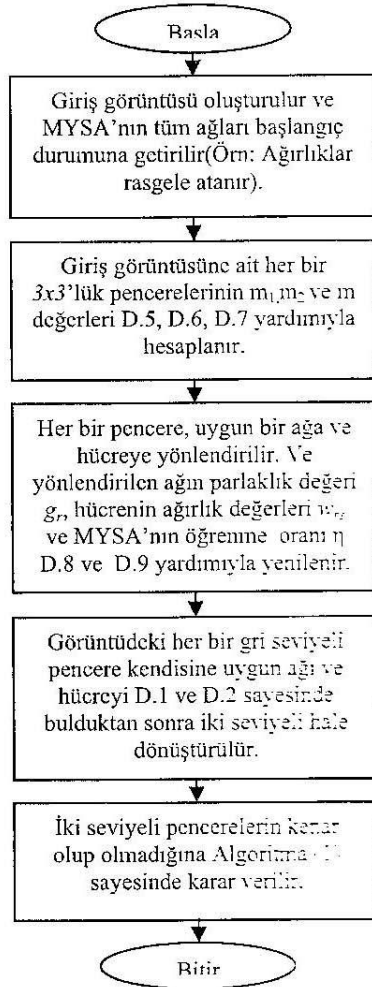
$$m_2 = \frac{\sum_{i=1}^{N^2} I(x_i \geq \bar{x}) x_i}{\sum_{i=1}^{N^2} I(x_i \geq \bar{x})} \quad (5)$$

```

BEGIN
h(x,y) = 0;
Eğer p(x,y) = 1;
    Eğer p(x,y+1) = 0 veya p(x,y-1) = 0
    veya p(x+1,y) = 0 veya p(x-1,y) = 0
        h(x,y) = 1;
END

```

Şekil-4: İki seviyeli görüntülerde kenar çıkarma metodu



Şekil-5: MSYA kullanarak kenar çıkarma metodunun akış şeması.

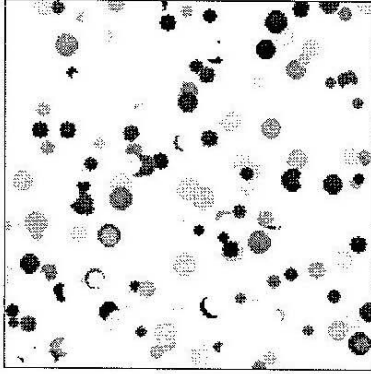
3. Deneysel Sonuçlar

Kullanılan MYSA'nın sonuçları ile geleneksel kenar çıkarma algoritmaları içerisinde en ideallerden biri olarak bilinen Canny kenar çıkarma algoritmasının sonuçlarını karşılaştırmak için iki tane görüntü seçilmiştir. Birincisi, Şekil-6'da gösterilen, farklı yoğunluk değerlerine sahip dairelerden oluşan, 255x255 piksel boyutunda, her bir pikseli 8 bit ile ifade edilen ve herhangi bir görüntü içermeyen yapay bir görüntüdür. Seçilen ikinci görüntü ise, Şekil-7'de gösterilen, görüntü işleme alanında en fazla tercih edilen görüntülerden biri olan, 255x255 piksel boyutunda, görüntü içeren "Lena" görüntüsüdür.

Şekil-8 (a), Şekil-6'daki yapay görüntünün arzu edilen ideal kenar görüntüsünü gösterir. Kullanılan MYSA algoritması için R (alt ağ sayısı) = 2, lr (öğrenme oranı) = 0.2 ve N (eğitim döngü sayısı) = 25 değerleri seçilmesi durumunda elde edilen kenar sonucu Şekil-8(c)'de gösterilir.

Canny kenar çıkarma algoritması α filtre katsayısı, t_1 ve t_2 eşik parametreleri olmak üzere üç tane parametreye ihtiyaç duyar. Bu parametreler arasındaki olası kombinasyonların sayısı oldukça büyüktür. Bu kombinasyonlar arasından orijinal görüntünün kenar sonucuna en yakın olan değerleri alınır.

Genellikle, t_1 ve t_2 değerleri düşük seçildiğinde daha fazla detay silinir. Aynı zamanda istenen kenarlarda da büyük bir azalma olur. Diğer durumda, eşik değerleri yüksek tutulduğunda istenmeyen kenarlar elde edilir. Bu çalışmada $t_1=20$, $t_2=25$ ve $\alpha=0.3$ seçildiği takdirde elde edilen kenar görüntüsünün arzu edilen kenar görüntüsüne en yakın sonuçlardan biri olduğu görülür. Bu değerler doğrultusunda Canny algoritmasından elde edilen kenar sonucu Şekil-8(e)'de gösterilir.



Şekil-6: Yapay görüntü



Şekil-7: Orijinal lena görüntüsü

Elde edilen sonuçların daha iyi analiz edilebilmesi için Şekil-8(b)'de arzu edilen kenarın, Şekil-8(d)'de MYSA'nın ve Şekil-8(f)'de ise Canny kenar çıkarma algoritmasının ($\alpha=0.3$, $t_1=20$, $t_2=25$) sonuçlarının sadece belli bir parçasının büyütülmüş şekli gösterilir. Buna göre, Şekil-8(d)'deki kenar sonucu gösterir ki; MYSA, zemin üzerindeki her hangi bir dairenin kenar piksellerini hatasız bir şekilde tespit edebilirken, birden fazla dairenin üst üste gelmesiyle oluşan kesişme noktalarındaki kenar piksellerini tam olarak tespit edemez. Bu piksellerin tespit edilmesi MYSA'nın eğitim döngü sayısı ile doğrudan alakalıdır. Eğitim döngü sayısı arttıkça kesişme noktalarındaki çıkarılmayan piksel sayısı azalacaktır.

Şekil-8(f)'deki kenar görüntüsü, bizlere Canny algoritmasının sonucunun MYSA'nın sonucu kadar hassas olmadığını gösterir. Hazırlanan yapay görüntüde, Canny'nin MYSA'ya göre daha fazla yanlış kenar yeri tespit ettiği gözlenir. Böylece, MYSA'nın kenar sonucu arzu edilen kenar görüntüsüne daha yakındır.

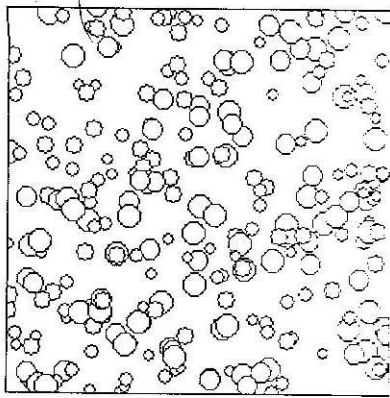
Doğal bir görüntü olan ve bir çok detay bilgisi ve gürültü içeren, gri seviyeli ve 255×255 piksel boyutundaki "Lena" görüntüsüne karşı MYSA'nın verdiği yanıt Şekil-9'da verilmiştir. Aynı görüntü için, Canny algoritmasının verdiği yanıt ise Şekil-10'de gösterilir. Şekil-9'daki kenar sonucu, Şekil-10'deki kenar sonucuna göre daha güçlü olmasına rağmen daha fazla hatalı kenar içermektedir. Bunun nedeni, Canny algoritması kenar çıkarma işlemine başlamadan önce görüntüdeki gürültüyü azaltmak için α katsayılı bir filtreden geçirmesine karşılık MYSA yönteminde giriş filtresinin kullanılmamasıdır.

4. Sonuç

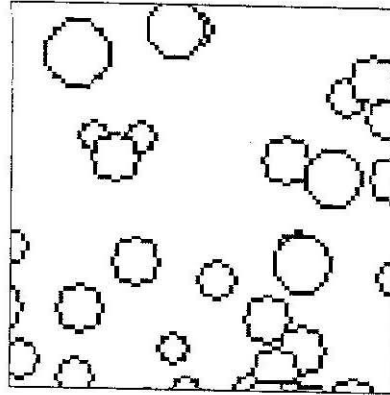
Elde edilen kenar görüntüleri incelendikten sonra, yapay görüntülerde MYSA mimarisi kullanılarak hazırlanan kenar çıkarıcıların geleneksel metotlardan çok daha doyurucu ve ümit verici sonuçlar verdiği sonucuna varılır. Ayrıca benimsenen mimari sayesinde kenar elementleri arasında kabul edilebilir ilişkilerde ve önemli kenar yapılarının olduğu noktalarda daha güçlü sonuçlar elde edilir.

Bundan sonraki hedefimiz, kenar çıkarma özelliğinin yanı sıra filtreleme işlemini de gerçekleştirebilen bir MYSA tasarlamaktır. Bu sayede, sadece yapay görüntülerden değil gerçek hayatta kullanılan doğal görüntülerden de arzu edilen sonuçlara ulaşılması hedeflenir.

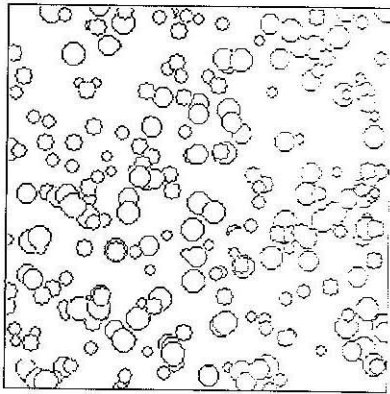
MYSA'nın, geleneksel yapay sinir ağı metotlarından çok daha yüksek sınıflandırma yeteneğine sahip olmasından dolayı, bu ve benzer alanlarda yapılacak olan çalışmaların daha fazla ilgi göreceğine inanıyoruz.



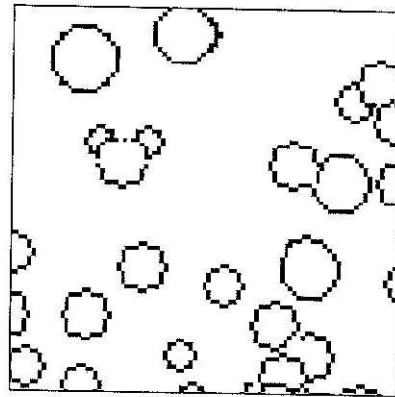
(a)



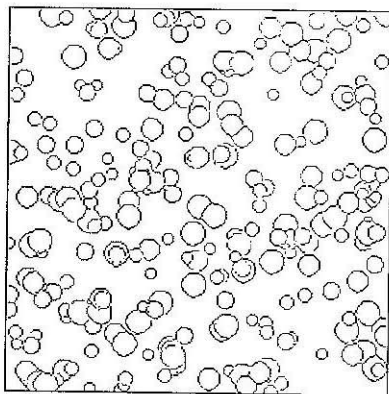
(b)



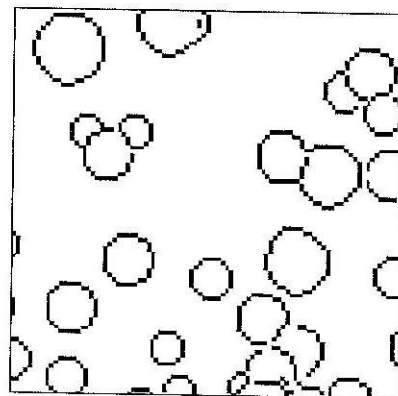
(c)



(d)



(e)

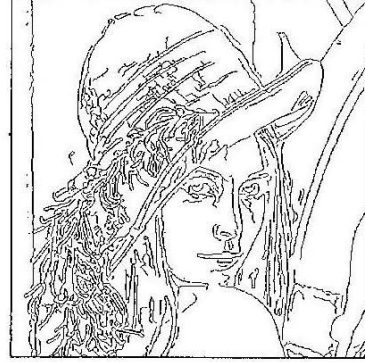


(f)

Şekil-8: (a) Arzu edilen kenar görüntüsü (b) Arzu edilen kenar görüntüsünün belli bir kısmının büyütüş hali (c) Orijinal görüntü için kullanılan MYSA'nın kenar sonucu (d) MYSA sonucunun görüntüsünün belli bir kısmının büyütüş hali (e) Farklı $t_1=20$ ve $t_2=25$ eşik değerleri için Canny kenar çıkarma algoritması kullanılarak elde edilen kenar sonucu (filtre parametresi $\alpha=1$) (f) Canny kenar çıkarma algoritmasının sonucunun belli bir kısmının büyütüş hali ($\alpha=1$, $t_1=20$, $t_2=25$).



Şekil-9: Orijinal "Lena" görüntüsü için kullanılan MYSA'nın kenar sonucu



Şekil-10: Orijinal "Lena" görüntüsü için Canny algoritmasının kenar sonucu

Kaynakça

- [1] Canny, J.F., 1986. A Computational Approach to Edge Detection, IEEE Transaction on Pattern Analysis and Machine Intelligence, vol.PAMI-8, no.6, pp. 679-698.
- [2] Bağcıoğulları, F., ve Bağcıoğulları, C.H., 1994. Nesne tanıma sistemi için geliştirilmiş özel bir kenar çıkarma algoritması, 2. Signal İşleme ve Uygulamaları Sempozyumu Bildiriler Kitabı, s. 47-52.
- [3] Abdou, I. E., and Pratt, W. K., 1979. Quantitative design and Evaluation of enhancement/thresholding high. Edge detectors, Proc. IEEE, vol. 67, no. 5, pp. 753-763.
- [4] Costa, L. F., and Cesar, R. M., 2001. Shape Analysis and Classification. pp. 240-244.
- [5] Türkoğlu, İ., ve Arslan, A., 1999. Nesne Tanıma İçin Geliştirilmiş Bir Kenar Çıkarma Yöntemi. International Conference On Electrical And Electronics Engineering, pp. 428-430.
- [6] Pinho, J., and Almeida, L.M., 1994. Some Results On Edge Enhancement With Neural Networks. Proc. of the 1st IEEE Int. Conf. on Image Processing, ICIP'94, Austin, TX, vol. III, pp. 893-897.
- [7] Perry, S., and Guan, L., 2000. Weight assignment for adaptive image restoration by neural Networks. IEEE Transactions on Neural Networks, vol. 11, pp. 156-170.
- [8] Shen, L., and Rangayyan, M. M., 1997. A segmentation-based lossless image coding method for high-resolution medical image compression. IEEE Trans. Medical Imaging, vol. 16, no. 3, pp. 301-307.
- [9] Caelli, T., Squire, D., Wild, T., 1992. Model-Based Neural Network, vol. 6, pp. 613-625.
- [10] Wong, H., Caelli, T., and Guan, L., 2000. A Model-Based Neural Network for Edge Characterization. Pattern Recognition, 33, 3, pp. 427-444.