TMMOB Harita ve Kadastro Mühendisleri Odası
10. Türkiye Harita Bilimsel ve Teknik Kurultayı
28 Mart - 1 Nisan 2005, Ankara

YAPAY SİNİR AĞLARININ RASTER GÖRÜNTÜLERİN VEKTÖREL İFADESİNDE KULLANIMI

H. Karabörk[1]. B.Koçer[2]. İ.Ö.Bildirici[1]. F.Yıldız[1]

[1]Selçuk Üniversitesi.Jeodezi ve Fotogrametri Mühendisliği Bölümü.Konya.karabork@selcuk.edu.tr.bildirici@selcuk.edu.tr. fyildiz@selcuk.edu.tr

[2]Selçuk Üniversitesi. Bilgisayar Mühendisliği Bölümü. Konya. **bariskocer@selcuk.edu.tr**

ÖZET

*Bu çalışmada, vektörizasyonda kullanılan "8'li zincir kodu" yöntemine alternatif olarak bir yöntem geliştirilmiştir. Geliştirilen yöntem, bulunduğu piksel matrisin merkezinde kalmak üzere her adımda 5*5 lik bir piksel matrisinin değerini yapay sinir ağı sistemine giriş olarak kullanmaktadır. Bu sistem çıkış olarak algoritmanın kenar izleyebilmesi için gerekli parametreleri sağlamaktadır. Ayrıca geliştirilen algoritma, kırılmalarla köşeleri ayırt etmesi sayesinde vektörizasyon problemini çözerken kenarların bulunmasını ve gereksiz bilgi fazlalığının önlenmesini sağlamaktadır. Algoritmayı denemek amacıyla bir yazılım geliştirilmiştir. Ayrıca değişik geometrik şekillerden oluşan bir resim üzerinde vektörizasyon sonucu oluşan alan ve çevre değişimleri incelenmiştir.*

Anahtar Sözcükler: Vektörizasyon, yapay sinir ağı, kenar çıkarma.

ABSTRACT

APPLICATION OF ARTIFICIAL NEURAL NETWORKS FOR VECTORIZATION

In this study, an alternative method to 8-chain code method used in vectorization has been introduced. The new method uses a matrix sized 5x5 pixel, being the actual pixel in the center of the matrix, as an input to the artificial neural network. The outputof the network provides the necessary parameters, with which the algorithm tracks edges. Since the algortihm distinguishes edges and peaks, it finds edges during vectorization, and eliminates the unnecessary data. A program has been developed to thest the new method. Besides, A test image is created from at drawing that includes several geometrical figures. This image is vectorized, and the changes in area and perimeter are determined.

Keywords: Vectorization. neural network. edge detection.

1. GİRİŞ

Raster görüntülerin vektörel ifadesi, grafik tanıma probleminin önemli bir parçasıdır. Bu amaçla birçok vektörizasyon algoritması geliştirilmesine rağmen hala işlemin doğruluk ve değişmezlik problemleri vardır. Vektöre dönüştürülecek objelerin farklılıklarından dolayı her disiplin için ayrı bir vektörizasyon algoritması geliştirilmesi gerekliliği açıktır. Harita sektöründe bilgisayar teknolojisinin kullanımının artmasından dolayı mevcut analog haritaların sayısallaştırılmasına ihtiyaç duyulmaktadır. Bu durumda vektörizasyon çok zaman alan elle sayısallaştırmaya bir alternatiftir. Haritaların sabit bir ölçeğe sahip olması, belirli büyüklükleri ve konumları olan objeleri içermesinden dolayı vektörizasyonun doğruluğu Harita Mühendisliğinde çok önemlidir.
Son 25 yıl içinde, birçok vektörizasyon metodu geliştirilmiştir. Hough dönüşüm tabanlı, inceltme tabanlı, şekil tabanlı, grafik yürütme tabanlı ve ayrık piksel tabanlı metodlar çok kullanılan metotlar olarak sayılabilir. Hough dönüşüm tabanlı metodun en basit versiyonu çizgileri tarar fakat daha karmaşık şekilleri çıkarmak için geliştirilebilir (Song et al, 2002a). İnceltme tabanlı metodlarda, piksel genişliği tek piksel olana kadar şekli oluşturan pikseller silinmektedir (Song et al 2002b). Şekil tabanlı metodlarda, ilk önce şekiller izlenir ve çizgileri tanımak için şekillerin benzerleri taranır. Merkez eksenler, bu şekil çiftleri arasından oluşturulmaktadır (Tombre ve Tabbone, 2000). Grafik yürütme tabanlı metodlar, run lenght encoding hesaplamak için ya satır ya da sütunlarda raster görüntüleri araştırmaktadır. Yürütmeler, grafik yapıları oluşturmak için analiz edilmektedir (Song et al, 2002a). Ayrık piksel tabanlı metod Dori tarafından geliştirilen Orthogonal Zig-Zag (OZZ) yönteminin geliştirilmiş halidir. Yöntemde, tüm pikselllere dokunmadan Zig-Zag çizerek şeklin vektörizasyonu yapılmaktadır (Liu ve Dori, 1999).
Bu çalışmada, raster görüntülerdeki şekillerin iskeletini çıkartmak amacıyla bir kenar çıkarma metodu geliştirilmiş ve iskeletin vektörizasyonu için yapay sinir ağı kullanılmıştır.

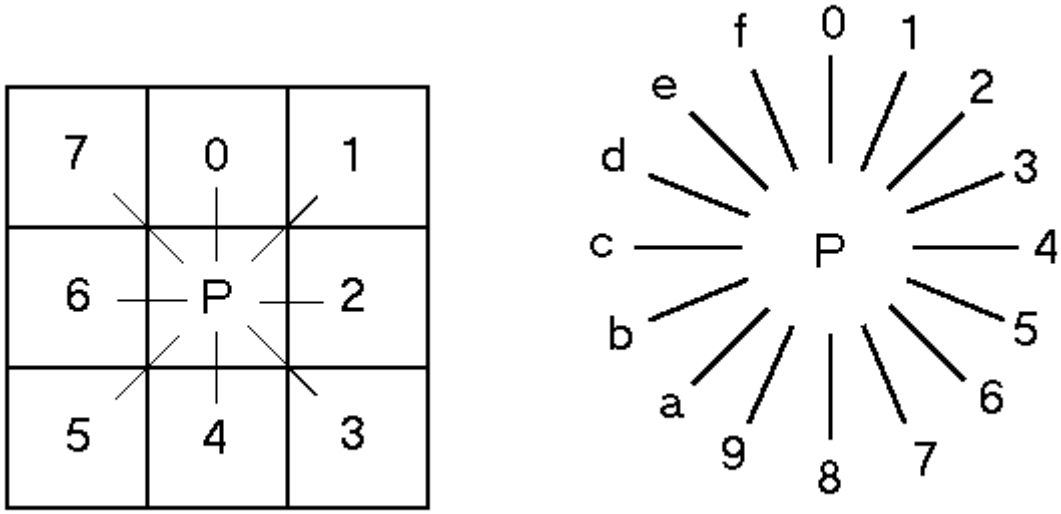
2. MATERYAL VE METOD

2.1 Kullanılan Kenar Bulma Metodu

Bu çalışmada geliştirilen metot, tek adımda hesaplama sağlamakta ve tek işlemede iskelet şeklini elde etmektedir. Tek adımda şeklin iskeletinin bulunması algoritmaya hız kazandırmaktadır. Geliştirilen yöntemde, resme ilk önce bir 3x3'lük bir ortanca filtre uygulanır. Çünkü resimdeki gürültü pikselleri yanlış çizgiler oluşmasına neden olabilir. Ortanca filtre uygulandıktan sonra şekil soldan sağa, satır satır taranır ve bulunan ilk siyah piksel temel alınarak, bu pikselin sağında, solunda, altında ve üstünde olmak üzere dört ana yönde, siyah olmayan piksel aranmaktadır. Bulunan ilk siyah olmayan piksele kadar olan siyah piksel sayıları bir dizide tutulur ve sıfırdan büyük, en küçük piksel sayısına sahip olan yönde inceltme işleminin yapılmasına karar verilir. İnceltme işlemi yapılacak yöndeki tüm siyah olmayan pikseller silinmek üzere işaretlenir ve bulunan uzunluğun tam ortasında kalan piksel ise işaretlenmeyerek siyah bırakılır. Diğer adımlarda ise işaretlenen pikseller işlenmez ve bu sayede bir hız kazancı sağlanır.

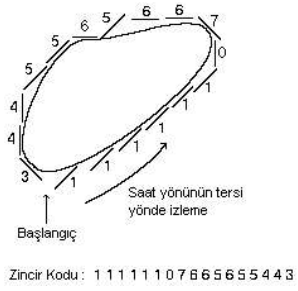
2.2 Kullanılan Yapay Sinir Ağı Modeli

Bir şekil, sıralı olarak yönlerle ifade edilen bir biçimde gösterilebilmektedir.. Yani bir şekil, tamamen yön değerlerinden oluşan bir sayısal dizi ile ifade edilebilmektedir. Şekli, bu yöntemle ifade etme metoduna zincir kodu denilmektedir.



Şekil 1: 8'li ve 16'lı zincir koduna göre komşu piksellerin yön değerleri

Eğer kulladığınız zincir kodu 8 yön içeriyorsa, bu zincir koduna "8'li zincir kodu"; eğer kullandığınız zincir kodu 16 yön içeriyorsa, bu zincir koduna "16'lı zincir kodu" denilmektedir. Bundan başka daha az ayrıntı veren 4'lü zincir kodu da mevcuttur. (**URL 1;2)**

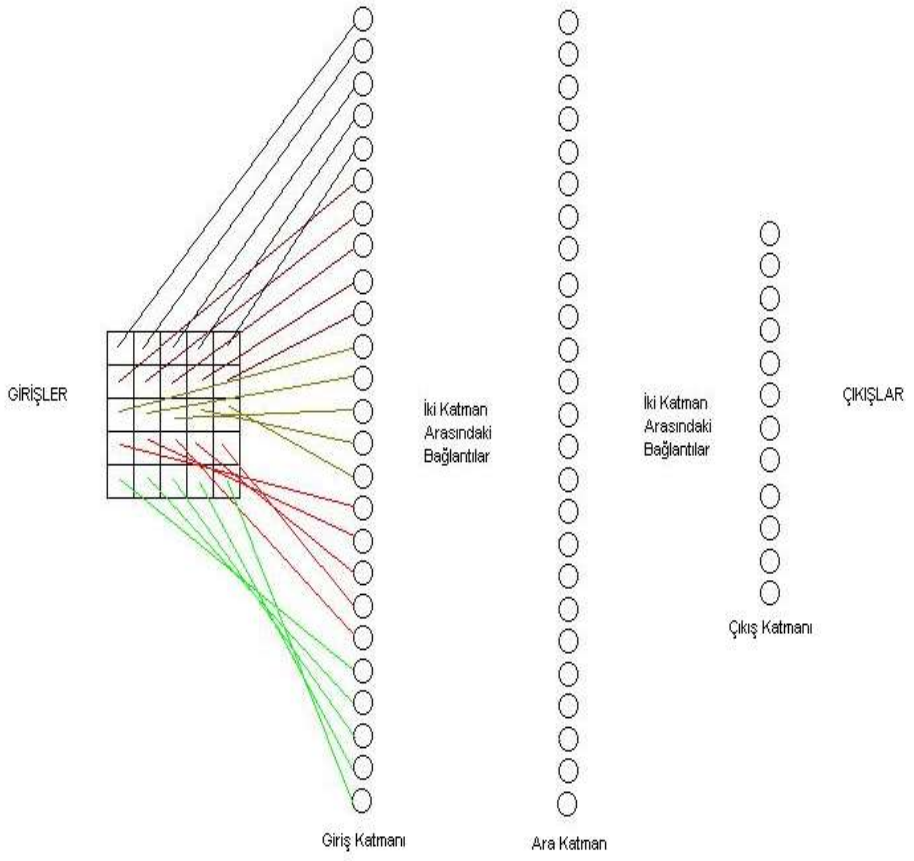


Şekil 2: Bir şeklin 8'li zincir koduna göre kodlanmış hali.

Geliştirilen uygulamada yapay sinir ağı, kenar bulma algoritmasıyla elde edilen kenarların izlenmesi ve bu kenarlardan vektörel bir şekil elde edilmesi için kullanılmıştır. Bu yöntemde, çizgi beşe beşlik bir matrisle izlenmiştir. İzlenilen çizgideki pikseller her defasında beşe beşlik matrisin ortasında kalacak şekilde yapay sinir ağına verilmiştir. Başka bir deyişle bir piksel ilerlemek için yapay sinir ağımıza 25 elemanlı bir giriş uygulamak gerekmektedir. Sinir ağımızın çıkışında ise sekizli zincir koduna göre hangi yönde gidildiği, x ve y yönlerinde kaç piksel ne yönde gidileceği gibi bilgileri almak üzere 13 adet çıkış bulunmaktadır. Bu çıkışlardan sekiz tanesi sekizli zincir koduna göre neyin hangi yönde hareket edeceğini, iki tanesi x yönünde artım mı yoksa azalış mı olacağını, iki

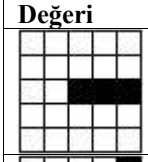
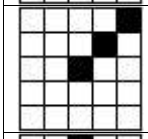
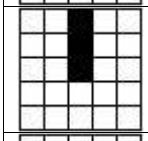
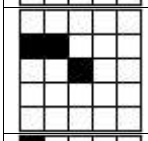
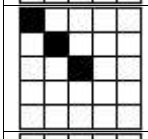
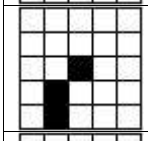
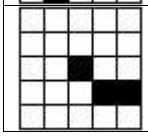
tanesi de y yönünde artım mı yoksa azalış mı olacağını, bir tanesi de bulunan noktanın kırılma olup olmadığını belirtmektedir. Yani giriş katmanı 25 ve çıkış katmanı 13 olan bir yapay sinir ağı kullanılmıştır. Sinir ağımızın doğrusal olmaması için bir ara katman geliştirilmiş ve bu ara katman da 25 elemanlı olarak tasarlanmıştır. Sinir ağımız geri beslemeli ve danışmanlı öğrenme (Supervised Learning) özelliklerine sahiptir. Yapay sinir ağının buradaki görevi kenar izleme ve kenar izlerken köşeleri belirlemektir. Yapay sinir ağı kenar izlemek için giriş olarak 5*5'lik bir matris kullanılmaktadır. Bu amaçla tüm resim sırayla satır satır taranır. Bulunan ilk kenar pikselden başlayarak bu çizgi izlenmeye başlanır. O an üzerinde bulunduğu ve izlediği piksel merkez olmak koşuluyla etrafındaki 25 piksel yapay sinir ağına verilir ve üzerinde bulunduğu yani merkezindeki pikseli silerek bu çizginin bir daha işlem görmesi engellenir. Yapay sinir ağı ise aldığı bu 25 girişle 8'li zincir koduna göre hangi yönde gideceğine karar verir ve sıradaki pikselin nerede olduğunu bulur. Bu sırada bir önceki pikselin yönünden yararlanılarak çizginin bir köşesinin olup olmadığı anlaşılır. Yani eğer çizginin yönü 8'li zincir koduna göre değişmişse buranın bir köşe olduğu kabul edilir. Şekilde bulunan her köşe, köşe tablosuna bu şekil için kaydedilmektedir. Her şekil başlangıç noktasına geri dönüldüğünde veya yapay sinir ağı bir sonraki noktayı belirleyemediği zaman bitmiş kabul edilmektedir. Şekil bittikten sonra istediğimiz kriterlere uyup uymadığının kontrol edilmesi gerekmektedir. Bu amaçla, şeklin en büyük genişliğini ve en büyük yüksekliğini çarparak piksel olarak alanı hesaplanır. Eğer alan belirttiğimiz değerin altında ise bu şekle ait tüm köşeler köşe tablosundan silinir.Eğer köşe izlenirken yapay sinir ağı hangi yöne gidileceğine karar veremezse hemen şekli bitirmek yerine köşenin devam etmesine çalışılmaktadır. Bunun için, o an üzerinde bulunulan pikselin 10*10 matris içinde kalacak şekilde merkezden dışa doğru çizgi pikseli aranmaktadır. Eğer bu komşuluk içinde de herhangi bir çizgi pikseli bulunmadığı takdirde şekil bitmiş sayılmaktadır. Her şekle ait köşeler köşe tablosuna eklendikten sonra, son adım olarak her şekil ayrı ayrı vektörel olarak ifade edilmekte ve vektör formatındaki veriler dxf formatında kaydedilmektedir. Dxf formatı yaygın kullanılan bir veri değişim formatı olduğundan tercih edilmiştir.

Şekil 3'de kullanılan yapay sinir ağı ifade edilmiştir.



Şekil 3: Kullanılan yapay sinir ağı modeli

Bu ağ, tablo.1'deki örnek değerlere göre eğitilmiştir.

Giriş Değeri	**1. Çıkış**	**2. Çıkış**	**3. Çıkış**	**4. Çıkış**	**5. Çıkış**	**6. Çıkış**	**7. Çıkış**	**8. Çıkış**	**9. Çıkış**	**10. Çıkış**	**11. Çıkış**	**12. Çıkış**	**13. Çıkış**
	1	0	0	0	0	0	0	0	1	0	0	0	0
	0	1	0	0	0	0	0	0	1	0	0	1	0
	0	0	1	0	0	0	0	0	0	0	0	1	0
	0	0	0	0	1	0	0	0	0	1	0	1	1
	0	0	0	1	0	0	0	0	0	1	0	1	0
	0	0	0	0	0	0	1	0	0	1	1	0	1
	1	0	0	0	0	0	0	0	1	0	0	1	1

Tablo 1: Ağın eğitildiği örnek değerler

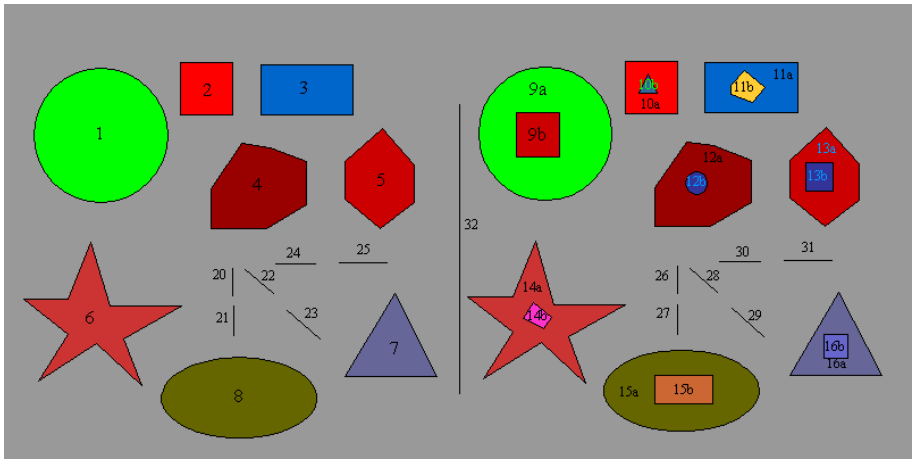
Tablo.1 'deki örnek değerlerdeki siyah görülen yerler, izlenen pikselin komşuları içinde çizgi pikseli olarak ifade edilen ve izlenecek yolu belirten piksellerdir. Her piksel izlenmesi tamamlandıktan sonra o an üzerinde bulunulan piksel bir defa işlenmesi gerektiği için silinir.

2.3 Uygulamada Kullanılan Yazılım

Yapay sinir ağı kullanılarak raster verilerden vektörel verileri elde etmek amacıyla C# programlama dilinde bir yazılım geliştirilmiştir. Bu dilin tercih edilmesindeki amaç, nesne yönelimli bir dil olmasıdır. Dilin nesne yönelimli olması sayesinde çok esnek programlama yapılabilmiş ve uygulama yüksek bir verimlikle çalıştırılmıştır. Bu yazılım VecNET olarak isimlendirilmiştir.

3. UYGULAMA

Algoritmaların doğruluğunu kontrol etmek için Macromedia Flash 5 programında hemen hemen tüm geometrik şekilleri kapsayan bir test görüntüsü oluşturulmuştur (Şekil.6).



Şekil 6: Macromedia Flash 5 programında hazırlanan test alanı

Bu test alanında, geometrik şekiller konum itibariyle değişik yerlere dağıtılmıştır. Ayrıca bazı şekillerin iç içe girmesi sağlanmıştır. Macromedia Flash 5 programında oluşturulan orijinal vektör veriden faydalanılarak kapalı geometrik şekillerin alanları ve çevresi, doğruların ise uzunlukları hesaplanmıştır. Vektöre dönüşüm sonucu oluşan geometrik şekillerin alanları ve çevresi hesaplanmıştır. Ayrıca bu değerlerin orijinal değerlerden farklılıkları yüzdelik olarak hesaplanmıştır (Tablo.2).

Obje Adı	**Orijinal Değerler**		**Vektöre Dönüşüm Değerleri**		**Alan ve Çevre Değ. Değerleri**			
	Alan	Çevre	Alan	Çevre	Al. Deg.	%Deg.	Çevre Değ.	% Değ.
1	10880.35	369.76	10591	368.88	289.35	0.03	0.88	0.00
2	2143.69	185.2	2116	184	27.69	0.01	1.20	0.01
3	3586.00	251	3564	250	22	0.01	1.00	0.00
4	5022.13	270.49	4884	268.69	138.125	0.03	1.80	0.01
5	3679.94	229.97	3461.5	224.42	218.435	0.06	5.55	0.02
6	5045.87	529.01	5094	497.48	-48.135	-0.01	31.53	0.06
7	2952.45	247.87	2885.5	245.12	66.95	0.02	2.75	0.01
8	7655.49	335.08	7481	334.88	174.49	0.02	0.20	0.00
9a	10880.35	369.76	10418.5	367.26	461.849	0.04	2.50	0.01
9b	1505.44	155.2	1482	154	23.44	0.02	1.20	0.01
10a	2143.69	185.2	2116	184	27.69	0.01	1.20	0.01
10b	145.32	55.27	137	53.58	8.32	0.06	1.69	0.03
11a	3586.00	251	3608	252	-22	-0.01	-1.00	0.00
11b	484.51	85.88	456.5	83.17	28.01	0.06	2.71	0.03
12a	5019.27	270.44	4862.5	267.89	156.765	0.03	2.55	0.01
12b	304.81	61.89	293	63.72	11.81	0.04	-1.83	-0.03
13a	3679.70	229.96	3419	222.76	260.7	0.07	7.20	0.03
13b	590.49	97.2	600	98	-9.51	-0.02	-0.80	-0.01
14a	5033.20	528.57	4978.5	490.63	54.7	0.01	37.94	0.07
14b	284.09	68.8	288	69.75	-3.91	-0.01	-0.95	-0.01
15a	7655.49	335.08	7519	335.33	136.49	0.02	-0.25	0.00
15b	1277.62	151.4	1275	152	2.62	0.00	-0.60	0.00
16a	2948.40	247.69	2900	245.29	48.4	0.02	2.40	0.01
16b	408.04	80.8	441	84	-32.96	-0.08	-3.20	-0.04

Tablo 2: Test alanına ait resmin yapay sinir ağları ile vektöre dönüştürülmesi sonucu elde edilen kapalı şekillere ait veriler

Buna ek olarak test alanı manuel olarak sayısallaştırılmıştır. Elde edilen bu vektörel veriden geometrik şekillerin alanları ve çevresi hesaplanmıştır. Ayrıca bu değerlerin orijinal değerlerden farklılıkları yüzdelik olarak hesaplanmıştır (Tablo.3).

Obje Adı	**Orijinal Değerler**		**Vektörel Değişimler**		**Alan ve Çevre Değ. Değişimler**			
	Alan	Çevre	Alan	Çevre	Al. Deg.	% Deg.	Çevre Değ.	% Değ.
1	10880.35	369.76	10798.8	369.26	81.53	0.01	0.50	0.00
2	2143.69	185.2	2103.54	183.46	40.15	0.02	1.74	0.01
3	3586.00	251	3494.44	247.56	91.56	0.03	3.44	0.01
4	5022.13	270.49	5004.11	269.9	18.015	0.00	0.59	0.00
5	3679.94	229.97	3652.39	229.31	27.545	0.01	0.66	0.00
6	5045.87	529.01	4962.56	528.08	83.305	0.02	0.93	0.00
7	2952.45	247.87	2915.25	246.35	37.2	0.01	1.52	0.01
8	7655.49	335.08	7617.67	335.23	37.82	0.00	-0.15	0.00
9a	10880.35	369.76	10713.2	367.91	167.189	0.02	1.85	0.01
9b	1505.44	155.2	1480.44	153.91	25	0.02	1.29	0.01
10a	2143.69	185.2	2083.04	182.56	60.65	0.03	2.64	0.01
10b	145.32	55.27	141.9	54.52	3.42	0.02	0.75	0.01
11a	3586.00	251	3521.92	249.18	64.08	0.02	1.82	0.01
11b	484.51	85.88	492.253	86.55	-7.743	-0.02	-0.67	-0.01
12a	5019.27	270.44	4961.5	268.95	57.765	0.01	1.49	0.01
12b	304.81	61.89	285.49	60.48	19.32	0.06	1.41	0.02
13a	3679.70	229.96	3665.01	229.24	14.69	0.00	0.72	0.00
13b	590.49	97.2	587.68	96.77	2.81	0.00	0.43	0.00
14a	5033.20	528.57	4980.28	525.2	52.92	0.01	3.37	0.01
14b	284.09	68.8	276.07	67.64	8.02	0.03	1.16	0.02
15a	7655.49	335.08	7603.32	334.43	52.17	0.01	0.65	0.00
15b	1277.62	151.4	1262.87	151.07	14.75	0.01	0.33	0.00
16a	2948.40	247.69	2933.52	247.06	14.88	0.01	0.63	0.00
16b	408.04	80.8	437.03	83.63	-28.99	-0.07	-2.83	-0.04

Tablo 3: Test alanına ait resmin manuel olarak vektöre dönüştürülmesi ile elde edilen kapalı şekillere ait veriler

Ayrıca her iki resimde elde edilen uzunluklara ait hesaplamalar yapılmıştır (Tablo.4).

Obje Adı	**Orijinal Değerler**	**Vektörel Değerler (YSA)**	**Vektörel Değerler (Manuel)**	**Vektörel Değerler (YSA)**		**Vektörel Değerler (Manuel)**	
	Uzunluk	Uzunluk	Uzunluk	Uz. Deg.	%Degisim	Uz. Deg.	%Degisim
20	24.90	24.00	24.70	0.90	0.04	0.20	0.01
21	26.60	26.00	26.87	0.60	0.02	-0.27	-0.01
22	34.00	32.65	33.65	1.35	0.04	0.35	0.01
23	39.39	38.29	38.94	1.10	0.03	0.45	0.01
24	36.50	35.00	35.56	1.50	0.04	0.94	0.03
25	42.80	42.00	42.48	0.80	0.02	0.32	0.01
26	24.90	24.00	24.66	0.90	0.04	0.24	0.01
27	26.60	26.00	26.55	0.60	0.02	0.05	0.00
28	34.00	31.89	32.68	2.11	0.06	1.32	0.04
29	39.40	37.56	37.82	1.84	0.05	1.58	0.04
30	36.50	36.50	36.67	0.00	0.00	-0.17	0.00
31	42.80	42.80	43.10	0.00	0.00	-0.30	-0.01
32	255.60	254.00	253.69	1.60	0.01	1.91	0.01

Tablo 4: Yapay sinir ağı ve manuel olarak elde edilen uzunluklara ait veriler

Şekillerin konumlarında bir kayma olup olmadığını belirlemek için tüm şekillerin MBR (Minumum Bounding Rectangle: Şekli içine alan en küçük dikdörtgen) merkezleri hesaplanmış ve x ile y yönündeki kayma miktarları belirlenmiştir (Tablo.5).

ObjeAdı	**Ağırlık Merkezi(MBR)**						**x ve y yönündeki farklar**			
	Orijinal		**Yapay sinir ağı**		**Manuel**		**Yapay sinir ağı**		**Manuel**	
	x (piksel)	y (piksel)	x (piksel)	y (piksel)	x (piksel)	y (piksel)	Dx	Dy	Dx	Dy
1	84.8	115	85	115	84.71	115.15	-0.2	0.0	0.09	-0.15
2	178.1	74	178	74	177.82	73.73	0.1	0.0	0.28	0.27
3	266.3	74.5	266.5	75	266.31	74.97	-0.2	0.5	-0.01	-0.47
4	223.8	159.3	224	159.5	224.13	158.84	-0.2	0.2	-0.33	0.46
5	330.8	152.9	330.5	151.5	330.53	152.72	0.3	1.4	0.27	0.18
6	86.2	272.1	84.5	270.5	86.81	271.78	1.7	1.6	-0.61	0.32
7	340.6	290.9	340	290.5	339.8	290.13	0.6	0.4	0.8	0.77
8	206.7	345.6	206.5	345.5	206.14	344.95	0.2	0.1	0.56	0.65
9a	476.4	113.6	476.5	113.5	476.35	112.93	-0.1	0.1	0.05	0.67
9b	469.9	114.2	470	114.5	469.99	113.61	-0.1	0.3	-0.09	0.59
10a	569.6	72.6	570	73	569.56	72.46	-0.4	0.4	0.04	0.14
10b	567	69.3	567	69.5	566.49	68.94	0	0.2	0.51	0.36
11a	657.9	73.2	658	73	658.07	72.81	-0.1	0.2	-0.17	0.39
11b	655.3	72	655	71.5	655.22	71.93	0.3	0.5	0.08	0.07
12a	615.4	157.9	615	158	615.14	157.83	0.4	0.1	0.26	0.07
12b	609.5	157.3	609.5	157	609.38	157.07	0	0.3	0.12	0.23
13a	722.4	151.6	722.5	150	723.92	151.7	-0.1	1.6	-1.52	-0.1
13b	717.7	151.5	718	151.5	717.98	151.57	-0.3	0.0	-0.28	-0.07
14a	477.7	270.7	475.5	268	477.42	270.54	2.2	2.7	0.28	0.16
14b	469.3	273.3	469	273.5	469.73	273.75	0.3	0.2	-0.43	-0.45
15a	596.3	339.4	596.5	339.5	596.16	340.23	-0.2	0.1	0.14	-0.83
15b	598.4	338.3	598.5	338.5	598.29	339.03	-0.1	0.2	0.11	-0.73
16a	732.1	289.4	732.5	289.5	734.06	288.73	-0.4	0.1	-1.96	0.67
16b	732.5	300.4	732.5	300.5	734.27	299.67	0.0	0.1	-1.77	0.73
20	201.3	243.2	201	244	201.05	243.5	0.3	0.8	0.25	-0.3
21	201.8	277.6	202	278	202.49	277.78	-0.2	0.4	-0.69	-0.18

22	223.8	243.2	224.5	243.5	224.52	243.01	-0.7	0.3	-0.72	0.19
23	262.9	280.8	263.5	281.5	263.73	281.43	-0.6	0.7	-0.83	-0.63
24	255.9	227.8	256.5	228	256.33	227.6	-0.6	0.2	-0.43	0.2
25	315.8	226.7	316	227	315.5	226.41	-0.2	0.3	0.3	0.29
26	592.8	241.8	593	242	593.07	241.99	-0.2	0.2	-0.27	-0.19
27	593.4	276.2	593	277	593.29	276.4	0.4	0.8	0.11	-0.2
28	615.4	241.8	616	242.5	616.12	242.31	-0.6	0.7	-0.72	-0.51
29	654.4	279.4	655	279.5	656.76	279.08	-0.6	0.1	-2.36	0.32
30	647.5	226.4	648	226	647.86	225.95	-0.5	0.4	-0.36	0.45
31	707.3	225.3	708	225	709.4	224.94	-0.7	0.3	-2.1	0.36
32	400.8	214.6	401	215	401.08	215.14	-0.2	0.4	-0.28	-0.54

Tablo 5: Test Alanına ait resmin manuel ve yapay sinir ağları ile vektöre dönüştürülmesi sonucu şekillere ait elde edilen MBR merkezleri

4. SONUÇ

Geliştirilen yöntem, yaygın kullanılan "8'li zincir kodu" yöntemine alternatif olarak sunulmaktadır. Sabit olarak sadece bir sonraki piksel değerini kontrol ederek ilerleyen ve kenarları bu yönteme göre bulan yöntemin aksine, geliştirilen yöntem yapay sinir ağlarını kullanarak yönünü bulmakta ve bu esnada komşu piksellerin yanı sıra bu piksellerden sonra gelen pikselleri de kullanmaktadır. Ayrıca, geliştirilen yöntem tolerans değeri içinde kopmuş bir çizgiyi devam ettirebilmektedir. Bu özelliğin toleransa bağlı olmasının sebebi, birbirinden farklı olan şekillerin de birleştirilip bir şekil olarak oluşturulmasının önüne geçmektir.

Test sonucu, geliştirilen metodla elde edilen şekillerin çevresinde %0-6, alanlarında %1-8 ve uzunluklarda %0-6 değişim görülmektedir. Manuel yöntemde, çevrelerde %0-4, alanlarda %0-4 ve uzunluklarda %0-4 değişim görülmektedir. Konum sapmaları ise x yönünde; geliştirilen yazılımda 0-2.2 piksel, manuel olarak 0-2.36 piksel, y yönünde; geliştirilen yazılımda 0-2.7 piksel , manuel yöntemde 0-0.77 piksel olarak elde edilmiştir. Ayrıca, tüm konum sapmaları içinde, x ve y yönünde 1 pikselden büyük olan değer sayısı; geliştirilen yazılımda 6, manuel yöntemde 5'tir.

Elde edilen bu değerler, yaklaşık aynı kabul edilebilir. Geliştirilen yöntem, bu alanda başarı ile uygulanabilir gibi görünmesine rağmen, daha da geliştirilmeye açıktır.

KAYNAKLAR

Liu W.. and Dori D.. 1999, *From Raster to Vectors: Extracting Visual Information From Line Drawings*, Pattern Analysis & Applications (1999) 2, 10-21

Song J.. Cai M.. Lyu M R.. and Cai S.. 2002a, *Graphics Recognition From Binary Images: One Step or Two Step*, 16 th International Conference on Pattern Recognition (ICPR'02) 3, Quebec City, QC, Canada

Song J.. Su F.. Tai C L.. and Cai S., 2002c, *An Object-Oriented Progressive-Simplification-Based Vectorization System For Engineering Drawings: Model, Algorithm, and Performance*, IEEE Transactions on Pattern Analysis and Machine Intelligence 24(8), 1048-1060

Tombre K. and Tabbone S.. 2000. *Vectorization in Graphics Recognition: To Thin or not to Thin*, International Conference on Pattern Recognition (ICPR'00)- 2, Barcelona, Spain

URL 1: http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/MARBLE/medium/contours/chain.htm. 28.02.2005

URL 2: http://rover.idi.ntnu.no/~cv/ZACC.shtml.28.02.2005