

T.C.
HALIÇ ÜNİVERSİTESİ
FEN BİLİMLER ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ

ARAÇ PLAKASI TANINMASI

YÜKSEK LİSANS TEZİ

Hazırlayan
MELİH ÇİÇEK

Tez Danışmanı
Prof. Dr. ALİ OKATAN

MAYIS 2008
İSTANBUL

T.C.
HALIÇ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ MÜDÜRLÜĞÜNE

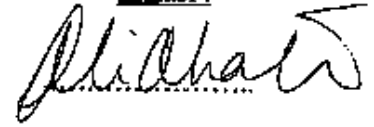
Bilgisayar Mühendisliği Programı Yüksek Lisans öğrencisi
Melih ÇİÇEK tarafından hazırlanan “ Araç Plakası Tanınması ” adlı bu
çalışma jürimizce Yüksek Lisans Tezi olarak kabul edilmiştir.

Tez Sınav Tarihi : 27.06.2008

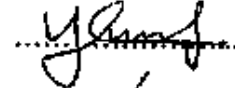
(Jüri Üyesinin Ünvanı , Adı , Soyadı ve Kurumu) :

İmzası :

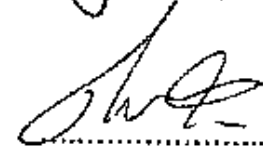
Jüri Üyesi: Prof.Dr.Ali OKATAN
(Danışman-HÜ.Bilgisayar Mth.ABD Öğr.Üyesi)



Jüri Üyesi : Yrd.Doç Dr.Yüksel BAL
(H.Ü. Bilgisayar Mth.ABD Öğr.Üyesi)



Jüri Üyesi :Yrd.Doç.Dr.Murat BEKEN
(H.Ü.Uygulamalı Matematik ABD Öğr.Üyesi)



İÇİNDEKİLER

	<u>Sayfa No.</u>
ÖNSÖZ.....	IV
ABSTRACT.....	V
ÖZET.....	VII
SUMMARY.....	VIII
1.GİRİŞ.....	1
2.MATERYAL VE YÖNTEM.....	2
2.1. Genel Algoritma Yapısı.....	3
3.GÖRÜNTÜ İŞLEME.....	4
3.1. Sayısal Resmin Matematiksel İfadesi.....	5
3.2. Histogram.....	7
3.3. Görüntü Ön İşlemede Kullanılan Algoritmalar.....	8
3.3.1. Filtrelerin Uygulanışı.....	8
3.3.1.1. Medyan Filtre.....	9
3.3.1.2. Sobel Kenar Sensörü.....	10
3.3.1.3. Canny Kenar Sensörü.....	11
3.3.1.4. Threshold.....	12
3.3.1.5. Inver.....	12
4.YAPISAL ELELEMANLAR.....	13
4.1. Morjoloji Yöntemleri.....	13
4.1.1. Genişletmek.....	13
4.1.2. Aşındırmak.....	14
4.1.3. Açmak.....	15

4.1.4. Kapatmak.....	15
5.SMEARING ALGORİTMASI.....	17
6.KARAKTER AYRIŞTIRMA.....	19
7.YAPAY SİNİR AĞLARI.....	20
7.1. Yapay Zeka.....	20
7.2. Biyolojik Nöron.....	21
7.2.1. Aksonlar.....	22
7.2.2. Dentritler.....	22
7.2.3. Hücre Gövdesi.....	22
7.2.4. Sinapsisler.....	23
7.3.Yapay Sinir Ağlarına Yaklaşım.....	23
7.3.1. Yapay Sinir Ağlarının Kullanım Alanları.....	24
7.4. Yapay Sinir Ağlarına Giriş.....	25
7.4.1. Aktivasyon Fonksiyonları.....	26
7.5. Yapay Sinir Ağı Mimarileri.....	28
7.5.1. İleri Beslemeli Yapay Sinir Ağları.....	28
7.5.2. Geri Beslemeli Yapay Sinir Ağları.....	28
7.6. Yapay Sinir Ağlarında Öğrenme.....	29
7.6.1. Denetimli Öğrenme.....	29
7.6.2. Denetimsiz Öğrenme.....	29
7.7. Geriye Yayılım Algoritması İle Öğrenme.....	30
8.YAPILAN UYGULAMALAR	34
8.1. Görüntü Ön İşlemeleri.....	34
8.2. Yapay Sinir Ağıının Eğitilmesi.....	36

9. SONUÇ.....	37
10. KAYNAKLAR.....	38
11. ÖZGEÇMİŞ.....	39

ÖNSÖZ

Yüksek lisans tez konumun belirlenmesinde ve çalışmalarımda bana yardımlarını esirgemeyen, değerli katkılarından dolayı danışman hocam sayın Prof. Dr. Ali OKATAN hocama teşekkürlerimi sunarım. Ayrıca tez çalışmam sırasında desteklerinden dolayı çalışma arkadaşlarım olan Yazılım Uzmanları Orkun ŞANLI ve Ömer AKBAŞ'a teşekkür ederim

ŞEKİLLER LİSTESİ

	<u>Sayfa No.</u>
Şekil – 1.1. Genel Algoritma Yapısı.....	3
Şekil – 2.1. Spektrum.....	4
Şekil – 2.2. Orijinal Plaka Resmi.....	5
Şekil – 2.3. İkilik Plaka Resmi.....	5
Şekil – 2.4. Resim Matrisi.....	5
Şekil – 2.5. RGB Kombinasyonu Çiçek Resmi.....	6
Şekil – 2.6. Red Kombinasyonu Çiçek Resmi.....	7
Şekil – 2.7. Red-Blue Kobinasyonu Çiçek Resmi.....	7
Şekil – 3.1. Histogram Öncesi Grafik.....	8
Şekil – 3.2. Histogram Sonrası Grafik.....	8
Şekil – 4.1. Araç Plaka Orijinal Resim.....	9
Şekil – 4.2. Araç Plaka Filtrelenmiş.....	9
Şekil – 5.1. Araç Plakası Sobel Filtresi.....	10
Şekil – 6.1. Araç Plakası Canny Filtresi.....	11
Şekil – 6.2. Araç Plakası Canny Filtresi 2.....	11
Şekil – 7.1. Örnek Gri Düzey.....	12
Şekil – 7.2. Threshold.....	12
Şekil – 8.1. Örnek Gri Düzey.....	12
Şekil – 8.2. İvert Edilmiş.....	12
Şekil – 9.1. Yapısal Elemanlar.....	13
Şekil – 9.2. Yapısal Elemanlar 2.....	13
Şekil – 10.1. Genişletilmiş Örnek.....	14

Şekil – 10.2. Aşındırılmış Örnek.....	15
Şekil – 10.3. Açılmış Örnek.....	15
Şekil – 10.4. Kapatılmış Örnek.....	16
Şekil – 11.1. Örnek Araç Resmi.....	17
Şekil – 11.2. Ön İşlenmiş Araç Resmi.....	17
Şekil – 11.3. Smearing Algoritması Araç Resmi.....	17
Şekil – 11.4. Plaka Resmi.....	18
Şekil – 12.1. Biyolojik Nöron.....	21
Şekil – 13.1. Yapay Sinir Ağ Modeli.....	25
Şekil – 13.1.1. Aktivasyon Fonksiyonu.....	26
Şekil – 13.1.2. Aktivasyon Fonksiyonu 2.....	27
Şekil – 13.1.3. Aktivasyon Fonksiyonu 3.....	27
Şekil – 14.1. Yapay Sinir Ağ Mimarisi.....	28
Şekil – 14.2. Yapay Sinir Ağ Mimarisi 2.....	28
Şekil – 15.1. Geriye Yayılım Algoritması.....	30
Şekil – 16.1. Uygulama.....	34
Şekil – 16.2. Uygulama 2.....	35
Şekil – 16.3. Örnek Plaka.....	35
Şekil – 16.4. İşlenmiş Örnek Plaka.....	35
Şekil – 16.5. Filtrelenmiş Örnek Plaka.....	36

ÖZET

Günümüzdeki teknolojik gelişmelerin ışığında büyüyen trafik problemi, güvenlik ve kontrol ihtiyaçları gibi problemlere çözüm olabilecek, insansız kontrol birimler tasarlamak ana hedeftir. Trafik denetleme, gişe otomasyonu, otopark giriş kontrolü, askeri sahalarda araç kontrolü gibi uygulama alanlarında verimli olarak kullanılabilir.

Bu çalışma Türk plaka standartlarına uyan sivil araçlar için geliştirilmiş bir uygulamasıdır. Bu çalışmada plaka tanıma için yapay sinir ağları kullanılmıştır. Tanıma işlemi 3 aşamada gerçekleşmektedir. Bunlar araç görüntüsünün ön işleme, plaka alanının bulunması ve plakadaki karakterlerin tanınmasıdır.

ABSTRACT

This project is developed to solve security and control problem in the growing traffic problem with technology in the near future. Main target is that develop without human unit. This Project can be used to check traffic, ticket Office automation, car park and military areas.

This Project is developed for Turkish license plate standart that fit civiltian car. We use neuron network for license plate recognition. Recognition realize in 3 steps that preprocessing car image, find license plate region and recogizing characters in license plate.

T.C.
HALIÇ ÜNİVERSİTESİ
FEN BİLİMLER ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ

PLAKA TANIMA YAZILIMI

YÜKSEK LİSANS TEZİ

Hazırlayan
MELİH ÇİÇEK

Tez Danışmanı
Prof. Dr. ALİ OKATAN

MAYIS 2008
İSTANBUL

İÇİNDEKİLER

	<u>Sayfa No.</u>
ÖNSÖZ.....	IV
ABSTRACT.....	V
ÖZET.....	VII
SUMMARY.....	VIII
1.GİRİŞ.....	1
2.MATERYAL VE YÖNTEM.....	2
2.1. Genel Algoritma Yapısı.....	3
3.GÖRÜNTÜ İŞLEME.....	4
3.1. Sayısal Resmin Matematiksel İfadesi.....	5
3.2. Histogram.....	7
3.3. Görüntü Ön İşlemede Kullanılan Algoritmalar.....	8
3.3.1. Filtrelerin Uygulanışı.....	8
3.3.1.1. Medyan Filtre.....	9
3.3.1.2. Sobel Kenar Sensörü.....	10
3.3.1.3. Canny Kenar Sensörü.....	11
3.3.1.4. Threshold.....	12
3.3.1.5. Inver.....	12
4.YAPISAL ELELEMANLAR.....	13
4.1. Morjoloji Yöntemleri.....	13
4.1.1. Genişletmek.....	13
4.1.2. Aşındırmak.....	14
4.1.3. Açmak.....	15

4.1.4. Kapatmak.....	15
5.SMEARING ALGORİTMASI.....	17
6.KARAKTER AYRIŞTIRMA.....	19
7.YAPAY SİNİR AĞLARI.....	20
7.1. Yapay Zeka.....	20
7.2. Biyolojik Nöron.....	21
7.2.1. Aksonlar.....	22
7.2.2. Dentritler.....	22
7.2.3. Hücre Gövdesi.....	22
7.2.4. Sinapsisler.....	23
7.3.Yapay Sinir Ağlarına Yaklaşım.....	23
7.3.1. Yapay Sinir Ağlarının Kullanım Alanları.....	24
7.4. Yapay Sinir Ağlarına Giriş.....	25
7.4.1. Aktivasyon Fonksiyonları.....	26
7.5. Yapay Sinir Ağı Mimarileri.....	28
7.5.1. İleri Beslemeli Yapay Sinir Ağları.....	28
7.5.2. Geri Beslemeli Yapay Sinir Ağları.....	28
7.6. Yapay Sinir Ağlarında Öğrenme.....	29
7.6.1. Denetimli Öğrenme.....	29
7.6.2. Denetimsiz Öğrenme.....	29
7.7. Geriye Yayılım Algoritması İle Öğrenme.....	30
8.YAPILAN UYGULAMALAR	34
8.1. Görüntü Ön İşlemeleri.....	34
8.2. Yapay Sinir Ağıının Eğitilmesi.....	36

9. SONUÇ.....	37
10. KAYNAKLAR.....	38
11. ÖZGEÇMİŞ.....	39

ÖNSÖZ

Yüksek lisans tez konumun belirlenmesinde ve çalışmalarımda bana yardımlarını esirgemeyen, değerli katkılarından dolayı danışman hocam sayın Prof. Dr. Ali OKATAN hocama teşekkürlerimi sunarım. Ayrıca tez çalışmam sırasında desteklerinden dolayı çalışma arkadaşlarım olan Yazılım Uzmanları Orkun ŞANLI ve Ömer AKBAŞ'a teşekkür ederim

ŞEKİLLER LİSTESİ

	<u>Sayfa No.</u>
Şekil – 1.1. Genel Algoritma Yapısı.....	3
Şekil – 2.1. Spektrum.....	4
Şekil – 2.2. Orijinal Plaka Resmi.....	5
Şekil – 2.3. İkilik Plaka Resmi.....	5
Şekil – 2.4. Resim Matrisi.....	5
Şekil – 2.5. RGB Kombinasyonu Çiçek Resmi.....	6
Şekil – 2.6. Red Kombinasyonu Çiçek Resmi.....	7
Şekil – 2.7. Red-Blue Kobinasyonu Çiçek Resmi.....	7
Şekil – 3.1. Histogram Öncesi Grafik.....	8
Şekil – 3.2. Histogram Sonrası Grafik.....	8
Şekil – 4.1. Araç Plaka Orijinal Resim.....	9
Şekil – 4.2. Araç Plaka Filtrelenmiş.....	9
Şekil – 5.1. Araç Plakası Sobel Filtresi.....	10
Şekil – 6.1. Araç Plakası Canny Filtresi.....	11
Şekil – 6.2. Araç Plakası Canny Filtresi 2.....	11
Şekil – 7.1. Örnek Gri Düzey.....	12
Şekil – 7.2. Threshold.....	12
Şekil – 8.1. Örnek Gri Düzey.....	12
Şekil – 8.2. İvert Edilmiş.....	12
Şekil – 9.1. Yapısal Elemanlar.....	13
Şekil – 9.2. Yapısal Elemanlar 2.....	13
Şekil – 10.1. Genişletilmiş Örnek.....	14

Şekil – 10.2. Aşındırılmış Örnek.....	15
Şekil – 10.3. Açılmış Örnek.....	15
Şekil – 10.4. Kapatılmış Örnek.....	16
Şekil – 11.1. Örnek Araç Resmi.....	17
Şekil – 11.2. Ön İşlenmiş Araç Resmi.....	17
Şekil – 11.3. Smearing Algoritması Araç Resmi.....	17
Şekil – 11.4. Plaka Resmi.....	18
Şekil – 12.1. Biyolojik Nöron.....	21
Şekil – 13.1. Yapay Sinir Ağ Modeli.....	25
Şekil – 13.1.1. Aktivasyon Fonksiyonu.....	26
Şekil – 13.1.2. Aktivasyon Fonksiyonu 2.....	27
Şekil – 13.1.3. Aktivasyon Fonksiyonu 3.....	27
Şekil – 14.1. Yapay Sinir Ağ Mimarisi.....	28
Şekil – 14.2. Yapay Sinir Ağ Mimarisi 2.....	28
Şekil – 15.1. Geriye Yayılım Algoritması.....	30
Şekil – 16.1. Uygulama.....	34
Şekil – 16.2. Uygulama 2.....	35
Şekil – 16.3. Örnek Plaka.....	35
Şekil – 16.4. İşlenmiş Örnek Plaka.....	35
Şekil – 16.5. Filtrelenmiş Örnek Plaka.....	36

ÖZET

Günümüzdeki teknolojik gelişmelerin ışığında büyüyen trafik problemi, güvenlik ve kontrol ihtiyaçları gibi problemlere çözüm olabilecek, insansız kontrol birimler tasarlamak ana hedeftir. Trafik denetleme, gişe otomasyonu, otopark giriş kontrolü, askeri sahalarda araç kontrolü gibi uygulama alanlarında verimli olarak kullanılabilir.

Bu çalışma Türk plaka standartlarına uyan sivil araçlar için geliştirilmiş bir uygulamasıdır. Bu çalışmada plaka tanıma için yapay sinir ağları kullanılmıştır. Tanıma işlemi 3 aşamada gerçekleşmektedir. Bunlar araç görüntüsünün ön işleme, plaka alanının bulunması ve plakadaki karakterlerin tanınmasıdır.

ABSTRACT

This project is developed to solve security and control problem in the growing traffic problem with technology in the near future. Main target is that develop without human unit. This Project can be used to check traffic, ticket Office automation, car park and military areas.

This Project is developed for Turkish license plate standart that fit civiltian car. We use neuron network for license plate recognition. Recognition realize in 3 steps that preprocessing car image, find license plate region and recogizing characters in license plate.

1. GİRİŞ

Son yıllarda araç sayısının artması ile trafiğin daha güvenli bir şekilde yürümesi için denetimli geçişlerin yapıldığı askeri üs, paralı otoyol, köprü gişeleri, otoparklar... vb. gibi yerlerde aracın konumu belirlemek, yaptığı yasal olmayan davranışları gözlemlemek için araç tanıma sistemlerine ihtiyaç duyulmaktadır. Araç tanıma işleminin plaka kullanılarak yapılmasının temel amacı her aracın farklı plakaya sahip olmasından kaynaklanır. Yani plaka araba özelliklerinde birincil anahtarıdır.

Araç Tanıma Sistemine yönelik çalışmalar çeşitli ülkelerde devam etmektedir. Radyo frekanslarını kullanan gömülü elektronik sistemler, sistem kapsamına giren araçları algılamak için yolun altına koyulan sensörler ve bunun gibi birçok neden araç maliyeti oldukça yükseltmektedir. Maliyetleri milyon dolarları kapsayan bir projeler hayata geçirilmiştir. Bu programlar araçları takip, elektronik ceza kesme, plaka tanıma ve araç tanımayı içermektedir. Bununla güvenliği, verimi ve konforu arttırmak, ekonomik çözümler üretmek, kirliliği azaltmak amaçlanmaktadır.

Araç tanıma amacıyla tasarlanan sistemlerin yüksek ticari değere sahip olmaları, kullanım alanlarını genişletmeyi engellemiştir. Oysa bilgisayar tabanlı sistemler tamamen gömülü elektronik donanıma sahip olan sistemlerden daha etkin bir çözüm sunmaktadır. Donanımsal olarak araçların izlenecek olan bölgeye sadece bir kamera konması yeterli olacaktır.

Plaka tanıma sistemi genel olarak 3 ana bölümden oluşmaktadır. Bunlar;

*Resim Ön İşleme

*Plaka Yerinin Saptanması

*Plakadaki karakterlerin tanınması

Yapılan çalışmada plaka yerinin saptanması ve plaka karakterlerinin tanınip metne dönüştürülmesi amaçlanmıştır. Plaka yerinin saptanması için görüntü işleme tekniklerinden yararlanılmıştır. Karakterlerin tanınması için de yapay sinir ağlarından yararlanılmıştır. Yapay Sinir Ağının' nın kullanılmasının en önemli nedenlerinden biride öğrenme için esnek bir yapıya sahip olması ve tanıma işleminin çok hızlı gerçekleşmesinden kaynaklanmaktadır.

2. MATERYAL VE YÖNTEM

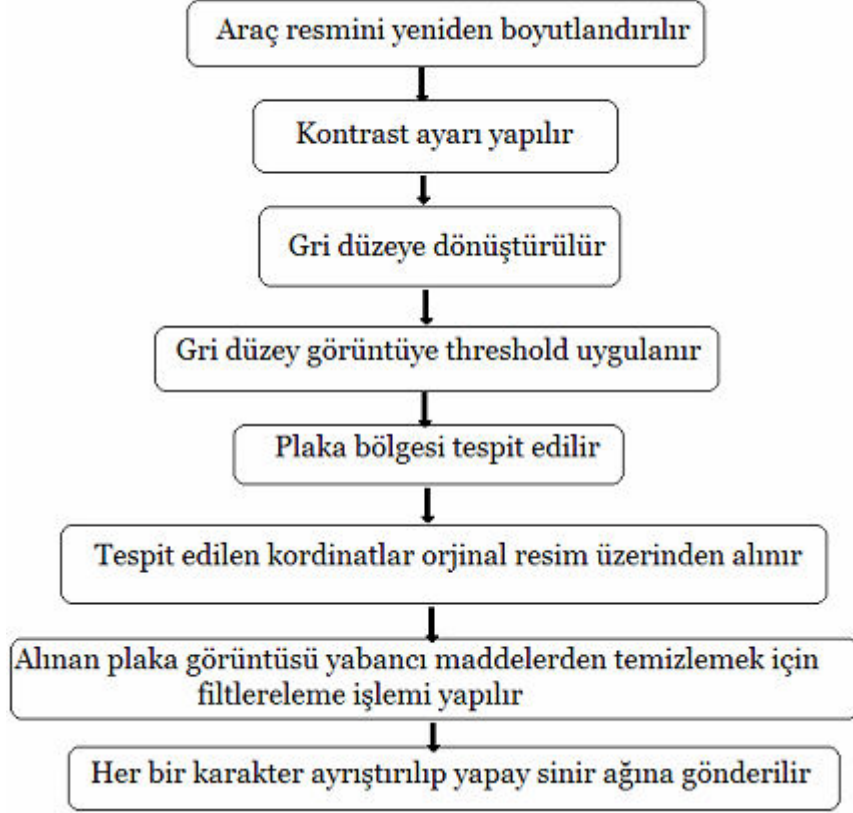
Bu bölümde plaka tanıma sisteminin oluşturulması sırasında izlenen yollar anlatılmaktadır. Birinci kısımda sistemin inşa edilmesi için gerekli donanımsal ihtiyaçlar diğer kısımda ise sistemin yapısından bahsedilmektedir.

Plaka tanıma sistemi için geliştirilen program intel Pentium M işlemci 1.86 Ghz işlemci, 512 MB RAM ve Intel Mobil 915 gm 128 MB ekran kartına sahip bilgisayar kullanılmıştır. Microsoft Windows XP Professional işletim sistemi üzerinde Microsoft Visual C#.Net kullanılarak geliştirilen program yapay sinir ağı eğitimi ve görüntü işleme algoritmaları kullanılarak geliştirilmiştir.

Programın kullandığı resimler 5.1 Megapixel fotoğraf makinesiyle çekilmiştir. Fotoğraflar araca 2–3 metre uzaklıktan çekilmiştir. Çekilen fotoğraflar orijinalde 2304 x 1728 çözünürlüğüne sahip olmasına rağmen resim üzerinde yapılacak olan ön işlemenin fazla uzun sürmemesi için resim boyut ayarlama fonksiyonu ile resize yapılmış ve 692 x 519 boyutuna düşürülmüştür. Orijinal resimde daha keskin çözümler elde etmek mümkündür fakat resim üzerinde yapılacak işlemler sırasında piksel sayısının fazlalığı yapılacak işlemlerin yavaşlamasına sebep olacaktır.

2.1. Genel Algoritma Yapısı

Tespit edilen araç resminin yapay sinir ağına gönderilmesine kadar geçen zaman süresince başından geçenler aşağıda gösterilmiştir



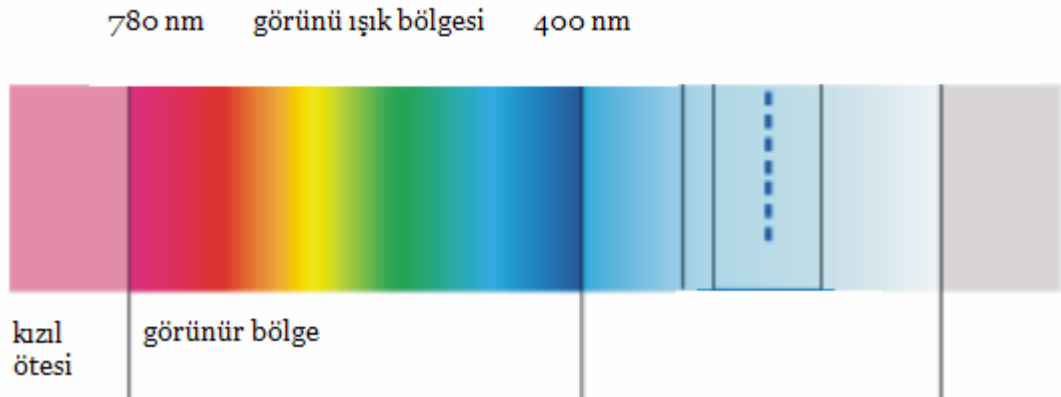
Şekil-1.1 Genel Algoritma Yapısı

3. GÖRÜNTÜ İŞLEME

Bilgisayar teknolojisinin son yıllarda hızla gelişmesi elektronik verilerin daha hızlı işlenmesine olanak sağlamaktadır. Görüntü işleme teknolojisi de buna paralel olarak gelişmiş ve birçok alanda kullanılmaya başlanmıştır. Görüntü işleme teknolojisi görüntü yakalama, gruplama ve analiz konusunda bilinen en karmaşık sistemdir.

İnsandaki görme sistemi gözlerimizle başlar. Işığın çok kanallı ve pankromatik dalga boyları her biri birer algılama sistemi olan gözlerimiz yardımı ile algılanır. Görülebilir spektrum tanımı; insan gözünün görebileceği elektro manyetik dalga boyu aralığını tanımlar. Buna karşın bir arının görebildiği spektral aralık ultraviyole bölgede başlar ve yeşil dalga boylarında sona erer.

Spektrum uzunluk ölçme birimleri ile ölçülebilen periyodik davranış sergileyen enerji dalgalarını temsil eder. Görülebilir alana ait dalga boyları $0.4\mu\text{m}$ - $0.7\mu\text{m}$ marasındadır.



Şekil-2.1. Spektrum

Gözlerimizle görülebilir alandaki ışığı algılayabiliriz ve beynimiz yardımı ile yorumlanabilir görüntü haline dönüştürebiliriz.

Gözlerin farklı kontrastlara adapte olabilme yeteneği parlaklık adaptasyonu (brightness adaption) olarak adlandırılır. İki parlaklık düzeyleri arasında ayırım yapabilme yeteneğine ise kontrast duyarlılığı adı verilir. Bu da gözün etrafını çevreleyen parlaklık düzeylerine bağlıdır. Güneşli bir günde farları yanan bir aracın farlarını görmek güçtür, fakat gece değildir.

İnsan da görüntü işleyen bir varlıktır aslında. Göz fotoğraf makinesi gibi çalışır. Görünen bölgedeki elektromanyetik ışık dalgaları göz yardımıyla algılanır ve içeri gelen ışık, elektrik sinyallerine dönüştürülerek beyine iletilir. İşte beyin de bizim görüntüyü analiz eden işlemcimizdir.

Resimlerin bilgisayar ortamında işlenebilmesi için bilgisayarın anlayabileceği formata yani sayısal bilgiye dönüştürülmesi gereklidir. Resimlerin sayısala dönüştürülmesi çeşitli şekillerde olanaklıdır. Mesela buna tarayıcılarda örnek olabilirler. Ama sonuçta hepsinin temelinde analog / sayısal dönüştürücüler vardır.

Sayısal bir resim deyince akla analog bir sinyalin sayısal bir sinyale dönüştürülmesi gelmelidir. Bu da obje tarafından yayılan enerjinin ki bu da bir analog sinyaldir, bir algılayıcı tarafından öngörülen elektromanyetik dalga aralığında algılanarak sayısal sinyal haline dönüştürülmesi ile olanaklıdır.

3.1. Sayısal Resmin Matematiksel İfadesi

Bir resmin sayısallaştırılmasının açıklanmasında öncelikle Siyah-Beyaz resim incelenebilir. Resim iki boyutlu bir matristen oluşmaktadır. Bu matris $f(x,y)$ şeklinde ifade edilebilir. Bu fonksiyonda x ve y matrisin koordinatlarıdır. Siyah-Beyaz resim sadece iki gri değerden oluşan bir resimdir. Böylesi bir görüntüde her bir piksel ya siyah ya da beyaz olarak oluşur. Biz sembolik olarak beyaz pikselleri 1, siyah pikseller 0 olarak değerlendirebiliriz. Bu şekilde 0 ve 1 kodlanmış piksellerden oluşan görüntülere ikilik görüntü (binary image) adı verilir. Şekil-2.3'te ikilik formata çevrilmiş resim görülmektedir.



Şekil-2.2. Orijinal Plaka Resmi



Şekil-2.3. İkilik Plaka Resmi

0	1	2	...	M
1	$x-1,y-1$	$x,y-1$	$x+1,y-1$	
2	$x-1,y$	x,y	$x+1,y$	
...	$x-1,y+1$	$x,y+1$	$x+1,y+1$	
N				

Şekil-2.4. Resim Matrisi

Şekil – 2.4’te $M \times N$ boyutlu matris gözükmemektedir. Bu x,y konumundaki piksellerin görüntüsü ve komşulukları görülmektedir.

Gri seviyeli resimdeki fonksiyonun değeri bu koordinatlardaki resmin gri-seviye değerini belirtir. Bu fonksiyondaki x değeri resim satırını, y değeri ise sütununu gösterir. Teknik olarak bu matrisin yani resmin her bir elemanı, bir pikselin değerini bize verir. Her bir elemanın değeri 0’dan başlayıp 255’e kadar farklı değerler alabilir. Genelde parlak beyaz 255 olarak alınır ve koyu siyahta 0 değerini alır diğer tonlar bu kod değerleri arasında bir kodla ifade edilir. Renklerin kod karşılıkları bu şekilde olabileceği gibi tam terside olabilir.

Gri değer aralıkları: $G=\{0,1,2,\dots,255\}$ şeklinde ifade edilir. Bunun anlamı şudur: Bir gri tonlu görüntüde 256 tane farklı gri ton değeri daha doğrusu gri değer bulunabilir. Burada 256 gri değer bir byte olarak tanımlanabilir. (1 Byte=8 Bit ve $2^8=256$). 0 gri değeri kural olarak siyah renge, 255 gri değeri ise beyaza karşılık gelir. Bu değerler arasında ise gri tonlar oluşur.

Renkli görüntüler bilgisayar ekranlarında 24 bitlik veri olarak görüntülenebilir. Görüntüleme R(Kırmızı), G(Yeşil), B(Mavi) kodlanmış aynı objeye ait üç adet gri düzeyli görüntünün üst üste ekrana iletilmesi ile oluşur. Elektromanyetik spektrumda 0,4–0,5 μm dalga boyu mavi renge; 0,5–0,6 μm dalga boyu yeşil renge; 0,6-0,7 μm dalga boyu kırmızı renge karşılık gelir. Bu dalga boylarında elde edilmiş üç gri düzeyli görüntü bilgisayar ekranında sırası ile kırmızı-yeşil-mavi kombinasyonunda üst üste düşürülecek olursa renkli görüntü elde edilmiş olur.

Orijinal resim şekil–2.5’te görülmektedir. Bu resim kırmızı - mavi- yeşil kombinasyonundan oluşur. Resmi yeniden işleyip sadece kırmızı tonlardan oluşan şekil–2.6 elde edilmiştir.



Şekil–2.5 RGB Kombinasyonu Çiçek Resmi



Şekil-2.6 Red Kombinasyonu Çiçek Resmi

Şekil-2.6’te pikseller yeniden işlenip her pikselin yeşil ve mavi değeri sıfır olarak ayarlanmasıyla elde edilmiştir. Yani başka bir deyişle kırmızı olarak filtrelenmiştir.

Şekil-2.7’te ise kırmızı ve mavi olarak filtrelenmiş resim görülmektedir.



Şekil-2.7 Red-Blue Kobinasyonu Çiçek Resmi

3.2. Histogram

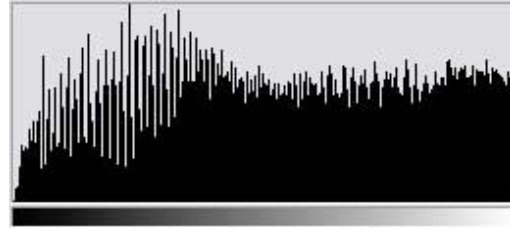
Resmimizdeki her pikselin sayısını bulup bunu bir grafiğe döktüğümüzde resmin histogramını çıkarmış oluruz. Histogram grafiğinde yatay eksen gri değer aralığını, dikey eksen de her bir aralıktaki piksel sayısı gösterilir. Böylelikle görüntünün kontrastına ilişkin bilgi elde etmek mümkün olmaktadır.

Histogram, görüntü üzerindeki piksellerin değerlerinin grafiksel ifadesidir. Buna görüntü histogramı veya gri-düzey histogramı denir. Görüntü histogramı, görüntünün her bir noktasındaki piksellerin tespiti ile bu piksellerin sayısının ne kadar olduğunu gösterir. Bu sayede histogram üzerinden görüntü ile ilgili çeşitli bilgilerin çıkartılması sağlanır.

Histogram eşitleme ise resimdeki kontrastı arttırmak için kullanılan bir yöntemdir. Amaç resimdeki piksellerin sahip olduğu değerleri bütün spektruma yaymaktır.



Şekil-3.1 Histogram Öncesi Grafik



Şekil-3.2 Histogram Sonrası Grafik

Şekil-3.1’de resmin önce histogram eşitlenmeden haldeki grafiği şekil - 3.2’de ise histogram eşitlendikten sonraki grafiği gösterilmiştir. Görüldüğü gibi renkler spektruma yayılmış piksel sayılarının değişimi aza indirilmiştir.

3.3. Görüntü Ön İşlemede Kullanılan Algoritmalar

Yapılan çalışmada plaka bölgesinin bulunması için birçok algoritma denenmiştir. Görüntüye birkaç işleme yapılarak plaka bölgesi elde edilmiştir. Plaka bölgesinin bulunması bu algoritmaların kombinasyonlarından oluşmaktadır. Plaka bölgesi elde edilmesinde farklı birkaç yol bulunmuştur.

3.3.1. Filtrelerin Uygulanışı

Filtreler resmin içerisindeki bazı ayrıntıları ortaya çıkartmak veya resim içerisindeki istenmeyen gürültülerin yok edilmesini sağlamak için kullanılmaktadır.

Özellikle; resimleri sayısal olarak kaydeden cihazların görüntüyü hatalı bir şekilde elde etmeleri ve aydınlatma gibi çevre koşullarının yetersizliğinden kaynaklanan birçok kayıp ya da pürüz görüntü işleme filtreleri kullanılarak en aza indirilebilmektedir.

Genelde filtreler 3x3 matris olarak ifade edilirler. Filtre örnekleri uygulama öncesi ve sonrası örnek resimler aşağıda gösterilmiştir.

Genelde 3x3 (piksel) matris halinde bir çekirdek görüntü üzerinde gezdirilir. Her piksel değeri gezdirilen nesnenin altında kalan değer ile çarpılır ve filtre içindeki elemanların toplamına bölünür. Örneğin resim;

1	5	7	3	34
0	2	44	3	9
34	76	23	34	3
6	55	57	12	5
0	12	54	23	3

Filtremiz ise;

1	2	1
0	-2	1
1	1	1

Bu filtrenin resmin sol üst köşesinden başlayarak resim üzerinde gezdirilir. 3 x 3 boyutundaki matrisimize filtrenin çekirdeği denilir. Filtrenin aşağıdaki matrisin üzerinde olduğu zaman yapılacak işlemler gösterilmektedir.

1 5 7

0 2 44

34 76 23

$((1*1) + (2*5) + (1*7) + (0*0) + (-2*2) + (1*44) + (1*34) + (1*76) + (1*23))$

(1 + 2 + 1 + 0 - 2 + 1 + 1 + 1 + 1 + 1)

Bu işlem resmin her pikseli için uygulanır ve sonuç güncellendiğinde resmin filtrelenmiş hali ortaya çıkar.

3.3.1.1. Medyan Filtre

Resmin keskin bölgelerini yumuşatmayı amaçlayan bir algoritmadır. Programda 3x3 matris halinde kullanılarak faydalanıldı. Bu filtreyi kullanmaktaki amaç plaka üzerindeki gürültülerin bu yöntemle yok edilmesi idi. Filtre uyguladığımız alandaki piksel değerleri alınarak, küçükten büyüğe sıralanır, ortadaki değer filtrenin ortasına yazılır.

Orijinal resim



Şekil-4.1 Araç Plaka Orijinal Resim

Filtrelenmiş Resim



Şekil-4.2 Araç Plaka Filtrelenmiş

3.3.1.2. Sobel Kenar Sensörü

Resimdeki kenarları bulmak için geliştirilmiş bir yöntemdir. Bu yöntemin çekirdek matrisleri aşağıda gösterilmiştir.

$$g_x = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix} \quad g_y = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix}$$

$$\text{Renk} = \sqrt{g_x^2 + g_y^2}$$

Yatay ve dikey olarak iki tane çekirdek vardır. Bunlardan çıkan sonuçların karelerinin toplamının karekökü rengi belirlemektedir.

Aşağıda şekil-5.1’de sobel kenar bulma algoritması uygulanmış resim görülmüyor. Resmin orijinalini şekil-4.1’de görülmektedir. Plaka bölgesinin bulunmasında sobel sensörünün önemli yeri vardır. Görüntüden de görüldüğü gibi plakanın dikdörtgen şekli ortaya çıkmıştır. Tabi bu yapılan ön işlemenin sadece bir kısmıdır.



Şekil-5.1 Araç Plakası Sobel Filtresi

3.3.1.3. Canny Kenar Sensörü

Bu filtreyi uygularken yine sobel'deki çekirdekler kullanılır. Buradaki fark resimde minimum ve maksimum threshold değerleri verilir. Canny kenar detektörünün uygulandığı resmin aşağıda görebilirsiniz.



Şekil-6.1 Araç Plakası Canny Filtresi

Bu resimde threshold minimum değeri 20 maksimum değeri ise 100'dür.



Şekil-6.2 Araç Plakası Canny Filtresi 2

Bu resimde maksimum değer 150'ye çıkartılması ile oluşturulmuştur.

3.3.1.4. Threshold

Gri düzeye çevrilmiş resimde belirli bir değerin altında ya da üstünde olan pikselleri uç noktalara çekme işlemine threshold denir. Örneğin gri düzeydeki resim için 128'i eşik seviyesi olarak belirlediğimizde bu değerden yüksek olanların değerleri 255, daha düşük olanlar ise 0 yapılır. Bu sayede resim sadece 0 ve 255 değerindeki piksellerden yani siyah ve beyazdan oluşur.



Şekil-7.1 Örnek Gri Düzey



Şekil-7.2 Threshold

Şekil-7.1 için threshold değeri 150 olarak seçildi ve resim işlendiği zaman ortaya şekil-7.2 çıkmıştır. Bu özelliğini plaka yerinin bulunması ve bulunan plakanın üzerinden karakterlerin ayırt edilmesinden önce resmi hazırlayan güzel bir yöntemdir.

3.3.1.5. İnvirt

Resmin tersini almaktır. Siyah beyaz resimlerde siyah pikseller beyaza, beyazlarda siyaha çevrilir. Gri düzey resimlerde ise piksel değeri o pikseli 255'e tamamlayan değer ile değiştiriliyor.



Şekil-8.1 Örnek Gri Düzey



Şekil-8.2 İnvirt Edilmiş

Yukarıda şekil-8.1' de resim orijinali şekil-8.2'de ise tersi alınmış hali görülmektedir. Burada örnek olarak gri düzey bir resim kullanılmıştır.

4. YAPISAL ELEMANLAR

Yapısal eleman birçok morfoloji (yapı bilimi) işleminin gerçekleştirilmesinde en önemli öğedir. Sayısal resimler matematiksel ifade oldukları için matematiksel morfoloji özellikleri kullanılır.

Aslında yapısal eleman olarak adlandırılan ifade istenilen boyutlarda ve istenilen şekilde hazırlanmış küçük binary bir resimdir. Yapısal eleman çeşitli geometrik şekillerden biri olabilir. En sık kullanılanları kare, dikdörtgen ve dairedir. Yapısal eleman örnekleri aşağıda görülmektedir

1	1	1
1	1	1
1	1	1

Şekil-9.1 Yapısal Elemanlar

0	1	0
1	1	1
0	1	0

Şekil-9.2 Yapısal Elemanlar 2

Yapısal elemanın merkez pikselinin binary bir resim içerisindeki objelere (1'lere) temas etmesiyle ortaya çıkan obje ile yapısal elemanın kesişim kümesi objeden çıkartılarak veya objeye eklenerek birçok morfolojik işlem gerçekleştirilir. Buradaki en önemli nokta objeye eklenen ya da objeden çıkarılan kısmın yapısal eleman tarafından belirlenmesidir.

Eğer morfolojik işlemin sonucunda resimdeki nesnelerin keskin hatları silinip yerlerine kavisli veya daha yumuşak hatlar getirilmek isteniyorsa dairesel yapısal eleman kullanılmalıdır. Örneğin erozyon işleminde resim içerisindeki nesnelerin en ve boyları aynı oranda azaltılmak (erozyona uğratılmak) isteniyorsa yapısal eleman kare seçilmelidir.

4.1 Morfoloji Yöntemleri

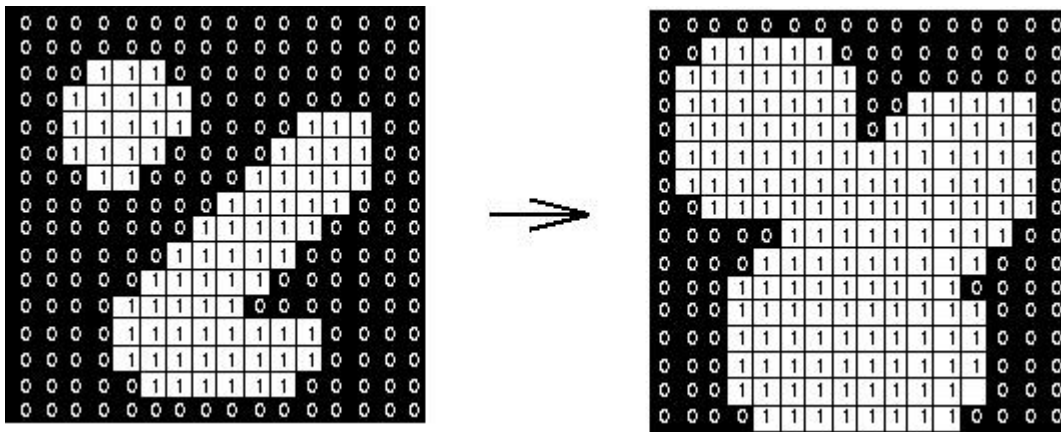
4.1.1 Genişletmek

Sayısal bir resmi genişletmek demek resmi yapısal elemanla kesiştiği bölümler kadar büyütme demektir. Bunu yapabilmek için yapısal eleman resim üzerinde piksel piksel dolaştırılır. Eğer yapısal elemanın orijini resim üzerinde 0 değerli bir piksel ile karşılaşırsa herhangi bir değişiklik meydana gelmez. Eğer değeri 1 olan bir piksel ile karşılaşırsa yapısal elemanla yapısal elemanın altında kalan

pikseller mantıksal or işlemine tabi tutulurlar. Yani herhangi 1 değeriyle sonuç 1'e çevrilir.

Genişletme (dilation) ile resim üzerindeki objeler şişer. Objeye içinde delikler var ise bunlar kapanma eğilimi gösterirler. Ayrık nesnelere birbirine yaklaşır ya da bağlanır.

Aşağıda 3x3 yapısal elemanı ile sayısal resim üzerine genişletme uygulanması ifade edilmiştir. 3x3 lük yapısal elemanın tüm değerleri 1'dir.



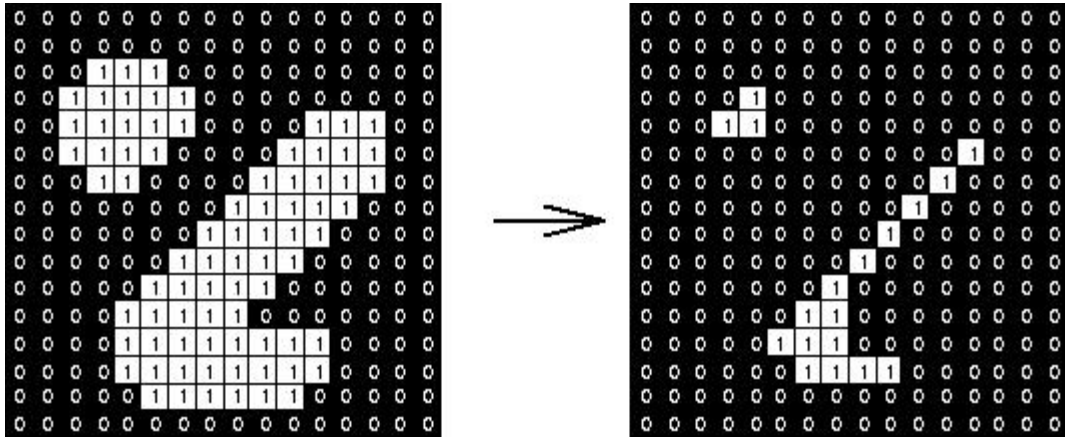
Şekil-10.1 Genişletilmiş Örnek

4.1.2. Aşındırmak

Aşındırma işlemi bir bakıma genişletmenin tersi gibi görülebilir. Burada yine aynı şekilde yapısal eleman resim üzerinde piksel piksel dolaştırılır fakat bu defa yapısal elemanın merkez pikseli 1 değeri ile karşılaşırsa yapısal eleman içerisindeki piksellerin durumuna bakılır. Eğer yapısal eleman içerisindeki 1 olan piksellerden herhangi biri altında resme ait 0 değeri varsa yapısal elemanın diğer 1'lerinin altındakilerle beraber bu piksel 0'a dönüştürülür.

Aşındırma (erozyon, erosion) işlemi ile sayısal resim aşındırılmış olur. Yani resim içerisindeki nesnelere ufalır, delik varsa genişler, bağlı nesnelere ayrılma eğilimi gösterir.

Aşağıda 3x3 yapısal elemanı ile sayısal resim üzerine aşındırma uygulanması gösterilmiştir. 3x3 lük yapısal elemanın tüm değerleri 1'dir.

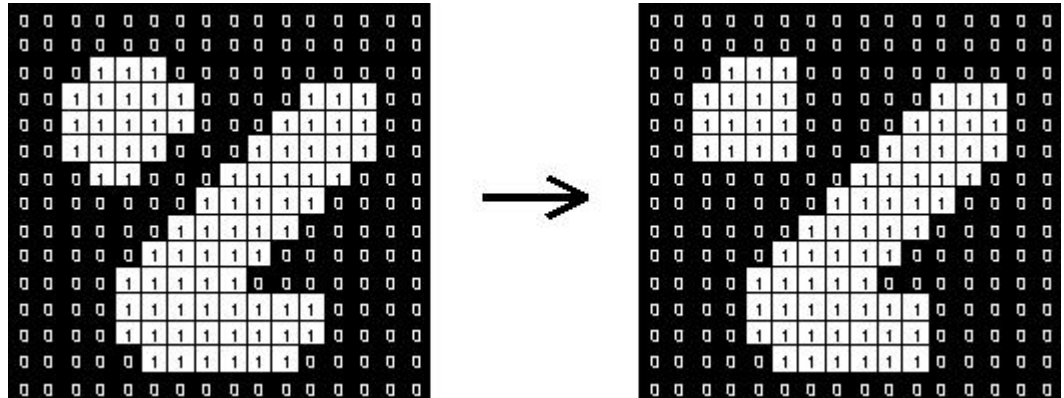


Şekil-10.2 Aşındırılmış Örnek

4.1.3. Açmak

Sayısal bir resme önce aşındırma daha sonra genişletme uygulanırsa resme Açma (Opening) işlemi uygulanmış olur.

Açma işlemine tabi tutulmuş bir görüntü ve değişimi aşağıda gösterilmiştir. Burada yine 3x3 lük yapısal eleman kullanılmıştır.

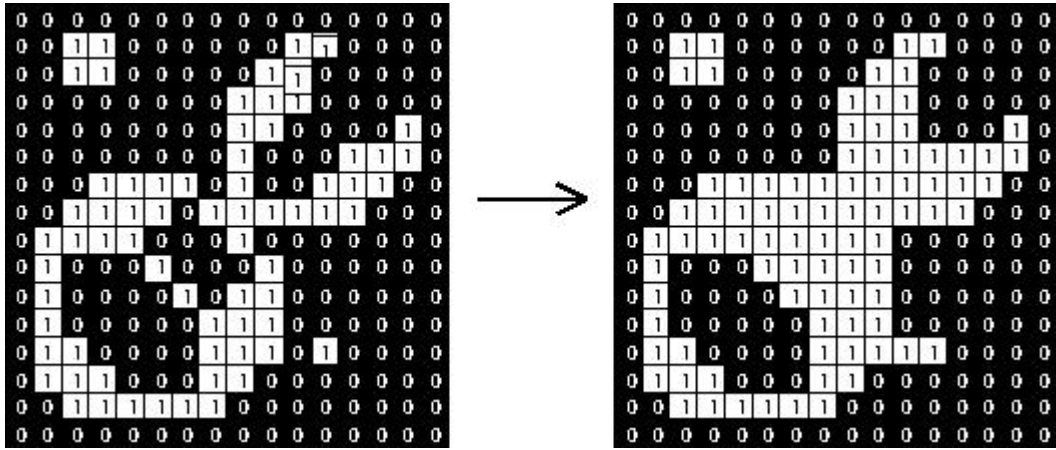


Şekil-10.3 Açılmış Örnek

4.1.4. Kapatmak

Sayısal resme önce genişletme daha sonra aşındırma uygulanırsa Kapatma (Closing) işlemi uygulanmış olur.

Aşağıda kapatma işlemi uygulanmış bir görüntünün önce ve sonrası durumları incelenmiştir.



Şekil-10.4 Kapatılmış Örnek

Açma işlemi ile birbirine yakın iki obje görüntüde fazla değişime sebebiyet vermeden ayrılmış olurlar. Kapatmada ise birbirine yakın iki nesne görüntüde fazla değişiklik yapılmadan birbirine bağlanmış olurlar.

5. SMEARING ALGORİTMASI

Smearing algoritması ikilik resme plaka bölgesinin bulunması için gerekli bir algoritmadır. Görüntünün tam karşıdan alınmış olması bu algoritmanın işlemesi açısından çok önemlidir.

Bu algortmada görüntü hem yatay hem de düşey olarak tarama yapılır. Yatayda ve düşeyde yapılan tarama sırasında beyaz piksellerin sayısı önemlidir. Eğer bu piksellerin sayısı bizim belirlediğimiz eşik seviyesinden büyük ya da küçükse pikseller siyaha çevrilir. Smearing algoritması uygulanırken plakanın boyutları konusunda deneme yaparak daha önceden bilgi sahibi olmalıyız. Böylece eşik seviyesi için fikir ediniriz. Bu algoritma yatay ve düşey eşik olarak iki tane eşik değerine sahiptir.



Şekil-11.1 Örnek Araç Resmi



Şekil-11.2 Ön İşlenmiş Araç Resmi



Şekil-11.3 Smearing Algoritması Araç Resmi

Resim üzerinde kontrast ayarı yapıp çeşitli filtrelerden (sobel,mean,,vb.) geçirdikten sonra resmimize threshold uygularız. Threshold'dan sonra içi kapalı olan dikdörtgenlerin içi doldurulur. Böylece ortaya şekil-11.1 çıkar.

```
if(150<beyaz_piksel_sayisi < 160)
```

```
    Değişiklik_yok;
```

```
Else
```

```
    Pikselleri_beyaz_yap;
```

Bu algoritma yatay pikseldeki işlemler için yapılmıştır.

Smearing algoritması şekil-11.2'ye uygulanır. Plaka genişliğine yaklaşık 10 ya da 11 karakter sığmaktadır. 692 x 519 boyutundaki bir resimde smearing algoritması uygulandığı zaman yatay eşik seviyesi 160 – 170 beyaz pikseldir. Düşey eşik değeri ise yaklaşık 60 – 70 pikseldir. Bu seviyelerin altında kalan satır ve sütunlardaki pikseller siyaha çevrilir. Böylece ortaya şekil-11.3 ortaya çıkar. Beyaz piksellerin bulunduğu alan orijinal resimden alınarak plakayı elde etmiş oluruz. Plaka şekil-11.4'te görülmektedir.



Şekil-11.4 Plaka Resmi

6. KARAKTER AYRIŞTIRMA

Plaka bölgesini elde ettikten sonra karakterleri ayırıştırmak için hazırlamamız gereklidir. Bunun için resimdeki var olabilecek çamur, logo veya vida gibi yabancı maddelerden temizlemek için resme önce threshold uygulanıp sonra morfoloji işlemlerinden geçirilir. Karakterlerin genişliğinin bulunması aralarındaki uzaklığın hesaplanması gibi işlemlerde bu modülde yapılır.

Gürültüleri yok edebilmek için, plaka bölgesine ilk olarak closing morfoloji algoritması uygulanır. Böylece vida gibi yabancı maddeler yok edilir. Ancak bazen plaka bölgesinde plakanın en başında olan Türkiye'nin tr logosu da dâhil olabilir. Bunu ve closing ile yok edilemeyen gürültüleri yok etmek için plaka bölgesinin başından itibaren karakter boyunda dikey bir çizgi alınarak kademeli olarak plakanın sonuna doğru kaydırılmaktadır.

Dikey çizginin herhangi bir objeyi kesişmesi durumunda ilgili objeyi sınırlayan zarf hesaplanarak, obje boyutları belirlenmektedir. Objeye yüksekliği ve genişliği öngörülen karakter yüksekliğine ve genişliğine uygun değil ise gürültü olarak algılanmakta ve görüntüden silinmektedir. Böylece plaka üzerindeki ilk etapta silinmeyen vida, çamur veya logo bu sefer silinmiş olur.

Bütün gürültüler ve Türkiye logosu silindikten sonra salt plaka karakterlerinden oluşan resmi elde ederiz. Daha sonra karakterleri birbirinden ayırdığı uzaklığında incelenmesi ile karakter gruplarını buluruz. İlk karakter grubu il kodu olacak ve iki karakterden oluşacaktır. Daha sonraki grup harfler son grupta rakamlardan meydana geldiğini biliyoruz. Elde edilen ilk grup rakam tanıyan yapay sinir ağına gönderilir. İkinci grup karakter tanıyan yapay sinir ağına gönderilir. Üçüncü ve son grup tekrar rakam tanıyan yapay sinir ağına gönderilir.

Gönderilme işleminde dikkat edilecek nokta yapay sinir ağına giriş düğüm sayısı sabittir. Ancak elde ettiğimiz her resimde farklı boyutlarda olabilir. Bunu standartlaştırmak için 20 x 30 piksel olacak şekilde yeniden boyutlandırılır. Böylece her karakterin boyutu aynı olacak şekilde ayarlanır.

7. YAPAY SİNİR AĞLARI

Günümüzde yaşamın vazgeçilmez bir parçası haline gelen bilgisayarlar, hem karar verebilmekte hem de olaylar arasındaki ilişkileri öğrenebilmektedir. Matematiksel olarak formülasyonu kurulamayan ve çözülmesi mümkün olmayan problemler sezgisel yöntemlerle bilgisayarlar tarafından çözülebilmektedir. Bilgisayarları bu özelliklerle donatan ve bu yeteneklerin gelişmesini sağlayan çalışmalar Yapay zekâ çalışmaları olarak bilinmektedir. Zeki sistemlerin en temel özellikleri, olaylara ve problemlere çözümler üretirken veya çalışırken bilgiye dayalı olarak karar verebilme özelliklerinin olmasıdır. Bu mekanizmaların taklit edilmesi, Yapay Sinir Ağları gibi farklı teknolojilerin doğmasına neden olmuştur. Dünyanın en karmaşık makinesi olarak kabul edebileceğimiz insan beyni, sayısal bir işlemi yapabilmek için dakikalarca uğraşmasına rağmen, idrak etmesine yönelik olayları çok kısa bir sürede yapabilmektedir. Genel olarak yapay sinir ağları da, insan beyninin sinir ağlarından esinlenerek geliştirilmiş bilgisayar programlarıdır. Bir taraftan donanım teknolojisi gelişmekte ve daha hızlı çalışabilen, daha çok bilgiyi saklayabilen, bilgisayarlar ve bilgisayar sistemleri oluşturulmakta iken diğer taraftan yazılım teknolojisi gelişmekte ve bilgi işleme yetenekleri, öğrenme, karar verebilme, problem çözme, muhakeme yapabilme yöntemleri ve bu yöntemlere dayalı yazılım sistemleri geliştirilmektedir. Teorik çalışmalar çok hızlı bir şekilde pratik hayatta uygulanmaktadır.

7.1. Yapay Zekâ

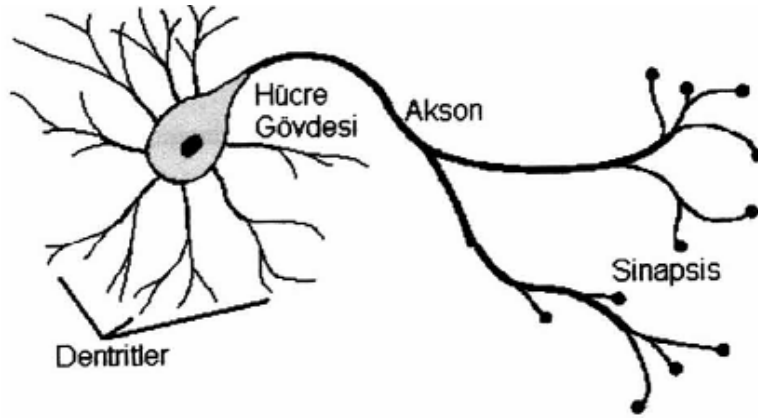
Yapay zekâ, zeka ve düşünme gerektiren işlemlerin bilgisayarlar tarafından yapılmasını sağlayacak araştırmaların yapılması ve yeni yöntemlerin geliştirilmesi hususunda çalışılan bilim dalıdır. Yapay zekâ; "düşünme, anlama, kavrama, yorumlama ve öğrenme yapılarının programlamayla taklit edilerek problemlerin çözümüne uygulanması" olarak da ifade edilebilir.

Yapay zekâ bilginin organizasyonu, öğrenme, problem çözme, teorem ispatlama, bilimsel buluşların modellenmesi gibi birçok konu ile ilgilenmektedir. Bu yetenekler ile donatılan bilgisayar sistemleri, problemlere çözüm üretirken insanın problemleri çözme sürecini taklit etmektedir. Özellikle belirli bir algoritma veya formülasyon kullanılarak çözülemeyen problemlerin çözülmesi için yapay zekâ

sistemleri geliştirilmektedir. Problemin çözümünü sağlayan bir algoritma geliştirilmiş ise geleneksel bilgisayar sistemleri problemi çözmek için yeterli olmaktadır. Önemli olan, problemin çözümünü veren bir formülün olmadığı durumlarda bilgisayarlara problemleri çözdürmektir. Yapay zekâ bu görevi üstlenmiş bir bilim dalıdır.

7.2. Biyolojik Nöron

Sinir sistemimizin temeli nöronlardır. Temel olarak vücuttaki diğer hücrelerle aynı yapıda olmasına rağmen beyindeki hesaplama ve haberleşme gibi fonksiyonları da yerine getirmektedir. Bir nöron üç ayrı bölüme oluşur ve her birinin farklı görevleri vardır. Bu bölümler akson, dentrit ve sinapsislerdir.



Şekil-12.1 Biyolojik Nöron

Dentritlerin görevi, sinapsis olarak adlandırılan temas noktalarında diğer hücrelerden sinyaller almaktır. Sinapsislerden sinyaller hücre gövdesine diğer sinyallerle birleşmek üzere gelirler. Kısa bir zaman aralığında gelen sinyallerin ortalaması yeterince büyükse, hücre harekete geçer ve kendisinden sonraki hücrelere bağlı olan aksonlarına bir çıkış darbesi üretir. Sinir hücresi diğer sinir hücresinden gelen sinyalleri dentrit üzerinden alır. Nöronun akson tepesinden çıkan akson, bazı kollara ayrılarak ilerler ve sinyallerini efektör adı verilen kas, salgı bezi ve diğer sinir hücrelerine iletir. Akson ve yan kolları, sinaptik düğüm adı verilen uçlar oluşturur ve sinapslerde diğer nöronların hücre gövdesi, dentrit ve aksonları ile bağlantı kurar.

Şekil-12.1'de tipik bir nöron hücresi verilmiştir. Şekilde de görüldüğü gibi bir nöronun binlerce dentrit çıkabilir. Bunların uzunlukları genelde 1mm'den daha kısadır. Bazıları ise birkaç milimetre uzunluğa ulaşabilir. Dentritler çevre hücrelerden gelen sinyalleri alıp gövdeye ulaştırırlar. Her nöron dentritler sayesinde diğer birçok nöronun gelen sinyalleri alıp birleştiren ve işleyen bir birimdir. Şimdi nöronları oluşturan parçaları inceleyeceğiz. (KAHRAMAN F., 2001)

7.2.1. Aksonlar

Nöronlar arasındaki iletim hattını oluşturur. Uç kısmına yakın bir yerde birçok dallara ayrılır. Bu dallar bir sinapsisle sonlanır. Sinapsislerde sinyal, diğer nöronlara dentrit üzerinden veya bazı durumlarda doğrudan hücre gövdesine aktarılır. Bu yol üzerinden bir nöron yüzlerce veya binlerce nöronu aktif hale getirebilir. Bu nöronların da her biri dentrit üzerinden diğer nöronlara bağlanarak çalışırlar. Nöron bağlantılarının fazla olması hesaplama kabiliyetini artırır. Aksonu sınırlayan sinaptik bağlantı, küçük soğan köküne benzer şekilde genişleyen, sinaptik kaparcık olarak adlandırılan küresel bir yapıdadır. Bunların her biri çok fazla sayıda nöron gönderici moleküller ihtiva eder. Akson üzerinden bir sinir darbesi ulaştığında bu kabarcıkların bazıları taşıdıkları bilgiyi sinaptik boşluk içersine bırakırlar. Böylelikle nöronlar arası haberleşme başlamış olur. Yeteri derecede uyarılmamış nöronlarda sinyal gönderir ancak kısa bir mesafe sonra sinyal kaybolur. (KAHRAMAN F., 2001)

7.2.2. Dentritler

Nöronlardan gelen birçok giriş sinyali hücre gövdesinden çıkan ve çok sık dallardan oluşan dentritler aracılığıyla hücre gövdesine aktarılır. Akson tarafından sinaptik boşluğun içersine bırakılan ve nörotransmitter olarak adlandırılan özel kimyasal maddeler dentrit içersindeki özel bir algılayıcı birime geçerek hücre gövdesine girerler. Nörotransmitterlardan bazıları uyarıcıdır ve hücreyi harekete geçirmek için bir çıkış darbesi üretmesine sebep olurlar. Diğerleri ise yasaklayıcıdır ve gelen darbeyi bastırırlar. Hücre gövdesi, dentritlerden gelen sinyalleri birleştirir ve sonuç sinyali eşik seviyesinin üzerinde ise bir darbe üretirler ve bu darbe aksonlar üzerinden diğer nöronlara aktarılır. (KAHRAMAN F., 2001)

7.2.3. Hücre Gövdesi

Hücre gövdesi nöronların harcadığı enerjiyi kontrol ederek, hücre içersindeki diğer aktiviteleri kontrol eder. Nöron hücresinin dış zarı sinir iletişim darbelerini üretme kabiliyetine sahiptir. Beyinde kimyasal ve elektriksel olarak iki çeşit haberleşme yapılmaktadır. Kimyasal sinyaller sinapsisler üzerinden, elektriksel sinyaller ise nöronlar arasında meydana gelmektedir. Hücrenin içersindeki sodyum konsantrasyonu, çevresindekinden on kat daha düşük olup, potasyum konsantrasyonu ise on kat daha yüksektir. Bu konsantrasyonlar zar içindeki gözenekler vasıtasıyla hücre içine sızarak konsantrasyon farkı giderilmeye çalışılır.

Nöronun harekete geçme hızı hücre gövdesi tarafından kısa bir zaman içerisinde uyarıcı ve yasaklayıcı girişlerin toplam etkisi ile belirlenir. (KAHRAMAN F., 2001)

7.2.4. Sinapsisler

Sinapsis, bir nöronun diğer bir kas hücresi ve bazen de bir salgı hücresi ile temas ettiği yerde bulunur. Canlılarda sinapsislerde elektriksel değil kimyasaldır. Sinapsisler sinyali tek yönlü iletirler. (KAHRAMAN F., 2001)

7.3. Yapay Sinir Ağlarına Yaklaşım

Yapay sinir ağları, insan beyninin işleyişini taklit ederek yeni sistem oluşturulmaya çalışılan yaklaşımlardır. Beynin işleyiş kuralları birçok yapay sinir ağı modelinin geliştirilmesinde kullanılmıştır. Yapay sinir ağları, insanlar tarafından gerçekleştirilmiş örnekleri (gerçek beyin fonksiyonlarının ürünü olan örnekleri) kullanarak olayları öğrenebilen, çevreden gelen olaylara karşı nasıl tepkiler üreteceğini belirleyebilen bilgisayar sistemleridir. İnsan beyninin fonksiyonel özelliklerine benzer şekilde, öğrenme, ilişkilendirme, sınıflandırma, genelleme, özellik belirleme ve optimizasyon gibi konularda başarılı bir şekilde uygulanmaktadırlar. Örneklerden elde ettikleri bilgiler ile kendi deneyimlerini oluşturur ve daha sonra, benzer konularda benzer kararları verirler. Yapay sinir ağları, insan beyninin özelliklerinden olan öğrenme yolu ile yeni bilgiler türetebilme, yeni bilgiler oluşturabilme ve keşfedebilme gibi yetenekleri herhangi bir yardım almadan otomatik olarak gerçekleştirmek amacı ile geliştirilmişlerdir. Yapay sinir ağları, öğrenmenin yanı sıra, ezberleme ve bilgiler arasında ilişkiler oluşturma yeteneğine de sahiptirler. Yapay sinir ağları tanımlarının bir kaç ortak noktası vardır. Bunların en başında yapay sinir ağlarının birbirine hiyerarşik olarak bağlı ve paralel olarak çalışabilen yapay hücrelerden oluşmaları gelmektedir. Proses elemanları da denilen bu hücrelerin birbirlerine bağlandıkları ve her bağlantının bir değerinin olduğu kabul edilir. Bilginin öğrenme yolu ile elde edildiği ve proses elemanlarının bağlantı değerlerinde saklandığı dolayısıyla dağıtık bir hafızanın söz konusu olduğu da ortak noktaları oluşturmaktadır. Proses elemanlarının birbirleri ile bağlanmaları sonucu oluşan ağa yapay sinir ağı denmektedir. Ağı oluşturan proses elemanları, bunların bilgileri işleme yetenekleri, birbirleri ile bağlantılarının şekilleri değişik modelleri oluşturmaktadır. Teknik olarak, bir yapay sinir ağı'nın en temel görevi, kendisine gösterilen bir girdi setine karşılık gelebilecek bir çıktı seti belirlemektir. Bunu yapabilmesi için ağ, ilgili olayın örnekleri ile eğitilerek (öğrenme) genelleme

yapabilecek yeteneğe kavuşturulur. Bu genelleme ile benzer olaylara karşılık gelen çıktı setleri belirlenir.

7.3.1. Yapay Sinir Ağlarının Kullanım Alanları

Yapay sinir ağları eğitim, finans, sağlık alanlarında endüstride, askeriyede ve diğer alanlarda başarı ile uygulanmaktadır. Bu yöntemin yaygın olarak kullanılmasını sağlayan ve güncel yapan, eksik bilgiler ile çalışabilme ve normal olmayan verileri işleyebilme yetenekleridir. Özellikle çok sayıda veriyi işleme gerektiren (radar verileri gibi) işlerde çok avantajlı sonuçlar üretebilmektedirler. Günümüzde birçok problem aslında şekil tanıma problemi haline getirilmekte ve ondan sonra çözülmektedir. Bu nedenle, yapay sinir ağlarının kullanılabilceği birçok alan vardır. Endüstriyel ve sosyal hayatta görülen binlerce örnek ile başarılı oldukları gösterilmiştir. Fakat her problemi yapay sinir ağı ile çözmek mantıklı olmayabilir. Eğer herhangi bir problemin çözümü için yeterli etkinlikte ve verimlilikte çözüm yöntemi söz konusu ise yapay sinir ağı'nın kullanılmasının bir anlamı yoktur. İlgili olay hakkında örneklerin olmayışı da (bulunmayışı da) bu ağları kullanmamak için önemli bir nedendir. Bir problemin yapay sinir ağı ile çözülmesi için, sadece yapay sinir ağı ile pratik çözümler üretilebilme durumunun söz konusu olması veya başka çözüm yolları olmasına rağmen yapay sinir ağlarının daha kolay ve daha etkin çözümler üretebilmesini sağlaması gerekir. Başarılı uygulamalar incelendiğinde yapay sinir ağlarının, doğrusal olmayan, çok boyutlu, gürültülü, karmaşık, kesin olmayan, eksik, kusurlu, hata olasılığı yüksek sensör verilerinin olması ve problemin çözümü için özellikle bir matematik modelin ve algoritmanın bulunmaması hallerinde yaygın olarak kullanıldıkları görülmektedir. Bu amaçla geliştirilmiş ağlar genel olarak şu fonksiyonları yerine getirmektedir:

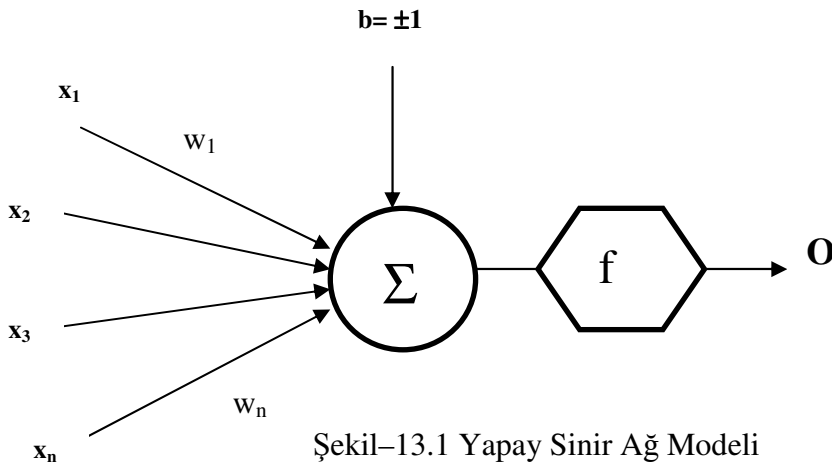
- 1)Örüntü tanıma
- 2)Sınıflandırma
- 3)Doğrusal olmayan sinyal işleme
- 4)İlişkilendirme veya örüntü eşleştirme
- 5)Doğrusal olmayan sistem modelleme
- 6)Zaman serileri analizleri
- 7)Sinyal filtreleme
- 8)Zeki ve doğrusal olmayan kontrol
- 9)Veri sıkıştırma.

Bunların ötesinde günlük hayatta kullanılan finansal uygulamalarından mühendisliğe ve tıp bilimine kadar bir çok alandan bazıları ise şöyle sıralanabilir:

- 1)Veri madenciliği
- 2)Optik karakter tanıma ve çek okuma
- 3)Bankalardan kredi isteyen müracaatları değerlendirme
- 4)Ürünün pazardaki performansını tahmin etme
- 5)Kredi kartı hilelerini saptama
- 6)Zeki araçlar ve robotlar için optimum rota belirleme
- 7)Güvenlik sistemlerinde konuşma ve parmak izi tanıma
- 8)Borsa ve para yönetimi
- 9)Kalite kontrolü
- 10)Bilgisayar oyunlarındaki zeki karakterlerin yaratılması
- 11)Radar ve sonar sinyalleri sınıflandırma

7.4. Yapay Sinir Ağlarına Giriş

Temel bir yapay sinir ağı hücresi biyolojik sinir hücresine göre çok daha basit bir yapıya sahiptir. En temel nöron modeli aşağıdaki şekilde görülmektedir. Yapay sinir ağı hücresinde temel olarak dış ortamdan ya da diğer nöronlardan alınan veriler yani girişler, ağırlıklar, toplama fonksiyonu, aktivasyon fonksiyonu ve çıkışlar bulunmaktadır. Dış ortamdan alınan veri ağırlıklar aracılığıyla nörona bağlanır ve bu ağırlıklar ilgili girişin etkisini belirler. Toplam fonksiyonu ise net girişi hesaplar, net giriş, girişlerle bu girişlerle ilgili ağırlıkların çarpımının bir sonucudur. Aktivasyon fonksiyonu işlem süresince net çıkışını hesaplar ve bu işlem aynı zamanda nöron çıkışını verir. Genelde aktivasyon fonksiyonu doğrusal olmayan (nonlinear) bir fonksiyondur. Şekilde görülen **b** bir sabittir, bias veya aktivasyon fonksiyonunun eşik değeri olarak adlandırılır. Nöronun matematiksel modeli şöyledir.



Şekil-13.1 Yapay Sinir Ağ Modeli

$$\text{Çıkış, } o = f(W.X + b)$$

Şekil-13.1'de nöron çıkışı hesaplanır. Buradaki W ağırlıklar matrisi, X ise girişler matrisidir. n giriş sayısı olmak üzere;

$$W = w_1, w_2, w_3, \dots, w_n$$

$$X = x_1, x_2, x_3, \dots, x_n$$

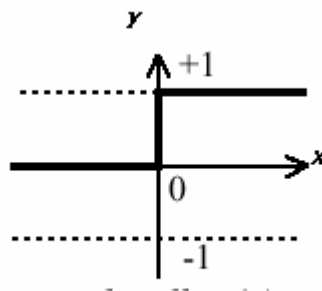
Şeklinde yazılabilir. Formalize edersek;

$$\text{net} = \sum_{i=1}^n w_i x_i + b \text{ ve } o = f(\text{net})$$

$$o = f\left(\sum_{i=1}^n w_i x_i + b\right) \text{ şeklinde de yazılabilir.}$$

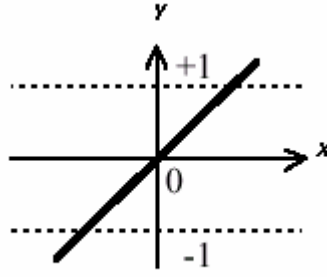
7.4.1. Aktivasyon Fonksiyonları

Aşağıdaki şekilde eşik aktivasyon fonksiyonunun grafiği görülmektedir. Eşik aktivasyon fonksiyonu eğer net değeri sıfırdan küçükse sıfır, sıfırdan daha büyük bir değer ise net çıkışında +1 değeri verir. Eşik aktivasyon fonksiyonunun -1 ile +1 arasında değişeni ise signum aktivasyon fonksiyonu olarak adlandırılır. Signum aktivasyon fonksiyonu, net giriş değeri sıfırdan büyükse +1, sıfırdan küçükse -1, sıfıra eşitse sıfır değerini verir.



Şekil – 13.1.1 Aktivasyon Fonksiyonu

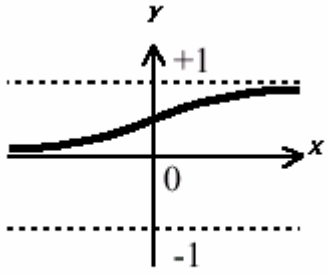
Aşağıdaki şekilde doğrusal aktivasyon fonksiyonu görülmektedir. Lineer aktivasyon fonksiyonunun çıkışı girişine eşittir. Sürekli çıkışlar gerektiği zaman çıkış katmanındaki aktivasyon fonksiyonunun lineer aktivasyon fonksiyonu olabildiğine dikkat edilmelidir.



Şekil – 13.1.2 Aktivasyon Fonksiyonu 2

$f(x) = x$ şeklinde ifade edilir.

Aşağıdaki şekilde logaritma sigmoid transfer fonksiyonu görülmektedir. Lojistik fonksiyon olarak da adlandırılmaktadır. Bu fonksiyonunun lineer olmamasından dolayı türevi alınabilmektedir böylece daha sonraki bölümlerde görülecek olan geri yayınlımlı ağlarda kullanmak mümkün olabilmektedir.



Şekil–13.1.3 Aktivasyon Fonksiyonu 3

Lojistik fonksiyonu,

$$f(x) = \text{lojistik}(x) = \frac{1}{1 + \exp(-\beta x)}$$

şeklinde ifade edilir. Buradaki β eğim sabiti olup genelde bir olarak seçilmektedir. Diğer bir aktivasyon fonksiyonu olan hiperbolik tanjant aktivasyon fonksiyonu da lineer olmayan türevi alınabilir bir fonksiyondur. +1 ile -1 arasında çıkış değerleri üreten bu fonksiyon lojistik fonksiyona benzemektedir. Denklemi aşağıda görüldüğü gibidir.

$$f(x) = \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

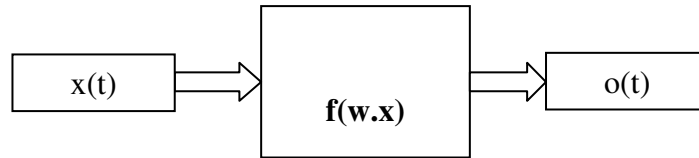
Bu aktivasyon fonksiyonlarından başka fonksiyonlar da vardır. Yapay sinir ağına hangi aktivasyon fonksiyonunun kullanılacağı probleme bağlı olarak değişmektedir. Yukarıda verilen fonksiyonlar en genel aktivasyon fonksiyonlarıdır.

7.5. Yapay Sinir Ağı Mimarileri

Yapay sinir ağları temel olarak iki grupta toplanır. Bunlar ileri beslemeli yapay sinir ağları ve geri beslemeli yapay sinir ağlarıdır.

7.5.1. İleri Beslemeli Yapay Sinir Ağları

Yapay sinir ağları giriş katmanı, saklı katman ve çıkış katmanı olmak üzere üç katmanlı bir yapıya sahiptir. Nöronlar bir sonraki katmanda bulunan nöronlara bağlanmalarına karşın aynı katmanda bulunan nöronlarla bağlantı kurmazlar. Bu tür ağ yapısında nöronlar arka arkaya beslenirler. Öğrenme aşamasında örnekler ağ girişine uygulanır. Birinci katmandaki nöronlar çıktıları hesaplayıp sonraki katmana girdi olarak sunarlar. Sırasıyla her tabaka aynı işlemi yapar. En uç tabakadaki çıktı değerleri de işlemi sonuçlandırır. İleri beslemeli yapay sinir ağları girdi ile çıktı arasında bir bağ kurar. Bulunan çıktı sadece o anki girişin fonksiyonudur.

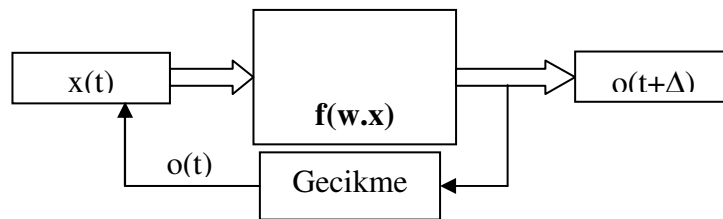


Şekil-14.1 Yapay Sinir Ağ Mimarisi

7.5.2. Geri Beslemeli Yapay Sinir Ağları

Bu tür ağlarda diğerlerinin aksine tabakalar arası bağlantının yanı sıra her bir nöronda birbiriyle bağlantılıdır. Bazı nöronların çıktıları ya kendilerine ya da kendinden önceki katmandaki nöronlara geri gönderilir.

Geri beslemeli nöral ağ, çıkışlar girişlere bağlanarak ileri beslemeli bir ağdan elde edilir. Ağın t anındaki çıkışı $o(t)$ ise, $t + \Delta$ anındaki çıkışı ise $o(t + \Delta)$ 'dır. Buradaki Δ sabiti sembolik anlamda gecikme süresidir. Çıktı aşağıdaki gibi yazılabilir.



Şekil-14.2 Yapay Sinir Ağ Mimarisi 2

$$o(t+\Delta) = f [W.o(t)]$$

Başlangıç anında $o(0) = x(0)$ 'dır.

Geri beslemeli yapay sinir ağlarının dinamik bir hafızası vardır. Belirli andaki çıktıları hem o anda hem de önceki girdilerin etkisiyle oluşmuştur.

7.6. Yapay Sinir Ağlarında Öğrenme

Yapay sinir ağları da insan beyni gibi örneklerle öğrenir. Yapay sinir ağlarında denetimli ve denetimsiz olmak üzere 2 temel öğrenme metodu vardır.

7.6.1. Denetimli Öğrenme

Yapay sinir ağının girişine uygulanan örneğe ait giriş ve bu örneğin yapay sinir ağı'nda istenen çıkış değerleri ağa verilir ve her çevrimde gerçek çıkış değeriyle ağın ürettiği çıkış değerleri karşılaştırılarak ağın hatası bulunur. Bu hata kabul edilebilir dereceye indirilene kadar, yapay sinir ağının nöronları arasındaki ağırlıkları değiştirilerek çevrime devam edilir. Denetimli öğrenme algoritmasında delta kuralı genelleştirilmiş delta kuralı veya hatanın geri yayılımı algoritması ve vektör niceliklerinin öğrenmesi örnek olarak verilebilir. Denetimli öğrenme genel olarak 3 tiptedir.

Hebbien Öğrenme Kuralına göre, her ikisi de yüksek derecede aktif olan işlem elemanlarından birisi diğerinden giriş alırsa aradaki bağlantı aralıkları arttırılmadılır. 1960 yılında Widrow ve Hoff tarafından geliştirilen Delta Öğrenme Kuralı, işlem elemanının gerçek çıkışı ile arzu edilen çıkış arasındaki fark değerinden hareket ederek, işlem elemanları arasındaki ağırlıklarının kontrol edilmesi esasına dayanır. Son olarak Rekabetli Öğrenme Kuralına göre, işlem elemanları kendi aralarında rekabet ederler ve verilen giriş için en güçlü tepkiyi veren nöron kendi kendini bu giriş değerlerine daha çok benzeyecek şekilde değiştirir. (KAHRAMAN F., 2001)

Geri yayılım algoritması denetimli öğrenmede kullanılan en genel algoritmadır. Basit olması ve iyi bir öğrenme kapasitesine sahip olması birçok alana uygulanmasını sağlamıştır.

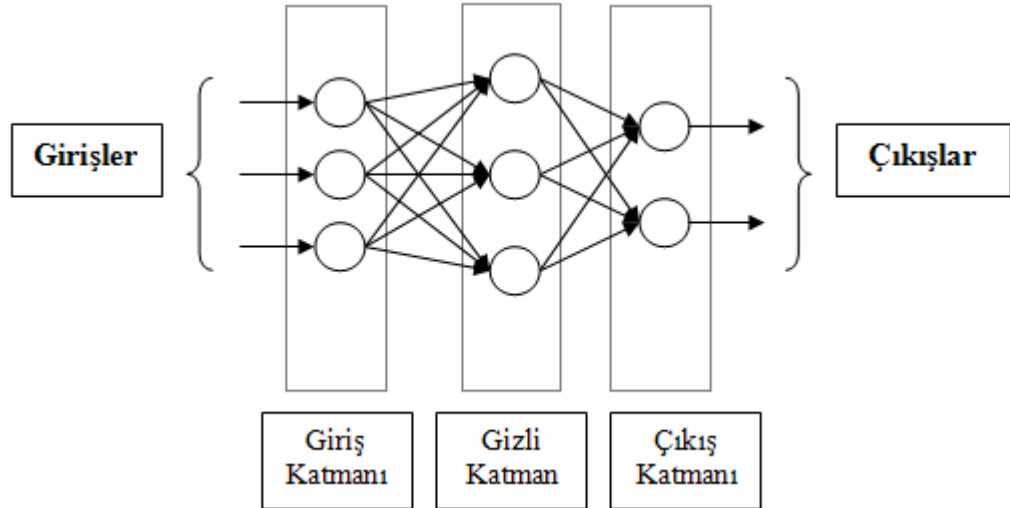
7.6.2. Denetimsiz Öğrenme

Bu tür öğrenmede ağa sadece giriş veri grubu sunulur ve ağdan bu veri grubuna uyumlu bir çıkış değeri üretecek şekilde kendisini uygun ağırlıklarla düzenlenmesi istenir. Denetimsiz öğrenmede çıkış değerlerinin verilmesine gerek yoktur. Yapay sinir ağının girişlerine giriş değeri verilir ve uygun çıkış değerleri üretmesi için kendi ağırlıklarını düzenlenmesi istenir. Denetimsiz öğrenme

algoritmalarına örnek olarak Kohonen ve Carpenter- Grassberg Rezonans Adapte Teorisi algoritmaları verilebilir. (KAHRAMAN F., 2001)

7.7. Geriye Yayılım Algoritması İle Öğrenme

Geriye yayılım algoritması öğrenmeye, ağırlıklara ilk değerlerin atanmasıyla başlar. Bazı yöntemlerin tersine, ilk değerlerin tahmin edilmesine gerek yoktur. Fakat ilk değerlerin küçük seçilmesi gerekmektedir. Büyük değerler eğitimi başarısız yapmaktadır.



Şekil-15.1 Geriye Yayılım Algoritması

Geriye yayılım algoritması; uygulanan girişin katmanlarda sonuç üreteceği ileri besleme ve hatanın bulunup katsayıların düzeltilebileceği geri besleme olmak üzere iki aşamadan oluşmaktadır. Burada giriş katmanı X, gizli katman çıktısı Y ve ağ çıktısı Z olarak tanımlanmıştır. Ağırlıkları, giriş ile gizli katman arasında W_{ih} ve gizli katman ile çıkış katmanı arasında W_{oh} olarak tanımlanabilir.

Ağırlıklara başlangıç değerlerinin atanması:

1. İlk değerler genelde küçük rastgele reel sayılardan seçilir. Başlangıç değeri olarak bunun -0.5 ve + 0.5 olması uygundur.

```
for(int i =0; i<giris_katman_sayisi-1; i++)
{
    for(int j = 0; j < gizli_katman_sayisi-1; j++)
    {
         $W_{ih}[i,j] = -0.5 + \text{rand}(-0.5,+0.5);$ 
    }
}
```

```
for(int i =0; i<gizli_katman_sayisi-1; i++)
{
    for(int j = 0; j < cikis_katman_sayisi-1; j++)
    {
        Woh[i,j] = -0.5 + rand(-0.5,+0.5);
    }
}
```

İleri Besleme:

2. Eğitim için kullanılan örneklerden rasgele bir örnek ağa giriş vektörü olarak seçilir.
3. Giriş vektörü gizli katmandaki her nörona uygulanarak nöron çıkışları hesaplanır. Önce her nörona gelen ağırlık değerleri ile bu ağırlıklı bağlantılı olan giriş değeri çarpılır ve bu çarpımlar toplanır.

$$Y_j \text{ hidden giriş} = \sum X_i \text{ giriş} * W_{ih}[i,j]$$

Sonra bu toplam aktivasyon fonksiyonundan geçirilerek gizli katman çıkışı hesaplanır.

$$Y_j \text{ hidden çıkış} = f(Y_j \text{ hidden giriş})$$

Bulunan sonuç Çıkış katmanının her bir nöronuna gönderilir.

```
for(int t =0; t<gizli_katman_sayisi-1; t++)
{
    toplam = 0;
    for(int s = 0; s < giris_katman_sayisi -1; s++)
    {
        toplam = toplam + X[s] * W_{ih}[s,t];
    }
    Y[t] = 1 / (1 + exp(-toplam));
}
```

4. Gizli katman nöronunun çıkışları, çıkış katmanına uygulanarak ağ çıktıları elde edilir. Her çıkış katmanı nöronuna gelen ağırlık değeri ile buna bağlantılı olan gizli katman çıktısı çarpılır ve çarpımlar toplanır.

$$Z_j.\text{çıkış nöronu girişi} = \sum Y_i.\text{hidden} * W_{oh}[i,j]$$

Hesaplanan toplam aktivasyon fonksiyonundan geçirilecek ağ çıktısı hesaplanır.

$$Z_j.\text{çıkış nöronu çıkışı} = f(Z_j.\text{çıkış nöronu girişi})$$

```
for(int t =0; t<cikis_katman_sayisi-1; t++)
{
    toplam = 0;
    for(int s = 0; s < gizli_katman_sayisi -1; s++)
    {
        toplam = toplam + Y[s] * Wih[s,t];
    }
    Z[t] = 1 / (1 + exp(-toplam));
}
```

Hatanın geriye yayılımı:

5. Bulunan çıkış nöron değerleri ile verilen k'ncü giriş örneğine karşılık gelen istenen çıkış karşılaştırılır. Çıkış nöronunun hata terimi hesaplanır.

$$\delta_{\text{çıkış}[k]} = (\text{istenençıkış} - Z_k)$$

Buradan hata düzeltme terimi hesaplanır.

$$\Delta W_{ho}[i,j] = \alpha \delta_{\text{çıkış}[k]} Z_k (1-Z_k) Y_j$$

```
for(int t =0; t<cikis_katman_sayisi-1; t++)
```

$$\delta[t] = Z[t] - \text{istenençıkış}[\text{ornekno},t];$$

```
for(int t = 0; t < gizli_katman_sayisi -1; t++)
```

```
for(int s = 0; s< cikis_katman_sayisi - 1; s++)
```

$$\Delta W_{jk} = - \alpha * Y[t] * Z[t] * (1 - Z[s]) * \delta[s];$$

Her gizli katman için önceden bulunan çıkış hata terimi ile gizli çıkış arasındaki ağırlıkların çarpımları toplanır.

$$\delta_{\text{çarpım}_j} = \sum \delta_k W_{ho}[j,k]$$

Hata terimi hesaplanır.

$$\delta_{gizli[j]} = \delta_{\text{carpim}_j} Y_j(1 - Y_j)$$

Hata düzeltme terimi hesaplanır.

$$\Delta W_{ih[j,k]} = \alpha \delta_{gizli[j]} X_j$$

```
for(int t = 0; t < gizli_katman_sayisi - 1; t++)
```

```
for(int s = 0; s < cikis_katman_sayisi - 1; s++)
```

$$\delta_t = \delta_t * \text{hidoutw}[t,s] * z[s] * (1 - z[s]) * \text{eout}[s];$$

Burada α öğrenme katsayısıdır.

6. Her çıkış nöronu ve gizli katman birimi için bulunan hata düzeltme terimleri eski katsayılara eklenerek ağırlık katsayıları yeniden düzenlenir.

Gizli katman ile çıkış katman arasındaki katsayılar için;

$$W_{ho[j,k]}(\text{yeni deęer}) = W_{ho[j,k]}(\text{eski deęer}) + \Delta W_{ho[j,k]}$$

Giriş ile gizli katman arasındaki katsayılar için;

$$W_{ih[j,k]}(\text{yeni deęer}) = W_{ih[j,k]}(\text{eski deęer}) + \Delta W_{ih[j,k]}$$

Hatanın Hesaplanması:

7. Bulunan katsayılar için gizli katman ve çıkış katmanı çıktı deęerleri hesaplanır ve hata deęeri bulunur. Hata, istenen çıkış ile hesaplanan çıkış arasındaki farkın karesiyle orantılıdır. K'ıncı örnek için hata;

$$E_k = (\text{istenen}_{\text{cikis}_k} - Z_k)^2$$

```
for(int t = 0; t < cikis_katman_sayisi - 1; t++)
```

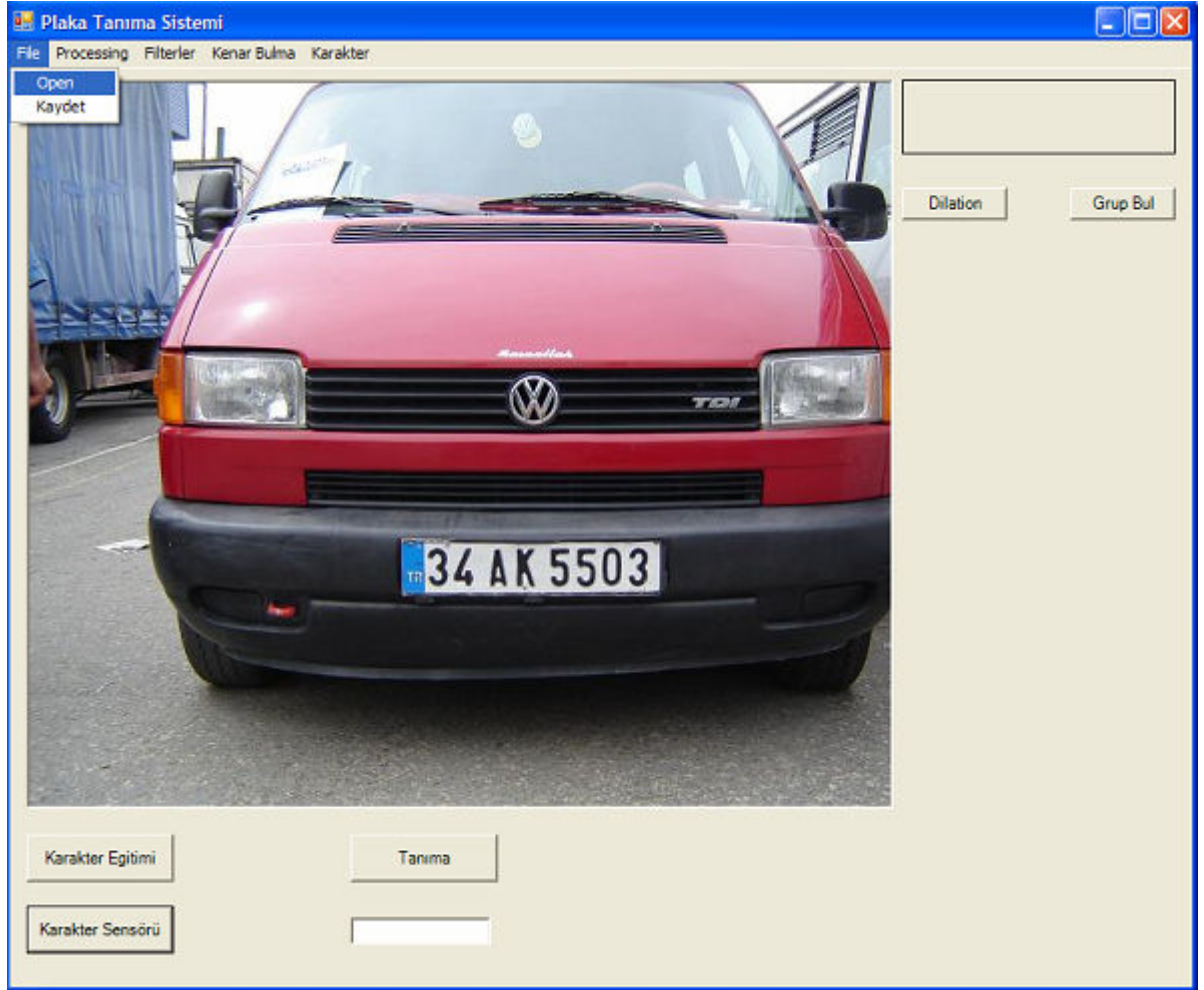
$$e = e + (\text{istenen}_{\text{cikis}[k,t]} - Z[t]) * (\text{istenen}_{\text{cikis}[k,t]} - Z[t]);$$

Hesaplanan bu hata deęeri kabul edilebilir maksimum hatadan büyükse başka bir örnek seçilerek eğitim algoritmasına yeniden başlanır.

Eğitim bütün örnekler için hatanın belirlenen bu E_{max} deęerinin altına düşmesiyle sonlandırılır.

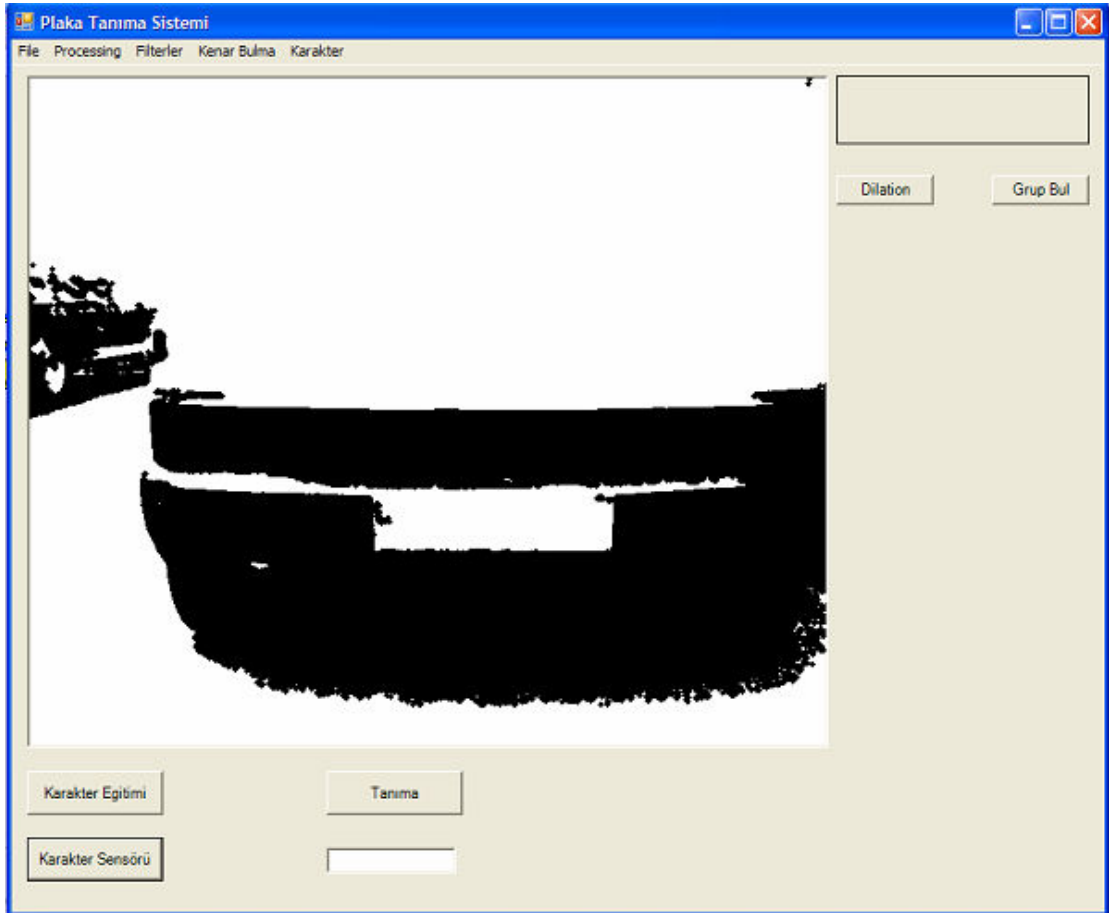
8. YAPILAN UYGULAMALAR

8.1. Görüntü Ön İşlemeleri



Sekil-16.1 Uygulama

İlk etapta file-open menüsünden araç resmini açıyoruz. Daha sonra Processing menüsünde bulunan fillenclosed sekmesini tıkladığımızda resim önce gri düzeye çevriliyor ve kenar bulma algoritması işletilip threshold' u alınıyor ve resim siyah ve beyazdan oluşan ikilik hale çevrilmiş oluyor.



Şekil – 16.2 Uygulama 2

Resim işlenip şekil-16.2'deki hale geldikten sonra smearing algoritması uygulanır. Bu algoritma olası plaka bölgesini çıkarmak için kullanılır. Bunun için yatay ve düşeyde uygulanan işlemle plaka bölgesi bulunur ve orijinal resimden bu koordinatlar çıkartılır.



Şekil-16.3 Örnek Plaka

Bu resmin ortaya çıkmasından sonra grup bul butonuna tıklanır. Böylece resim önce gri düzeye çevrilir sonra 130 eşik değeri belirlenerek siyah beyaza çevrilir. Ardından karakterlerin dışındaki siyah bölgeler temizlenir. Karakterlerin boyutları 14 x 27 gibi değer alabildiği için bunların dışındakiler yabancı madde olarak algılanır ve resimden atılır.



Şekil-16.4 İşlenmiş Örnek Plaka

Şekil-16.4'te görülen resimden karakter olmadığı anlaşılan bölgeler çıkartılır.

34 AK 5503

Şekil-16.5 Filtrelenmiş Örnek Plaka

Şekil-16.5' elde edildikten sonra karakterler yapay sinir ağına gönderilir. Bunun için karakter sensörü butonuna tıkladığımızda her seferinde bir karakter elde edilir ve tanıma butonu ile öngörülen karakter textbox'a yazılır.

8.2. Yapay Sinir Ağının Eğitilmesi

Eğitim için örnek olarak Türkçe karakterlerin ve rakamların bulunduğu 6 tane öğrenme seti için resim kullanıldı. Ayrıca kullanılacak daha fazla sayıdaki öğrenme seti tanıma işlemi için bir artı olacaktır.

Öğrenmesini istediğimiz karakter seti için resmi açıp karakter eğitimi butonuna tıklıyoruz böylece eğitim başlıyor. Her öğrenme seti 33 karakterden oluşmaktadır. Bunların 23 tanesi harf 10 tanesi rakamdır. Her bir karakter elde edilip 20 x 30'luk boyuta çevrildikten sonra yapay sinir ağına gönderilir.

9. SONUÇ

Bu çalışmada araç görüntülerinden plaka tanıyan bir sistem geliştirilmiştir. Yapılan çalışma için değişik mekânlarda çekilen araç resimleri test edilmiştir. Araçların görüntülerinden plakanın tespit edilmesinde ışığın geliş açısı resmin parlaklığı gibi etkenler plaka bölgesini bulamada zorlaştıran etkenlerdendir.

Araç plakasını tanımadaki en büyük sorunlardan biri de plaka üzerinde karakterlere yapışık olarak görülen vida veya çamur gibi izlerin olmasıdır. Çünkü program birbirine birleşik olan bu gürültüler ayıramaz. Ayrıca plaka karakterlerinin fontları ve ince ya da kalın olması plaka görüntüsünde hepsinde bir standart bulunmaması tanımayı zorlaştıran etkenlerdendir.

Plaka tanımada ortaya çıkan bu sorunlara rağmen bazı işlemlerin yapılmasıyla sistemin başarısının artırılması ön görülmektedir.

- Araç resminde yapılan threshold algoritması geliştirilerek ışık şiddetinin etkileri azaltılabilir. Bunun için resmin her yerine aynı eşik değerinin uygulanması yerine bölgesel eşik değeri belirleme plaka bölgesinin bulunmasında önemli bir etki yapabilir.
- Araç resimleri alınırken başka görüntülerde çerçeveye girmektedir. Bunun engellenmesi için ayrıca bir donanımsal sistem eklenerek sadece aracın resminin alınması sağlanabilir. Bunun için aracı bir bariyerle durdurup daha yakından resminin çekilmesi düşünülebilir.

10. KAYNAKLAR

ALGAN S., Her Yönüyle C#, Pusula Yayıncılık, İstanbul, 2004.

DENİZ U., Ardışıl Görüntülerde Hareket Analizi, Antakya, 2004.

ERDEM A. O., UZUN E., Yapay Sinir Ağları İle Türkçe Times New Roman, Arial Ve El Yazısı Karakterleri Tanıma, Ankara, 2004.

FREEMAN A. J., SKAPURA M. D., Neural Networks: Algorithms, Applications And Progaming Techniques, Addison Wesley Publishing Company, Houston, 1991.

GONZALEZ C. R., WOODS E. R., EDDINGS L. S., Digital Image Processing Using Matlab, Addison Wesley Publishing Company, New Jersey, 2004.

KAHRAMAN F., Sayısal Görüntü İşleme İçin Bir Yazılım Paketi Geliştirilmesi ve Sayısal Resimlerde İnsan Yüzünün Yer Tespitinde Kullanımı, Hatay, 2001.

KROSE B., SMAGT P., An Introduction To Neural Networks, University of Amsterdam, Amsterdam, 1996.

YELOĞLU Ö., UĞUR A., Modern Programlama Platformlarında Yapay Sinir Ağı Yazılımlarının Geliştirilmesi, İzmir, 2004.

ÜRGÜN Y., C# İle Temel Görüntü İşleme Yöntemleri, Erişim:

<http://www.csharpnedir.com/makalegoster.asp?MIId=657> (Ocak 2008).

ATASEVER V., C# İle Resim İşlemeye Kısa Bir Bakış, Erişim:

<http://www.csharpnedir.com/makalegoster.asp?MIId=659> (Ocak 2008)

KIRILOV A., Neural Network Ocr, Erişim:

http://www.codeproject.com/KB/cs/neural_network_ocr.aspx (Ekim 2007).

11. ÖZGEÇMİŞ

Melih Çiçek 1984 yılında İstanbul'da doğdu. İlk, orta ve lise öğrenimini İstanbul'da tamamladıktan sonra Kütahya Dumlupınar Üniversitesi Elektrik – Elektronik Mühendisliğini 2002 yılında kazandı ve 2006 yılında bitirdi. Bitirdiği aynı yıl Haliç Üniversitesi Bilgisayar Mühendisliği bölümünde yüksek lisansa başladı. Arçelik ve Aselsan'da yaptığı stajlar dışında profesyonel iş yaşamına ilk defa linkom teknolojide yazılım uzmanı olarak 2007 – 2008 yılları arasında çalıştı.