

**T.C.
YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**SAYISAL DONANIM GERÇEKLEMESİNE YÖNELİK EL YAZISI KARAKTER
TANIMA SİSTEMİ TASARIMI**

NURULLAH ÇALIK

**YÜKSEK LİSANS TEZİ
ELEKTRONİK VE HABERLEŞME MÜHENDİSLİĞİ ANABİLİM DALI
ELEKTRONİK PROGRAMI**

**DANIŞMAN
PROF. DR. VEDAT TAVŞANOĞLU**

İSTANBUL, 2013

T.C.
YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

**SAYISAL DONANIM GERÇEKLEMESİNE YÖNELİK EL YAZISI KARAKTER
TANIMA SİSTEMİ TASARIMI**

Nurullah ÇALIK tarafından hazırlanan tez çalışması 16.08.2013 tarihinde aşağıdaki jüri tarafından Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü Elektronik ve Haberleşme Mühendisliği Anabilim Dalı'nda **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Tez Danışmanı

Prof. Dr. Vedat TAVŞANOĞLU
Yıldız Teknik Üniversitesi

Jüri Üyeleri

Prof. Dr. Vedat TAVŞANOĞLU
Yıldız Teknik Üniversitesi

Prof. Dr. Herman SEDEF
Yıldız Teknik Üniversitesi

Prof. Dr. Oruç BİLGİÇ
İstanbul Kültür Üniversitesi

ÖNSÖZ

Mühendislik eğitimim boyunca gerek bilim insanı olma adına gerek düşünce yapısı ile bana birçok değer katan ve çok değerli nasihatleri ile farkındalık sağlayıp beni yönlendiren saygıdeğer hocam Prof. Dr. Vedat TAVŞANOĞLU' na teşekkürlerimi sunarım. Ayrıca tezim boyunca benden yardımlarını esirgemeyen Arş. Gör. Evren CESUR ile Arş. Gör. Dr. Nerhun YILDIZ hocalarıma da müteşekkirim.

Özellikle tezim boyunca beni hiç yalnız bırakmayan kıymetli dostlarım Burak AKGÜL Erdoğan ALDEMİR Sami EPBAYGÜN' e ve Erdal TEMİZ kardeşime ayrıca teşekkür ederim.

Benden her konuda desteğini esirgemeyen kıymetli ailem ile değerli eşime şükranlarımı sunarım.

İÇİNDEKİLER

	Sayfa
SİMGE LİSTESİ	vii
KISALTMA LİSTESİ	viii
ŞEKİL LİSTESİ.....	ix
TABLO LİSTESİ	x
ÖZET.....	xi
ABSTRACT	xiii
BÖLÜM 1	
GİRİŞ.....	1
1.1 Literatür Özeti	1
1.2 Tezin Amacı	3
1.3 Hipotez	4
BÖLÜM 2	
GENEL TEORİ	5
2.1 Yapay Sinir Ağları (YSA)	5
2.1.1 Yapay Sinir Ağının Yapısı ve Özellikleri.....	8
2.1.2 Yapay Sinir Ağlarının Çeşitleri	9
2.1.3 Hopfield Sinir Ağları	10
2.2 Hücrel Sinir Ağları	11
2.2.1 Hücrel Sinir Ağının Mimarisi.....	12
2.2.2 Komşuluk Tanımı	12
2.2.3 Hücresel Devre Yapısı ve Matematiksel Modeli	13
2.2.4 Klonlama Şablonlarının Matematiksel Modeli	16
2.3 Hücrel Sinir Ağları İle Görüntü İşleme.....	17
2.3.1 Doğrusal HSA İle Görüntü İşleme	19
2.3.2 HSA Denkleminin Nümerikleştirilmesi	19
2.4 HSA ile Filtreleme.....	21

2.5	HSA İle Gabor Filtresi Tasarımı	22
2.5.1	(2-B) Gabor Filtreleri	24
2.5.2	HSA İle Gabor-tipi Filtre Tasarımı.....	28
BÖLÜM 3		
HSA GABOR-TİPİ FİLTRELER İLE EL YAZISI KARAKTER TANIMA.....		36
3.1	Giriş	36
3.2	Gabor-tipi Filtreler ile El Yazısı Karakter Tanıma	37
3.2.1	Ön İşleme	38
3.2.2	Öznitelik Çıkartımı	39
3.2.3	Sınıflandırma.....	41
3.3	Tasarlanan Yeni Algoritma	42
3.3.1	Ön İşleme	43
3.3.2	Filtreleme ve Öznitelik Vektörlerinin Çıkartılması.....	45
3.3.3	Sınıflandırma.....	52
3.3.4	Sonuç	54
BÖLÜM 4		
MORFOLOJİK İŞLEMLER İLE EL YAZISI TANIMA.....		56
4.1	Giriş	56
4.2	Morfoloji Matematiği.....	57
4.2.1	Genişletme İşlemi.....	59
4.2.2	Aşındırma	60
4.2.3	Türetilen İşlemler.....	60
4.3	Morfoloji İle Yönlü Filtreleme	61
4.4	Morfolojik Filtreleme İle Harf Tanıma	63
4.4.1	Ön İşleme	64
4.4.2	Filtreleme.....	65
4.4.3	Öznitelik Vektörünün Çıkarımı.....	68
4.4.4	Sınıflandırma.....	70
4.4.5	Sonuç	70
BÖLÜM 5		
SONUÇ VE ÖNERİLER		72
KAYNAKLAR.....		73
EK-A		
EL YAZISI HARF TABLOSU		77
ÖZGEÇMİŞ		78

SİMGE LİSTESİ

$M \times N$	M satır N sütun boyutlarında ki görüntü
$\varphi(\cdot)$	YSA' nın aktivasyon fonksiyonu
Θ_i	YSA' nın bias değeri
$A(i,j;k,l)$	HSA geri besleme operatörü
A	Geri (besleme) klonlama şablonu
$B(i,j;k,l)$	HSA giriş(kontrol) operatörü
B	İleri (besleme) klonlama şablonu
$a_{k,l}, b_{k,l}$	Ave B şablonları matris elemanları
$C(i,j)$	HSA mimarisindeki i. satır j. sütun hücre ifadesi
S_{ij}	HSA mimarisinde $C(i,j)$ hücresinin etki küresi
T_s	Sayısal örnekleme için örnek alma süresi
1-B	Bir boyutlu
2-B	İki boyutlu
$h(x,y)$	Gabor fonksiyonu
$g(x,y)$	Gauss fonksiyonu
$\theta_{yön}$	Gabor filtresinin görüntü içerisinde filtrelediği yön
KÜ	Kompleks üssel
FIR	Finite Impulse Response
$W \times W$	FIR Gauss filtresinin şablon boyutları
σ	Gauss fonksiyonunun standart sapması
λ	Gabor filtrelerin band genişliği
$v(m,n)$	Doğrusal direnç-ızgara yapısındaki HSA' nın (m,n) noktasındaki gerilim değeri
ω_{x0}, ω_{y0}	Gabor ve Gabor-tipi filtrelerin merkez frekansları
ω_0, Ω	Uzamsal frekans genliği
Δ	HSA iterasyonundan çıkmak için pikseller arasında ki yeterli fark
$h_x(i)$	Yatay çizgisel iz düşümü
$h_y(j)$	Düşey çizgisel iz düşümü
$I_{HSA}(i,j)$	Gabor-tipi filtrelere giriş olarak verilen görüntü
$I_{bin}(i,j)$	Binary giriş görüntüsü
U_θ	θ yönündeki baskın bileşenli görüntünün 1×6 boyutlarındaki öznitelik vektörü
$d(p,q)$	p ve q vektörleri arasında ki mesafe
$A \ominus B$	A binary görüntüsünün B yapısal eleman ile aşındırılması

- $A \oplus B$ A binary görüntüsünün B yapısal eleman ile genişletilmesi
 $A \circ B$ Morfolojik açma işlemi
 $A \cdot B$ Morfolojik kapama

KISALTMA LİSTESİ

OKT	Optik karakter tanıma
HSA	Hücrel Sinir Ağları
SOM	Self-organizing Map
k-NN	k Nearest Neighbor
SVM	Support Vector Machine
YSA	Yapay Sinir Ağları
VHDL	VHSIC Hardware Description Language
KZFD	Kısa Zamanlı Fourier Dönüşümü
FIR	Finite Impulse Response
FPGA	Field Programmable Gate Array
AKDF	Ayrık Konum Fourier Dönüşümü
NSN	Nonlinear Shape Normalization
WTA	Winner Take All
FBM	Filtre Bankası Modülü
VÇM	Vektör Çıkartma Modülü
VHDL	VHSIC Hardware Description Language
KZFD	Kısa Zamanlı Fourier Dönüşümü
FIR	Finite Impulse Response
HDE	Histograma Dayalı Eşikleme
MM	Matematiksel Morfoloji
YE	Yapısal Eleman

ŞEKİL LİSTESİ

	Sayfa
Şekil 1. 1	“B” harfinin açılara göre yönlü bileşenleri..... 4
Şekil 2. 1	Doğal Sinir Hücresinin Yapısı..... 6
Şekil 2. 2	Doğal Nöron ile oluşturulan yapay sinir hücresi..... 7
Şekil 2. 3	Katmalar halindeki hücreler ile oluşturulan bir YSA 8
Şekil 2. 4	Dinamik Hücre Modeli 10
Şekil 2. 5	Hopfield sinir ağına bir örnek..... 11
Şekil 2. 6	Kartezyen HSA mimarisi..... 12
Şekil 2. 7	Farklı r değerleri için C(i,j)’ye komşu hücreler 13
Şekil 2. 8	HSA içerisindeki bir hücrenin devre yapısı..... 13
Şekil 2. 9	Hücrenin çıkış fonksiyonu 14
Şekil 2. 10	Bir hücrenin blok diyagramı 16
Şekil 2. 11	HSA ile görüntü işleme modeli..... 18
Şekil 2. 12	İntegrale dikdörtgen yaklaşıklığı 19
Şekil 2. 13	HSA ile kenar belirleme uygulaması 21
Şekil 2. 14	HSA ile alçak geçiren filtreleme uygulaması 22
Şekil 2. 15	Bir işaretin KZFD ile analizi 23
Şekil 2. 16	Fourier düzlemindeki karşılığının daralması 24
Şekil 2. 17	Bir işaretin zaman-frekans düzleminde olabilecek minimum belirsizliği.... 24
Şekil 2. 18	Gauss ve Gabor filtrelerinin frekans cevapları ($w_{x0} = \frac{\pi}{2}, w_{y0} = \frac{\pi}{2}, \sigma = 2$)... 25
Şekil 2. 19	Gauss ve modülasyon ile elde edilen Gabor filtreleri 26
Şekil 2. 20	Giriş görüntüsünden 135° açısında yönelmiş bileşenlerin çıkartılması 27
Şekil 2. 21	Düşük değerli W için pencereleme etkisi..... 28
Şekil 2. 22	(2.23b)’ ye göre tasarlanan şablon ile pencereleme etkisinin giderilmesi .. 28
Şekil 2. 23	D. dirençli ızgara yapısı ile HSA alçak geçiren filtrenin devre gerçekleştirilmesi 30
Şekil 2. 24	$H(e^{jw_x}, e^{jw_y})$ ’ nin frekans karakteristiği 31
Şekil 2. 25	Gauss benzeri filtrelerin frekans karakteristikleri..... 32
Şekil 2. 26	Gabor-tipi filtrelerin frekans karakteristikleri..... 33
Şekil 2. 27	İki boyutlu HSA Gabor-tipi filtrenin analog devresi 34
Şekil 3. 1	[29]’ de tasarlanan EKT sisteminin genel blok diyagramı 37
Şekil 3. 2	Giriş görüntüsünün yatay (h_x) ve düşey (h_y) iz düşümleri..... 39
Şekil 3. 3	Farklı uzamsal frekans genliklerinde ve 4 farklı açıda bulunan Gabor-tipi filtreler(sol), Filtrelerin birim impuls yanıtları (sağ)..... 40
Şekil 3. 4	WTA ile karşılaştırmadan çıkan 4 farklı açıdaki baskın bileşenli görüntülerin sınıflandırılması 42

Şekil 3. 5	Algoritmanın blok diyagramı.....	42
Şekil 3. 6	Eğim düzeltme algoritması ile eğik harfin düzeltilmesi ı	44
Şekil 3. 7	a) Eğik ve boyutu ayarlanmamış harf b) Eğikliği giderilmiş olan harf	44
Şekil 3. 8	a) İskeleti çıkarılmış harf b) YE ile genişletilmiş harf	45
Şekil 3. 9	Öznitelik çıkartma bloğunun iç yapısı	46
Şekil 3. 10	“O” harfinin FFT’ si ve filtrelerin merkez frekansları	47
Şekil 3. 11	Filtrelerin çıkışları	48
Şekil 3. 12	90° ve orta frekans bölgesindeki filtre çıkışının gösterimi.....	48
Şekil 3. 13	HDE ile elde edilen ikili değer türündeki bileşenler	49
Şekil 3. 14	Aynı açıdaki görüntülerin birleştirilmesi ile elde edilen birleşik görüntüler	50
Şekil 3. 15	İçerdikleri bileşenlere göre dik olan normal doğrusu ile ayrılmış görüntüler	50
Şekil 3. 16	0° ve 45° için iz düşümü vektörleri ve bölgesel toplamları.....	51
Şekil 3. 17	90° ve 135° için iz düşümü vektörleri ve bölgesel toplamları.....	51
Şekil 3. 18	Sınıflandırıcıya yeni üye girişi	52
Şekil 3. 19	En yakın üç uzaklığın tespit edilmesi.....	52
Şekil 3. 20	Yeni üyenin bir sınıfa dahil edilmesi	53
Şekil 4. 1	Binary (ikili) görüntülerde temel mantık işlemleri.....	57
Şekil 4. 2	MM yöntemler kullanılarak görüntüye ait sınır bölgelerinin tespiti	58
Şekil 4. 3	MM genişletme işlemi uygulama örneği	59
Şekil 4. 4	WTA ile karşılaştırıcıdan çıkan 4 farklı açıdaki baskın bileşenli görüntülerin sınıflandırılması	60
Şekil 4. 5	L = 21 uzunluklu YE.....	61
Şekil 4. 6	El Yazısı “O” Harfi	62
Şekil 4. 7	L = 21 olan YE’ ler için “O” harfi üzerine uygulanmış açma işlemi.....	62
Şekil 4. 8	Tasarlanan Morfoloji filtre bankası	63
Şekil 4. 9	Filtre bankasının çıkışları.....	64
Şekil 4. 10	Tasarlanan sistemin blok diyagramı.....	65
Şekil 4. 11	a)Eğik ve boyutu ayarlanmamış harf b)Eğikliği giderilmiş 64 x 64olan harf	65
Şekil 4. 12	a) İskeleti çıkarılmış harf b) YE ile genişletilmiş harf	66
Şekil 4. 13	Morfolojik filtre bankası için tasarlanan 24 farklı YE	66
Şekil 4. 14	Filtre için giriş görüntüsü	67
Şekil 4. 15	0° (üst) ve 135° (alt) için filtreleme	67
Şekil 4. 16	Filtrelenecek olan “B” harfi görüntüsü	68
Şekil 4. 17	“B” harfinin filtre çıkışları.....	68
Şekil 4. 18	Aynı açıdaki filtre çıkışlarının toplanması ile oluşan görüntüler	69
Şekil 4. 19	İz düşüm eksenini ve 90° de ki bileşenler	69
Şekil 4. 20	Ayrılan sağ ve sol bölgelerin iz düşüm toplamları.....	70

ÇİZELGE LİSTESİ

	Sayfa
Çizelge 2. 1	Filtre şablonunun 1 piksel için işlem yükü hesabı..... 29
Çizelge 2. 2	HSA Gabor-tipi filtrenin 1 piksel için hesap yükü..... 34
Çizelge 3. 1	Test kümeleri ve doğruluk yüzdeleri 55
Çizelge 4. 1	Test kümeleri ve doğruluk yüzdeleri 71
Çizelge 5. 1	7 farklı kullanıcı ile oluşturulmuş 10 farklı harf setleri..... 77

SAYISAL DONANIM GERÇEKLEMESİNE YÖNELİK EL YAZISI KARAKTER TANIMA SİSTEMİ TASARIMI

Nurullah ÇALIK

Elektronik ve Haberleşme Mühendisliği Anabilim Dalı

Yüksek Lisans Tezi

Tez Danışmanı: Prof. Dr. Vedat TAVŞANOĞLU

Günümüz dünyasında insan makine etkileşiminin giderek artmaktadır. Geliştirilen yeni algoritmalar ile makineler yapay zeka kazandırılabilen ve kullanılan otomat sistemler insanların günlük hayatında birçok işi kolaylaştırmaktadır. Bu sistemlerin geliştirilmesinde görüntü işleme teknikleri yoğun bir şekilde kullanılmaktadır.

Görüntü işleme teknikleri ile birçok problem tasarlanan otonom sistemler ile çözülmektedir. Bu problemlerin başında optik karakter tanımanın bir alt dalı olan el yazısı karakter tanıma gelmektedir.

El yazısı karakter tanıma yönelik, tanıma oranı yüksek algoritmalar literatürde mevcuttur. Geliştirilen bu algoritmaların tanıma oranları yüksek olmasına rağmen içerdikleri karekök, bölme gibi sayısal tasarımlar için hesap yükü oluşturan matematiksel işlemlerden dolayı FPGA gibi donanımlarda gerçeklemeleri zordur.

Bu problem göz önünde bulundurularak hücreli sinir ağları (HSA) ile tasarlanan Gabor-tipi filtreler ile harflerin içerdikleri yönlü bileşenler çıkartılarak hızlı bir tanıma algoritması tasarlanabilir. Literatürde bulunan birçok çalışma ile HSA Gabor-tipi filtreler FPGA donanımında verimli bir şekilde gerçekleştirilebilmektedir. Bu sayede Gabor-tipi filtrelerin yön seçici özelliği ile hızlı ve tanıma oranı yüksek algoritmalar oluşturulabilir.

Ayrıca işlenen görüntülerin ikili (binary) türünde olduğu göz önünde bulundurularak Gabor-tipi filtrelerin yön seçici filtreleme yapması gibi morfolojik işlemlerle de harf görüntüsü üzerinden yönlü bileşenler çıkartılabilir.

Bu alıřmada sayısal tasarımlara uygun hızlı ve doęruluk oranları % 80 ve % 91 civarında olan el yazısı harf tanıma üzerine iki farklı algoritma geliştirilmiştir. İlk algoritmada HSA Gabor-tipi filtreler kullanılmış; ikinci algoritmada ise Gabor-tipi filtreler gibi yönlü bileşenleri çıkartabilen morfolojik yönlü filtreme anlatılmıştır.

Anahtar Kelimeler: El yazısı karakter tanıma, hücrel sinir aęları, Gabor filtreleri, morfolojik filtreleme

**HANDWRITTEN CHARACTER RECOGNATION SYSTEM DESING
FOR DIGITAL HARDWARE IMPLEMENTATION**

Nurullah ÇALIK

Department of Electronics and Communications Engineering

MSc. Thesis

Advisor: Prof. Dr. Vedat TAVŞANOĞLU

Interaction relationship between human and machine is increasing in modern-day gradually. The machines are acquired artificial intelligence by improved new algorithms and current autonomous systems facilitate various services in the human daily life. Image Processing technique are intensively used for developing this systems.

A various problems are solved by using designed autonomous structures. Handwriting recognition, which is the subfield of optical characters recognition, is one of the major of the problems.

There are successful algorithms on handwriting recognition according to recognition rate in technical literature. Improved algorithm has also high recognition rate however the algorithm has difficulties by implementation on FPGA or other hardware because of contenting square root, division operations which get more load of mathematical processing.

According to this specific problem, a fast recognition algorithm can be designed by obtaining oriented components of letters using Gabor type filters in cellular neural networks (CNN). CNN gabor type filters are implemented in many literature works efficiently. By this means, fast recognition rate algorithms are designed with obtaining oriented components feature of Gabor filters.

On the other hand, oriented components could be obtained from the letters by morphological operations such as Gabor type filters direction selector structure considering the images which is processing are binary.

In this study, two different algorithms which is convenient for digital design and has 80% to 91% accuracy rate are improved on handwriting recognition. In first method CNN gabor type filters are used, in other methods morphological operations such as Gabor type filters direction selector structure is used.

Keywords: handwriting character recognition, cellular neural Networks, Gabor filter, Morphological filter

1.1 Literatür Özeti

Teknolojinin ilerlemesi ile beraber insanoğlunun bilgi birikimi inanılmaz boyutlara gelmiştir. Gelişen sensör teknolojisi ile dış dünyadan alınan görüntü, ısı, ses ve manyetik alan gibi bilgiler toplanabilmektedir. Toplanan bu verilerin sayısal sistemler ile işlenmesi, analiz edilmesi ve verilerin sınıflandırılması bilim insanlarının üzerinde çalıştığı başlıca konulardan biridir [1]. Bilginin işlenmesi ve sınıflandırması ile insan-makine iletişimi sağlanmış ve birçok zor problemler makinelerin eğitilmesi ile çözülmüştür. Özellikle bilgisayar gibi sayısal sistemler için geliştirilen görüntü ve ses işleme konusunda ki algoritmalar ile yapay zeka uygulamalarında hız kazanılmıştır.

Görüntü tanıma problemi sayısal işaret işlemenin en önemli problemlerinden biridir. Günlük hayatta kullanılan bilgi akışının büyük bir oranı görüntüler üzerinden olmakta ve zamanla bu oran daha da artmaktadır. Bu nedenle görüntü işleme birçok uygulama alanında büyük kolaylık sağlamaktadır. Özellikle insandan bağımsız otomat cihazların tasarımında önemli bir yer tutar.

Görüntü işlemenin en temel problemlerinden biri olan karakterler tanıma ve dokümanların sınıflandırılması, 1960' lı yıllarda ki bilgisayar teknolojisinin hızla gelişmesi ile üzerinde araştırmalar yapılan popüler konuların başında gelmiş ve geliştirilen sayısal sistemler geniş bir uygulama alanı bulmuştur. Bu uygulamalar el yazısı metinlerin sayısal ortamlara aktararak işlenmesi, tarihi belgelerin arşivlenmesi ve posta kurumlarında zarfların sınıflandırılması gibi önemli problemlerin giderilmesinde kullanılmaktadır. Yazı tanıma oranlarının yüksek olması bu otomat sistemlerin birçok devlet kurumunda ve şirkette

kullanılmasına ve maliyeti düşürerek hayatı kolaylaştırmasına neden olmuştur. Ucuz olan karakter tanıma algoritmaları ile şirketlerin bilgi girişine ki maliyetleri % 55 - % 90 oranında düşürebilmektedir [2].

El yazısı karakter tanıma problemine yönelik, kendi içlerinde avantajları ve dezavantajları olan birçok yöntemler ve sayısal sistemler geliştirilmiştir [3]. Bu yöntemler iki ana grupta toplanabilir: Etkileşimli (çevrimiçi) ve etkileşimsiz (çevrimdışı) yöntemler. Etkileşimli sistemler el yazısını yazıldığı sırada tanımaya yönelik geliştirilen sistemlerdir. Buna elektrostatik kalemler kullanılarak yazılan tabletler gösterilebilir. Etkileşimsiz sistemler ise genelde kağıt üzerine yazılmış olan metinlerin kamera veya tarayıcı gibi sistemler ile görüntüledikten sonra işlemesi üzerine geliştirilen sistemlerdir [4]. Etkileşimsiz olarak geliştirilen bu algoritmaların çoğu optik görüntülerin işlenmesine yönelik olup genel olarak Optik Karakter Tanıma (OKT) ("Optical Character Recognition") olarak sınıflandırılır.

OKT sistemleri el yazısı tanıma işleminde iki ana problem ile karşılaşılır: 1) Çok miktarda karakter olması 2) Benzer şekilde karakterlerin olmasıdır [5]. Bunun yanı sıra harfin görüntü içerisinde yeri, yazıcıya göre şekli, kalem boyutuna göre kalınlığı, eğikliği ve büyüklüğü gibi problemler OKT sistemlerini zorlamakta tanıma oranlarını düşürmektedir.

Gene bir OKT sistemi;

- Ön işleme
- Öznitelik çıkartma
- Sınıflandırma

olmak üzere üç ana adımdan oluşmaktadır.

Ön İşleme

Bu adımda görüntüleyici tarafından alınan ve boyutları rastgele olan görüntüler gürültüleri giderildikten sonra boyutları sabit bit $M \times N$ boyutlarında ikili görüntü formatına getirilir. Harfin eğikliği, kalınlığı, resim içerisindeki yeri gibi problemler bu adımda giderilir.

Öznitelik Çıkartma

Ön işleme adımından gelen boyutları sabit $M \times N$ olan görüntü üzerinden harfin kimliğini belli edecek özniteliklerin çıkartıldığı adımdır. Öznitelikler için önemli olan kendi sınıfı içerisindeki

değişintisinin minimum diğer sınıflar arasında ki değişintinin maksimum olmasıdır. Bu sayede daha doğru sınıflandırılabilir ve doğruluk oranı yükselir.

Öznitelik çıkartmak için birçok yöntem geliştirilmiştir. Karhunen-Loeve dönüşümü, açılal radyal dönüşümü ve Zernike momentleri gibi dönüşümlerin yanı sıra görüntü içerisindeki harfin bağlantı noktaları, pikseller arasında ki komşulukları ve zincir kodu gibi şekli özelliklerinden yola çıkarak geliştirilen şekilsel öznitelik çıkarıcılar geliştirilmiştir [6]. Dönüşüm yöntemlerinin doğruluk oranları yüksek olmasına rağmen sayısal sistemler için gerçeklemesi zor olan karekök alma, bölme gibi matematiksel işlemler içermektedir. Bu sistemlerin sayısal tasarımları hem maliyetli olup hem de zaman ve güç tüketimleri de yüksektir. Şekilsel öznitelik çıkarıcılar ise dönüşümlere göre daha kolay sayısal tasarım yapılarına sahipken tanıma oranları düşüktür.

Sınıflandırma

Sınıflandırma, harfe ait özniteliğin veri tabanı ile karşılaştırılıp hangi sınıfa ait olduğuna karar verme problemidir. Görüntü işlemede yapay sinir ağları, özdüzenleyici haritalar ("Self-organizing Map ") (SOM), en yakın k komşu ("k nearest neighborhood") (k-NN) ve destek vektör makineleri ("Support Vector Machine") (SVM) gibi birçok algoritma kullanılmaktadır.

Özellikle OKT sistemlerinde k-NN matematiksel olarak basitliği ve tanıma oranlarındaki başarısından dolayı tercih edilir [6][7].

1.2 Tezin Amacı

OKT için tasarlanan algoritmalar için başarı oranı olarak tanıma oranının yanında algoritmanın FPGA ve DSP gibi sayısal mimariler için uygunluğu ve hızı da önemli bir parametredir. Literatürde bulunan algoritmalar tanıma oranının yüksek olması için öznitelik çıkartma ve sınıflandırma kısımlarında karmaşık matematiksel işlemler içeren yöntemler kullanmışlardır [8][9][10]. Bu algoritmaları içerdikleri yoğun matematiksel işlemlerden dolayı sayısal sistemlerde ki tasarımı oldukça karmaşık ve işlem yükleri yüksektir.

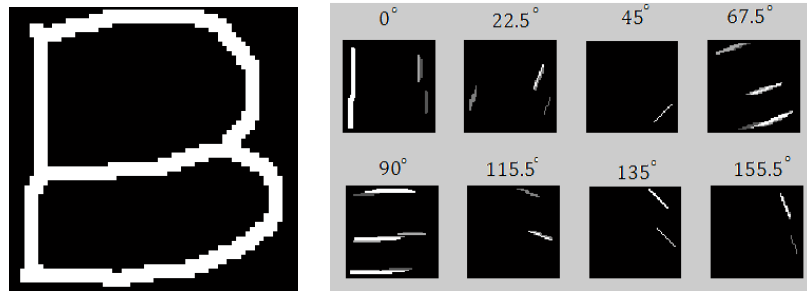
Bu tezdeki çalışmanın amacı, hız ve karmaşıklık problemi göz önünde bulundurularak tanıma oranı yüksek ve sayısal mimarilere uygun bir algoritma geliştirmektir.

1.3 Hipotez

OKT için tasarlanan algoritmaların işlem yükünün büyük bir bölümünü öznitelik çıkartma ve sınıflandırma adımları içermektedir. Buradaki adımlar için kullanılan yöntemler OKT sisteminin tanıma ve tasarım performansını belirlemektedir.

Öznitelik çıkartma adımında harfin şekilsel özelliklerine bakarak harfin kimliğini belirleyen vektörün çıkartılması sayısal tasarım için uygundur. Şekilsel öznitelik çıkartmada harfin istatistiksel özelliklerine bakılmaktadır. Literatürde harfin Freeman zincir kodu [12][13], kontur bilgisi [14][15] veya harf içerisindeki bağlantı noktası gibi şekilsel özellikler öznitelik vektörünün çıkartılmasında sıklıkla kullanılır. Şekilsel algoritmalar ile her ne kadar hızlı ve işlem yükü düşük algoritmalar geliştirilse bile bu algoritmaların tanıma oranları düşük seviyelerde kalmaktadır [6].

Bu problem göz önünde bulundurularak harflerin yönlü bileşenleri kullanılarak (Şekil 1.1) yeni bir şekilsel öznitelik çıkarıcı önerilmiştir. Yönlü bileşenlerin çıkartılmasında başlıca kullanılan yöntemler arasında Gabor filtreleri gelmektedir. Gabor filtrelerin yönlü bileşenleri geçirme özelliği görüntülerin kenar bilgisinin çıkartılmasında, iyileştirilmesinde ve görüntülerden öznitelik çıkartımında sıklıkla kullanılmasına rağmen sayısal tasarımlar için ciddi bir işlem yükü getirmektedir. El yazısı harf tanıma için yönlü bileşenlerin çıkartılması için hücrel sinir ağları (HSA) ile tasarlanan Gabor-tipi filtreler kullanılabileceği [16] ve [17]'de gösterilmiştir. Ayrıca [18] ve [19]'de HSA Gabor-filtrelerin FPGA için uygun tasarımları geliştirilmiştir. Bütün bu özellikler kullanılarak HSA Gabor-tipi filtreler ile bir OKT sistemi tasarlanabilir.



Şekil 1.1 "B" harfinin açılara göre yönlü bileşenleri

GENEL TEORİ

Hücrel sinir ağı ile karakter tanıma algoritmasının öncesinde bu bölümde yapay sinir ağı ve hücrel sinir ağlarının keşfinden ve matematiksel yapısından bahsedilmiştir. Özellikle hücrel sinir ağlarının görüntü işlemedeki uygulamaları ve görüntülerden öznitelik çıkartmaya yönelik kullanımı daha ayrıntılı anlatılmıştır.

2.1 Yapay Sinir Ağları (YSA)

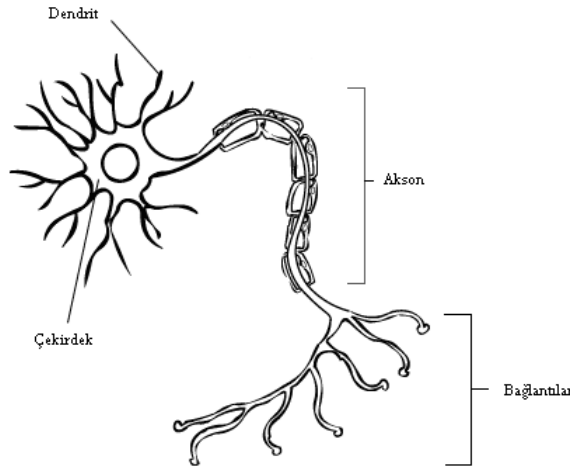
Bilim birçok alanda doğayı taklit etmektedir. Doğada tasarlanmış olarak bulunan problem çözme, veri işleme ve optimizasyon mekanizmaları günümüzde birçok bilim sahası için ilham kaynağı olmuştur. Bu tasarımlardan biri de insan beynidir. İnsan beyninin problem çözme yetisi ve üstün özellikleri bilim insanlarını üzerinde çalışmaya teşvik etmiştir. Bu çalışmalar sayesinde beyin sinir hücresinin matematiksel modelleri geliştirilmiştir. Bu modeller doğrultusunda da yapay sinir hücreleri ve ağ yapıları tasarlanmıştır. Yapay sinir ağları sayesinde yeni problem çözme ve veri işleme algoritmaları geliştirilmiş ve bu algoritmalar yapısı itibariyle çeşitli problemlere uygulanabilmektedir.

Beyin en hızlı analog paralel bilgisayardır. Paralel veri işleme yapısı sayesinde kendisinden milyonlarca kat daha hızlı bilgisayarlardan daha hızlı bir şekilde veri işleyebilmektedir. Bilgisayar veri depolamada ve seri bilginin işlenmesinde beyinden daha üstündür. Ama ses ya da görüntülerin tanınması ve etiketlenmesi, durumlar karşısında hızlı karar verme ve yorum yapabilmesi gibi yönlerden bakıldığında beyin tartışılmaz bir üstünlüğü vardır. Aslında beyin sadece 100 çevrim hızı ile çalışmaktadır. Kullanılan bilgisayarlara göre oldukça düşük bir hız olmasına rağmen, yapısında bulunan 10^{11} nöronun paralel çalışması ile daha hızlı bilgi işleyebilmektedir.

YSA hakkında ilk çalışmalar 1940'ların başlarına dayanır. McCulloch ve Pitts tarafından yürütülen ilk çalışmalar yapay nöron çalışmalarına farklı bir boyut kazandırmıştır. İki bilim adamı beyin içerisindeki hücrelerin çalışma mekanizmasına ilişkin hücrenin belli bir eşikten sonra çıkış ürettiği fikrini ortaya atmışlardır. Bu fikir doğrultusunda ilkel YSA'lar tasarlanmıştır. Bu basit görünen fikir, YSA devresinin paralel işleyişi ile karmaşık hesapları yapabilen bir yapıya dönüşmüştür.

1960' lı yılların başlarında beynin işleyişinin biraz daha iyi anlaşılması ile yapay nöron modellerinde çeşitlilik artmıştır. Bunlara örnek olarak Rosenblatt tarafından geliştirilen 'Perseptron', Widrow ve Hoff'un türettikleri 'Adaline' , 1980'lerde ortaya atılan Hopfield dinamik YSA' sı ve Chua' nın Hopfield ağlarından türettiği hücresel sinir ağları (HSA) örnek olarak verilebilir.

YSA' nın yapısı, doğal sinir hücreleri örnek alınarak türetildiğinden dolayı YSA' yı anlamak için bir sinir hücresinin yapısını bilmek gerekir. Sinir sisteminin en temel birimine sinir hücresi yada nöron denilmektedir. Bir sinir hücresinin yapısı üç bileşenden meydana gelir. Bunlar dentritler, hücre çekirdeği ve aksondur. İki hücrenin bağlantı noktasına da sinapsis denilmektedir.



Şekil 2.1 Doğal Sinir Hücresinin Yapısı

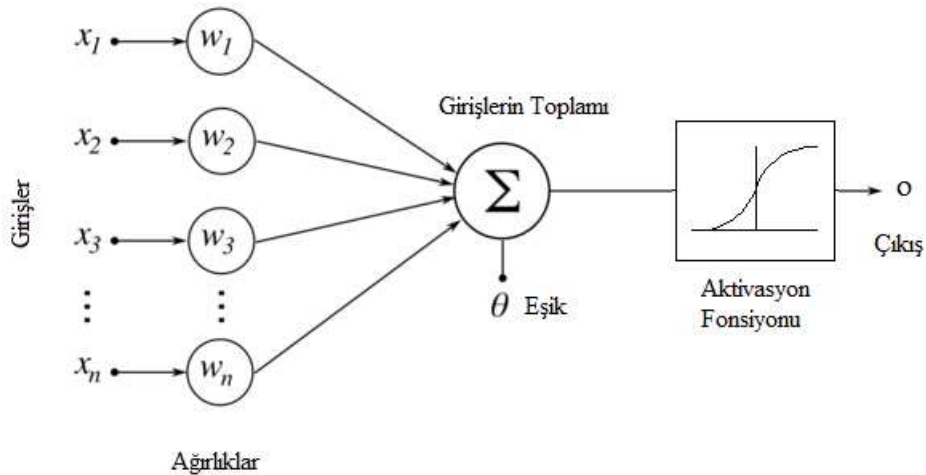
Dentritler : Hücre gövdesine bağlı olan kısa kısa kıvrımlı uzatılar şeklinde bulunan bu yapılar hücrenin, komşu hücrelerden gelen sinyallerin alındığı yeridir. Gelen sinyaller dentritlerde toplanır ve hücre çekirdeğine yönlendirilir. Herbir dentrit ile hücre çekirdeği arasındaki iletim farklı ağırlık oranlarında olmaktadır. Yani bazı dentritler ağırlıklı paya sahipken diğerleri pasiftirler. Bu hücreye veri seçiciliği özelliğini kazandırmaktadır.

Hücre Çekirdeği : Dentritler üzerinden gelen sinyallerin toplandığı yerdir. Burada toplanan sinyallerin seviyesi belli bir eşik değerinin üzerinde ise hücre çıkış sinyali üretir. Üretilen sinyal aksona yayılır.

Akson : Hücre çekirdeğinin ürettiği elektriksel çıkış sinyalini, diğer hücelere dağıtmakla görevlidir. Çıkış sinyali diğer hücrelerin dentritleri ile bağlantı noktası olan sinapsa iletilir.

Sinapslara gelen elektriksel sinyal, sinaps boşluğunda kimyasal yollar ile diğer hücrelerin dentritlerine iletilirler. Bu kimyasal iletim her bir dentritte farklı ağırlıklar ile tekrar elektrik sinyaline dönüşür. Bu ağırlıklar sayesinde, diğer hücrelerin dentritlerine gelen giriş sinyali daha baskın yada daha pasif bir seviyeye gelmektedir ve bilginin hangi nöronlar üzerinden iletileceği belirlenir. Beyin içerisindeki öğrenme de bu ağırlıkların güncellenmesi ile oluşmaktadır.

Bir yapay sinir hücresi de, doğal sinir hücresinin birimlerinden esinlenerek oluşturulmuştur. Hücrenin, bilgi sinyalini dışarıdan aldığı girişleri olan dentritler yapay modelde giriş birimini oluşturmaktadır. Dentritlerden alınan sinyallerin topladığı ve belli bir eşik seviyesinden sonra çıkışın üretildiği yer olan hücre çekirdeği toplam birimi olarak modellenir. Akson boyunca iletilen bilgi sinyalinin, diğer hücelere aktarılırken belli bir aktivasyon fonksiyonundan geçirildiği ve farklı ağırlıklar ile çarpıldığı yer olan sinapsların da modeli oluşturulduğunda (Şekil 2.2)' de bulunan yapay sinir ağı oluşturulmaktadır [20].



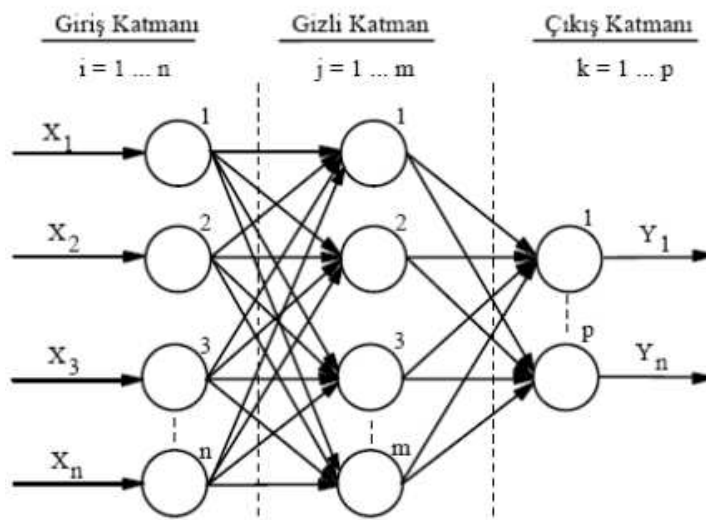
Şekil 2.2 Doğal nöronun modellenmesi ile oluşturulan yapay sinir hücresi

Basit birim kullanarak yapay sinir hücresinin (Şekil 2.2)' de ki gibi lineer olmayan modeli tasarlanabilir. Oluşturulan bu modelle yapay sinir hücreleri katmanlar şeklinde birleştirilerek

yapay sinir ağını oluştururlar. Her ne kadar bir yapay nöron basit bir yapıya sahip olsa da yapay sinir ağları gelişmiş karmaşık yapılardır.

2.1.1 Yapay Sinir Ağının Yapısı ve Özellikleri

YSA beynin yapısı göz önünde bulundurularak tasarlanmıştır. Yapay nöronların katmanlar şeklinde birbirine bağlanması ile oluşur. Her bir nöron bütün birimleri ile bir hücreyi oluşturur. Her katmandaki hücrelerin aktivasyon fonksiyonları birbirine eşittir. Bu katmanlar donanım olarak ya da yazılımsal olarak gerçekleştirilebilmektedir. Donanım gerçeklemeleri, YSA'nın paralel işleme özelliği ile hızlı ve verimli çalışmaktadır.



Şekil 2.3 Katmalar halindeki hücreler ile oluşturulan bir YSA [21]

Paralellik

YSA'nın yapısı, görevi tüm hücrelere dağıtılmış bir paralel işlemci yapısındadır. Her hücre kendi işlemini kendisi yapar. Aynı katmandaki hücreler eş zamanlı çalışır. Ayrıca diğer işlemcilerden farklı olarak ayrıca öğrenme özelliğine de sahiptir.

Öğrenme ve Genelleme

YSA'lar programlama yerine giriş verileri sayesinde öğrenirler. Eğitim işlemi eğitici ve öğrenci olmak üzere iki şekilde yapılabilmektedir. Eğitici öğrenme, uygulanan girişe karşı beklenen çıkışın YSA'nın çıkışında üretilinceye dek katmanlardaki ağırlıkların güncellenmesi işlemi olarak tanımlanmaktadır. YSA'nın çıkışı ile istenilen çıkış arasındaki hataya bağlı olarak öğrenme kuralı ile ağırlıklar güncellenir. Öğreticisiz öğrenme de ise ağ için belli bir hedef çıkış

yoktur. Ağ, giriş vektörlerine göre kendini örgütler. Genellikle sınıflandırma problemlerinde kullanılırlar [22].

Ayrıca YSA' lar, yerel bellek ve bilgi toplama özellikleri ile daha önceden öğretilen giriş verilerinden ayrı olarak genelleme yapabilir. Genelleme, öğrenme ve paralel işleme özellikleri sayesinde YSA' lar birçok zor problemin üstesinden gelebilmektedir. Özellikle sınıflandırma problemlerinde sıklıkla kullanılırlar.

Hata Toleransı

Ağın içerisindeki bir hücrenin çalışmaması durumunda ağın genel işleyişinde ve performansında ciddi bir düşüş olmaz. Diğer hücrelerin aktif çalışması ağ içerisinde oluşabilecek hataları tolere eder. Ayrıca sistem gürültülere karşı kararlı bir yapıdadır. Dışarıdan gelen gürültü tüm ağa dağıtıldığında etkisi minimize edilir. Bu sayede daha gürbüz sistemler tasarlanabilmektedir.

Uyarlanabilme

YSA belirtilen özelliklerinin yanında doğrusal yapılar değildir. Bu özelliği sayesinde karşılaşılan karmaşık problemler kolaylıkla YSA ile çözülebilmektedir. Bu özellikler neticesinde YSA' nın uygulama alanı geniş bir yelpazeyi kapsar.

2.1.2 Yapay Sinir Ağlarının Çeşitleri

Yapay sinir ağları yapısını oluşturduğu hücrelerin matematiksel modeline göre iki sınıf altında incelenebilir.

- Statik Hücre Modeli
- Dinamik Hücre Modeli

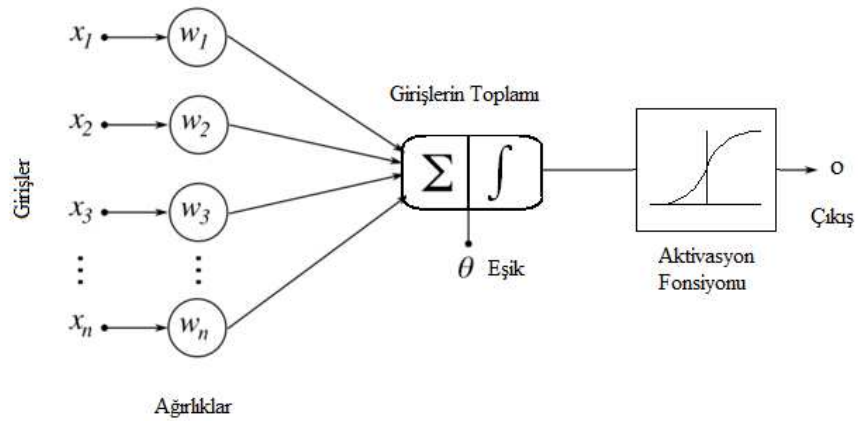
Statik hücre modeli YSA yapılarında işlemler çarpma, toplama ve aktivasyon fonksiyonundan geçirme şeklindedir. Hücreler doğrusal ya da doğrusal olmayan denklem takımları tarafından modellenebilir.

$$v = \sum_{i=1}^n W_i \cdot x_i + \theta_i \quad y = \varphi(v) \quad (2.1)$$

(2.1)' de statik hücrenin matematiksel modeli verilmiştir. Girişler W_i ağırlıkları ile çarpılıp toplandıktan sonra toplam θ_i bias değeri ile toplanır. Çıkış işareti bu toplamın aktivasyon fonksiyonundan geçmesi ile olur. Bir statik hücre modeli (Şekil 2.2)' de verilmiştir.

Bu denklem entegral ve türev ifadeleri içermediği için statik bir matematiksel yapıya sahiptir. Sınıflandırma ve örüntü tanıma uygulamalarında sıklıkla kullanılır. Dinamik sistemlerin modellerinin oluşturulması ve dinamik problemlerin çözümü için dinamik hücre modeline ihtiyaç duyulmaktadır.

Dinamik hücreler, bünyesinde geçici rejimlerin olduğu yapılarıdır. Bu hücrelerin içerisinde entegral birimi de bulunmaktadır. Hücre içindeki entegral biriminin devreler ile gerçekleştirilmesi istendiğinde RC devreleri kullanılabilir.



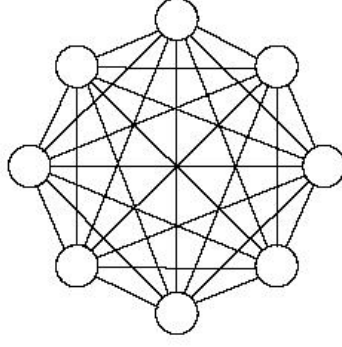
Şekil 2.4 Dinamik Hücre Modeli

Dinamik hücre modeli ile gerçekleştirilen ağlara Hopfield ve Hüresel Sinir Ağları (HSA) örnek olarak verilebilir.

2.1.3 Hopfield Sinir Ağları

Hopfield 1982 yılında, YSA' nın içerisindeki nöronların her birinin birbirleri ile etkileşimine dayanan yeni bir ağ modeli geliştirdi. Bu yeni ağ modeli, birinci dereceden doğrusal olmayan minimum bir enerji seviyesine yakınsayan diferansiyel denklem takımları ile tasarlanmıştır. Bu ağ yapılarında öğrenme, ağın içerisindeki sinaptik katsayıların, enerji düzeyini minimuma doğru yaklaştıracak şekilde güncellenmesi ile olmaktadır. Enerjinin minimuma indiği seviyede öğrenme işlemi tamamlanmış demektir. Bu fikirden hareketle Hopfield '**enerji kuyuları**' kavramını ortaya atmıştır.

Hopfield sinir ağlarının ağ topolojileri de diğer YSA' lardan farklıdır. Her hücre birbirleri ile bağlantılıdır ve veri bir bağlantı üzerinden çift yönlü akmaktadır. Ayrıca Hopfield sinir ağları sadece tek bir katmandan oluşmaktadır. Katman içerisindeki her hücrenin yapısı aynıdır.



Şekil 2.5 Hopfield sinir ağına bir örnek

Hopfield sinir ağları dinamik yapıları sayesinde birçok karmaşık problemi çözebilmektedirler. Ayrıca dinamik sistemlerin ve beynin çağrışimli bellek yapısının modellenmesinde kullanılmaktadırlar.

Ancak her bir hücrenin birbirleri ile bağlantı yapması ağın karmaşıklığını artırmaktadır. Böylesine karmaşık bir ağın tümleşik devreler ile gerçekleştirilmesi bağlantı yollarının çokluğundan dolayı çok zordur. Ayrıca güç tüketimi de oldukça yüksektir.

2.2 Hücresel Sinir Ağları

Hopfield ağlarında karşılaşılan bağlantı sorununu ortadan kaldırmak için Leon O. Chua ve Lin Yang 1988 yılında yeni bir yapay sinir ağı mimarisi ortaya koydular. Bu mimarideki yerel bağlantı fikri ile Hücresel Sinir Ağları (HSA) Hopfield ağlarından türetilmiş oldu.

HSA, birbirine yerel komşuluklar ile bağlı ve genellikle iki boyutlu hücrelerden meydana gelmiş olan yapay sinir ağlarıdır. Bu yapının Hopfield ağlarından ayıran en önemli özelliği, Hopfield ağlarında ki gibi her hücrenin diğer tüm hücreler ile bağlantılı olması yerine, hücrelerin birbirleri ile bağlantılarının yerel komşuluk tanımına göre yapılmış olmasıdır [20].

Az bağlantı yapısı sayesinde güç tüketiminde de önemli ölçüde tasarruf sağlanmıştır. Bu yapının diğer önemli bir avantajı ise sistemin çıkış değerlerine yakınsama süresi devrenin boyutundan bağımsız olup gecikmeler birkaç nöronun işlem süresi kadardır. Ayrıca bağlantılar arasındaki katsayıların tüm ağ boyunca sabit kalması da diğer önemli avantajlarından biri sayılmaktadır [22].

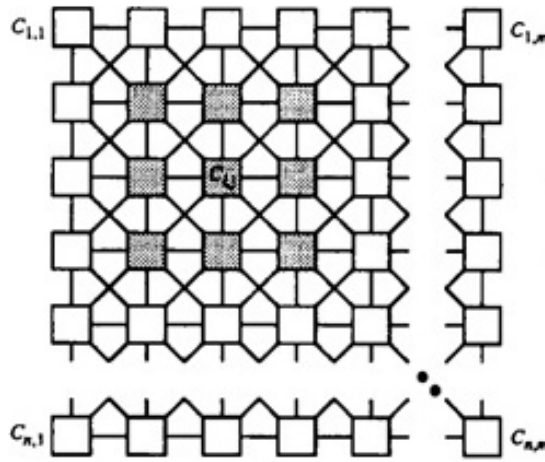
HSA'ların iki boyutlu yapısı görüntü işleme uygulamalarına elverişlidir. İki boyutlu yapısından dolayı görüntüler giriş olarak hücrelere verilmektedir. İlk giriş değerleri uygulandığında hücreler üzerinde geçici rejimler oluşur. Hücrelerin yakınsadığı değerler sistemin çıkışı

olmaktadır. HSA birçok ön işleme probleminin yanında video işaretlerinde hareket sezme ve görüntülerden öznitelik çıkartımında kullanılmaktadır. HSA uygun tasarımlar altında öznitelik çıkarıcı olarak kullanılabilir [22].

Genel olarak bir HSA' nın görüntü işlemedeki fonksiyonu, giriş görüntüsünü belli bir amaca yönelik çıkış görüntüsüne dönüştürmek olarak tanımlanabilir.

2.2.1 Hücresel Sinir Ağının Mimarisi

2-B (iki boyutlu) hücresel sinir ağları, hücre adı verilen birimlerden oluşmuş analog devrelerdir [23]. İki boyutlu dizi oluşturacak şekilde sıralanmış hücreler M satır ve N sütun olmak üzere MxN boyutlarında kartezyen koordinat sistemine benzer bir yapıyı meydana getirirler.



Şekil 2.6 Kartezyen HSA mimarisi [24]

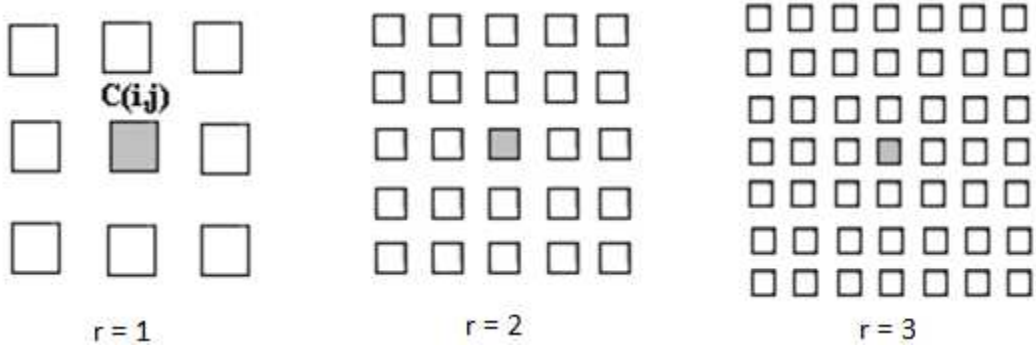
(Şekil 2.6)' te verilen HSA yapısında i. satırda ve j. sütundaki bir hücre C(i,j) olarak tanımlanmaktadır. Ayrıca her bir hücre komşusu olan yedi hücre ile bağlantılıdır. Diğer hücreler ile de komşu hücreler üzerinden dolaylı bir bağlantıya sahiptir.

2.2.2 Komşuluk Tanımı

HSA içerisindeki bir hücrenin r yarıçaplı komşuları $S_r(i,j)$ (2.1)'de ki gibi tanımlanabilir.

$$S_r(i,j) = \{C(k,l) \mid \max\{|k-i|, |l-j|\} \leq r, 1 \leq k \leq M, 1 \leq l \leq N\} \quad (2.1)$$

C(i,j) hücrelerine r yarıçaplı bir mesafe dahilinde bulunan tüm hücreler C(i,j) hücrelerinin komşusudur. C(i,j) hücrelerinin sadece bu komşuları ile bağı vardır. $S_r(i,j)$, r yarıçaplı etki küresi olarak adlandırılmaktadır.



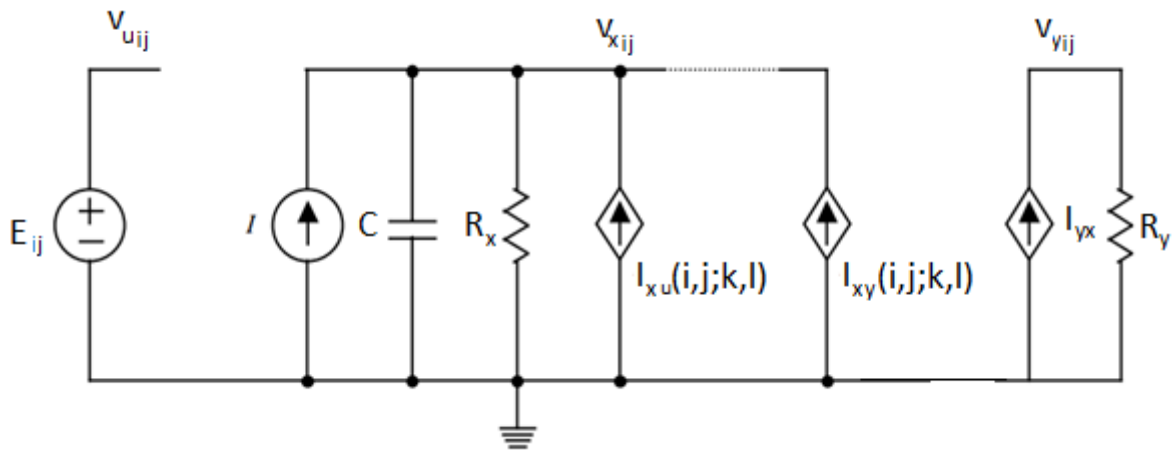
Şekil 2.7 Farklı r değerleri için $C(i,j)$ 'ye komşu hücreler

Görüntü işleme uygulamalarında genellikle $r = 1$ olarak seçilir. r değeri belirlendikten sonra hücreler, belirtilen r değeri içindeki komşular ile bağlantı kurarlar.

HSA içerisindeki hücrelerin devre yapısı, doğrusal direnç, kondansatör ve doğrusal veya doğrusal olmayan bağımlı/bağımsız kaynaklar ile modellenebilmektedir [22].

2.2.3 Hücrenin Devre Yapısı ve Matematiksel Modeli

HSA yapısındaki bir hücrenin gerçekleştirildiği devresel eşdeğer (Şekil 2.8)'de gösterilmiştir. Hücresel sınır ağındaki bir $C(i,j)$ hücreye karşılık gelen bu devrede $v_{u_{ij}}$ giriş, $v_{x_{ij}}$ durum ve $v_{y_{ij}}$ çıkışa olarak tanımlanmıştır. $C(i,j)$ hücreye ait $v_{x_{ij}}$ durum geriliminin başlangıç koşulu ile giriş gerilimi olan $v_{u_{ij}}$ 'nin değerleri $[-1,1]$ arasında kabul edilir. Görüntü işlemede de piksel değerleri hücrelere giriş olarak uygulanmadan önce $[-1,1]$ değerlerine normalize edilir.



Şekil 2.8 HSA içerisindeki bir hücrenin devre yapısı

Hücre içerisindeki devre elemanları, $v_{u_{ij}}$ giriş gerilimini oluşturan bağımsız gerilim kaynağı E_{ij} ; hücrenin $v_{x_{ij}}$ durumuna eklenen bias olarak bağımsız akım kaynağı I ; hücre üzerinde geçici rejimlerin oluşması için kullanılan doğrusal C kondansatör ve R_x direnç elemanları; hücreye,

komşu hücrelerin girişlerinden ve komşu hücrelerin üzerlerinde oluşan gerilimden gelen katkıların gerçekleşmesi için sırası ile $I_{xu}(i,j;k,l)$ ve $I_{xy}(i,j;k,l)$ bağımlı akım kaynakları ve çıkış gerilimi v_{yij} ' yi oluşturmak için kullanılan doğrusal olmayan gerilim kontrollü akım kaynağı I_{yx} ve doğrusal direnç elemanı R_y ' dir.

$C(i,j)$ hücresine komşu olan hücrelerin sayısı n ise, komşu hücrelerden gelen katkılar için kullanılan bağımlı akım kaynağı sayısı $2n$ olur.

Komşu hücrelerin girişlerinden $C(i,j)$ hücresine olan katkı;

$$I_{xu}(i,j;k,l) = B(i,j;k,l)v_{u_{kl}} \quad , \quad \forall C(k,l) \in S_r(i,j)$$

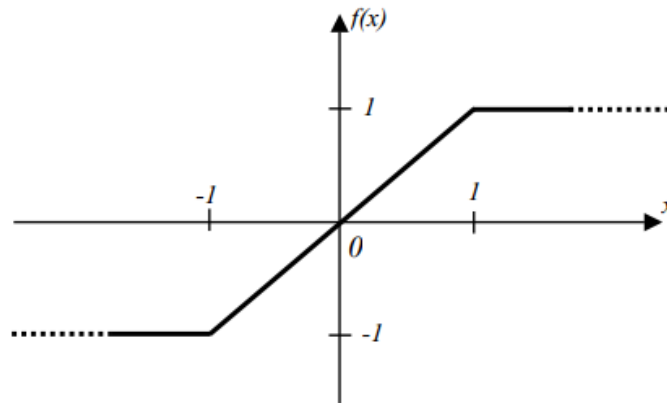
Komşu hücrelerin üzerinde oluşan gerilimden gelen katkı;

$$I_{xy}(i,j;k,l) = A(i,j;k,l)v_{y_{kl}} \quad , \quad \forall C(k,l) \in S_r(i,j)$$

Devrenin çıkış gerilimi (v_{yij}) ise (2.2)' de ki doğrusal olmayan akım kaynağı gerçekleşmektedir.

$$I_{yx} = \frac{1}{R_y} \cdot f(v_{x0}) \quad (2.2)$$

Çıkış için tanımlanan $f(.)$ fonksiyonu (Şekil 2.9)' de gösterildiği gibi parçalı doğrusal bir fonksiyondur. Bu sayede çıkış gerilim değerleri $[-1,1]$ sınırları içerisinde kalır.



Şekil 2.9 Hücrenin çıkış fonksiyonu

Devre üzerindeki v_{xij} durum geriliminin bulunduğu düğüm üzerinden Kirchoff akımlar yasası kullanılarak devrenin eşitliği aşağıdaki gibi türetilebilir.

Devre üzerindeki durumun eşitliği;

$$C \cdot \frac{dv_{x_{ij}}(t)}{dt} = -\frac{1}{R_x} \cdot v_{x_{ij}}(t) + \sum_{C(k,l) \in N_r(i,j)} A(i,j;k,l) \cdot v_{y_{kl}}(t) + \sum_{C(k,l) \in N_r(i,j)} B(i,j;k,l) \cdot v_{u_{kl}}(t)$$

$$1 \leq i \leq M ; 1 \leq j \leq N \quad (2.3)$$

Devrenin çıkış eşitliği;

$$v_{y_{ij}} = 0.5(|V_{x_{ij}}(t) + 1| - |V_{x_{ij}}(t) - 1|) \quad (2.4)$$

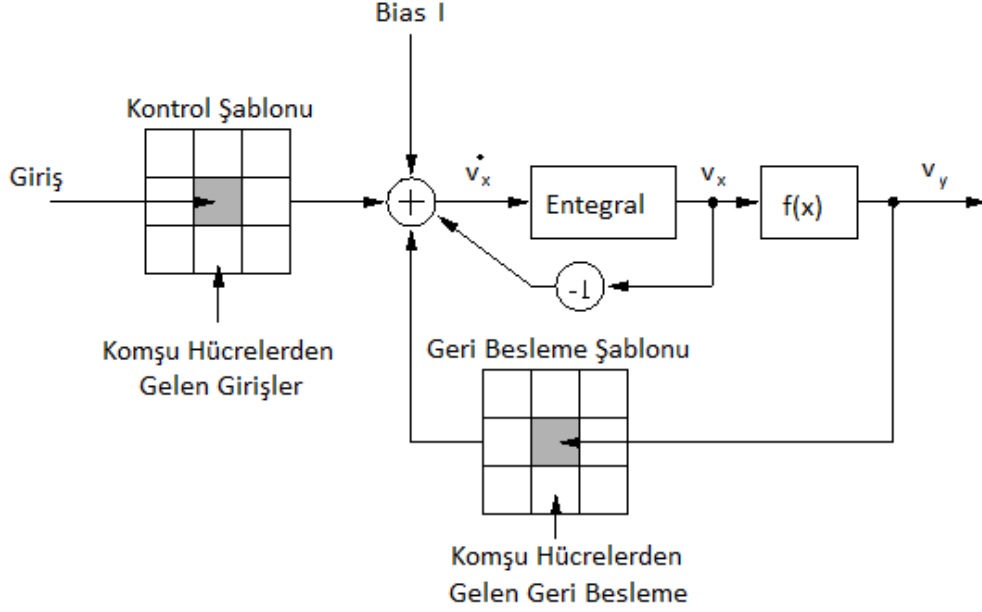
olarak ifade edilebilir.

Devrenin çalışması özet olarak (2.3)' ün içerdiği diferansiyel ifade sayesinde dinamik bir yapıdadır. Hücre içerisindeki kapasite devrenin bellek elemanıdır.

Buraya kadar olan bilgiler doğrultusunda HSA' nın yapısını ve çalışma şeklini şu şekilde özetleyebiliriz. HSA, hücre denilen devresel yapıların, iki boyutlu bir düzlemde r etki küresi içerisindeki komşu hücreler ile bağlantıları ile oluşan ızgara yapısındaki dinamik yapay sinir ağlarıdır. Tüm hücre C(i,j) merkez hücreleri aynı devre yapısına sahiptir. Her hücre doğrusal olmayan denklem takımı ile tanımlanır.

Hücre içerisindeki kondansatör ve direnç elemanları hücrenin özünü teşkil eder. Kondansatör elemanı sayesinde hücreler dinamik ve hafızalı bir yapı kazanırlar. Girişlere değeri [-1,1] arasında olan E_{ij} gerilim kaynağı uygulanarak hücreler üzerinde giriş gerilimi $v_{u_{ij}}$ oluşturulur. C(i,j) merkez hücresi için B(i,j;k,l) giriş operatörü (ileri klonlama şablonu) ile etki küresi içerisindeki hücrelerin girişlerinden katkı gelir. A(i,j;k,l) Geri besleme operatörü (geri klonlama şablonu) ile çıkışlar üzerinden C(i,j) hücrelerine katkı gelir. Bağımsız akım kaynağı ile gerçekleştirilen I bias akımı da eklenerek hücrenin üzerindeki durum oluşur.

Yazım kolaylığı açısından bundan sonraki ifadelerde giriş gerilimi v_u için \mathbf{u} , v_x durumu için \mathbf{x} , v_y çıkışı için \mathbf{y} ve I bias değeri için de \mathbf{z} ifadeleri kullanılacaktır.



Şekil 2.10 Bir hücrenin blok diyagramı [20]

Hücre içerisindeki $B(i,j;k,l)$ ileri klonlama şablonu, $A(i,j;k,l)$ geri klonlama şablonu ve I_{ij} bias akım değeri ağ içerisindeki her hücre için aynı yapıda ise konumdan ve zamandan bağımsız bir özelliktedir [20]. Uygulamaların çoğunda bu tip HSA yapıları kullanılmaktadır.

2.2.4 Klonlama Şablonlarının Matematiği

Klonlama şablonu olarak tanımlanan $A(i,j;k,l)$ ve $B(i,j;k,l)$ operatörler, $C(i,j)$ merkez hücresine etki küresi içerisindeki (k,l) indisli hücrelerin çıkış ve girişlerinden gelen katkıları tanımlar. HSA uygulamalarının çoğunda $r = 1$ olarak seçilir. Bu hem matematiksel analizi kolaylaştırır hem de devrenin VLSI tasarımlara daha elverişli olması sağlar.

İleri ve geri besleme operatörü ve z_{ij} bias akımı her hücre için aynı yapıda olduklarından dolayı şablon özelliği kazanırlar.

Geri besleme sinaptik operatörü $A(i,j;k,l)$ ' den gelen katkı;

$$\begin{aligned}
 \sum_{C(k,l) \in S_r(i,j)} A(i,j;k,l) y_{kl} &= \sum_{|k-i| \leq 1} \sum_{|l-j| \leq 1} A(k-i, l-j) y_{kl} \\
 &= a_{-1,-1} \cdot y_{i-1,j-1} + a_{-1,0} \cdot y_{i-1,j} + a_{-1,1} \cdot y_{i-1,j+1} + a_{0,-1} \cdot y_{i,j-1} \\
 &+ a_{0,0} \cdot y_{i,j} + a_{0,1} \cdot y_{i,j+1} + a_{1,-1} \cdot y_{i+1,j-1} + a_{1,0} \cdot y_{i+1,j} + a_{1,1} \cdot y_{i+1,j+1} \\
 &= \sum_{k=-1}^1 \sum_{l=-1}^1 a_{k,l} \cdot y_{i+k,j+l}
 \end{aligned} \tag{2.5}$$

(2.5) ifadesi şablon olarak gösterilirse;

$$\begin{array}{|c|c|c|} \hline a_{-1,-1} & a_{-1,0} & a_{-1,1} \\ \hline a_{0,-1} & a_{0,0} & a_{0,1} \\ \hline a_{1,-1} & a_{1,0} & a_{1,1} \\ \hline \end{array} \otimes \begin{array}{|c|c|c|} \hline y_{i-1,j-1} & y_{i-1,j} & y_{i-1,j+1} \\ \hline y_{i,j-1} & y_{i,j} & y_{i,j+1} \\ \hline y_{i+1,j-1} & y_{i+1,j} & y_{i+1,j+1} \\ \hline \end{array} = A \otimes Y_{ij} \quad (2.6)$$

$$A = \begin{array}{|c|c|c|} \hline a_{-1,-1} & a_{-1,0} & a_{-1,1} \\ \hline a_{0,-1} & a_{0,0} & a_{0,1} \\ \hline a_{1,-1} & a_{1,0} & a_{1,1} \\ \hline \end{array} \quad Y_{ij} = \begin{array}{|c|c|c|} \hline y_{i-1,j-1} & y_{i-1,j} & y_{i-1,j+1} \\ \hline y_{i,j-1} & y_{i,j} & y_{i,j+1} \\ \hline y_{i+1,j-1} & y_{i+1,j} & y_{i+1,j+1} \\ \hline \end{array}$$

Y_{ij} , $C(i,j)$ merkez hücrenin çıkış görüntüsüdür. $M \times N$ çıkış görüntüsü üzerinde 3×3 ' lük bir maske dolaştırılarak oluşturulur. (2.6)' da A geri besleme şablonu ile Y_{ij} çıkış görüntüsü şablon nokta çarpımı ile çarpılmıştır. Bu çarpma uzamsal düzlemde konvolüsyon işlemine karşılık gelmektedir [24].

İleri besleme sinaptik operatörü $A(i,j;k,l)$ ' den gelen katkı;

$$\begin{aligned} \sum_{C(k,l) \in S_r(i,j)} B(i,j;k,l) y_{kl} &= \sum_{|k-i| \leq 1} \sum_{|l-j| \leq 1} B(k-i, l-j) u_{kl} \\ &= \sum_{k=-1}^1 \sum_{l=-1}^1 b_{kl} u_{i+k, j+l} \end{aligned}$$

$$\begin{array}{|c|c|c|} \hline b_{-1,-1} & b_{-1,0} & b_{-1,1} \\ \hline b_{0,-1} & b_{0,0} & b_{0,1} \\ \hline b_{1,-1} & b_{1,0} & b_{1,1} \\ \hline \end{array} \otimes \begin{array}{|c|c|c|} \hline u_{i-1,j-1} & u_{i-1,j} & u_{i-1,j+1} \\ \hline u_{i,j-1} & u_{i,j} & u_{i,j+1} \\ \hline u_{i+1,j-1} & u_{i+1,j} & u_{i+1,j+1} \\ \hline \end{array} = B \otimes U_{ij} \quad (2.7)$$

$$B = \begin{array}{|c|c|c|} \hline b_{-1,-1} & b_{-1,0} & b_{-1,1} \\ \hline b_{0,-1} & b_{0,0} & b_{0,1} \\ \hline b_{1,-1} & b_{1,0} & b_{1,1} \\ \hline \end{array} \quad U_{ij} = \begin{array}{|c|c|c|} \hline u_{i-1,j-1} & u_{i-1,j} & u_{i-1,j+1} \\ \hline u_{i,j-1} & u_{i,j} & u_{i,j+1} \\ \hline u_{i+1,j-1} & u_{i+1,j} & u_{i+1,j+1} \\ \hline \end{array}$$

U_{ij} , merkez $C(i,j)$ hücrenin giriş görüntüsüdür. $M \times N$ giriş görüntüsü üzerinde 3×3 ' lük bir maske dolaştırılarak oluşturulur. (2.7)' de B ileri besleme şablonu ile U_{ij} giriş görüntüsü nokta çarpım ile çarpılmıştır.

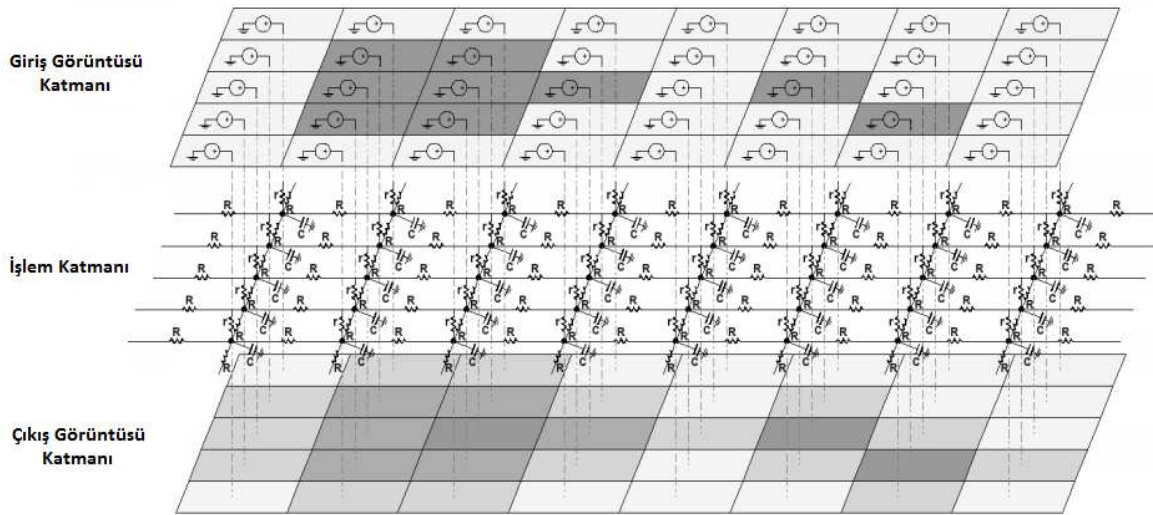
2.3 Hücresel Sinir Ağları İle Görüntü İşleme

HSA ile örüntü tanıma, hareket sezme, nesne sayma ve yol takibi gibi ileri düzeyde görüntü işleme uygulamaları gerçekleştirilebilmektedir [25]. Özellikle yüksek çözünürlükteki büyük

boyutlu resimlerin işlenmesi, analizi ve bu görüntülerden öznitelik çıkarımı gibi uygulamalar çok yaygındır. Büyük görüntülerin işlenmesi yüksek maliyetler getirir. Yapılan işlem süresi daha uzun olmakla beraber görüntü işleme için kullanılan sistemlerin maliyetleri de yüksek olmaktadır.

Hücreyel sinir ağları yapısı itibari ile resmin boyutundan bağımsız yüksek hızlı görüntü işlemeye elverişlidir. Devreye giriş görüntüsü uygulandıktan sonra devrenin zaman sabiti olan R.C kadar bir gecikmeden sonra çıkış görüntüsü elde edilebilmektedir. Geliştirilen şablonlar ile görüntü işlemede kullanılan birçok algoritma HSA' lar tarafından daha verimli bir şekilde gerçekleştirilebilmektedir.

Her hücre $M \times N$ boyutlarında ki ayrık uzamsal görüntünün bir pikseline karşılık gelir. Piksel değerleri $[-1,1]$ arasına normalize edildikten sonra ağ içerisindeki hücelere giriş gerilimi olarak verilir. Ağ içerisindeki dinamik yapıdaki hüceler, uygulanan gerilime tepki üretir. Hücreler üzerinde geçici rejimler oluşur. Geçici rejimlerden sonra hüceler üzerinde oluşan kararlı durumda elde edilen çıkışlar, girişin işlenmesi ile elde edilen çıkış görüntüsünü oluşturur. (Şekil 2.11) ' da HSA ile bir görüntü işleme modeli gösterilmiştir.



Şekil 2.11 HSA ile görüntü işleme modeli [26]

Görüntü işleme uygulamaları için HSA analog çipleri mevcuttur. Bu çipler eş zamanlı hızlı görüntü işleme için uygundur. Ayrıca devre gerçeklemesi itibariyle görüntü işleme için ADC ve DAC yapılarına ihtiyaç duymaz. Fakat bu çipler, maliyeti ve tasarım karmaşıklığı açısından pek tercih edilmemektedir.

HSA' nın denklemleri doğrusal ve doğrusal olmayan denklem takımı olarak ikiye ayrılabilir. Görüntü işleme uygulamaların da sıklıkla doğrusal HSA denklemleri kullanılmaktadır.

2.3.1 Doğrusal HSA İle Görüntü İşleme

HSA denklem takımında bulunan (2.4)' te ki çıkış fonksiyonunun parçalı doğrusal olması nedeni ile HSA yapısı doğrusal değildir. $f(\cdot)$ aktivasyon fonksiyonu $[-1,1]$ aralığında doğrusal diğer bölgelerde sabittir. Eğer aktivasyon fonksiyonu (2.8)' de ki gibi seçilirse hücrelerin kararlı haldeki durumları çıkış görüntüsüne eşittir.

$$y_{ij}(x_{ij}) = x_{ij} \quad (2.8)$$

Doğrusal ve konumdan bağımsız bir HSA denklemi $r = 1$ için (2.9)' de ki gibi yazılır.

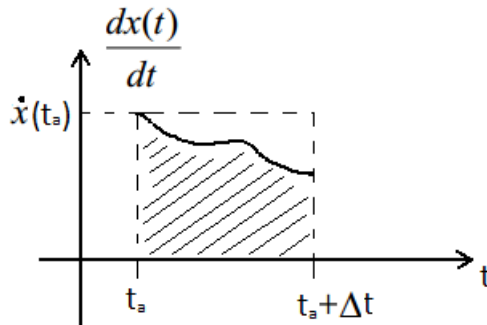
$$\frac{dx_{ij}}{dt} = -x_{ij} + \sum_{k=-1}^1 \sum_{l=-1}^1 A(k, l)x_{i+k, j+l} + \sum_{k=-1}^1 \sum_{l=-1}^1 B(k, l)u_{i+k, j+l} + z_{ij} \quad (2.9)$$

(2.9)' de ki eşitlik sürekli zamanlı durumlar için tanımlanmıştır. Diferansiyel denklemleri nümerik biçimde sayısal ortamlarda iteratif olarak hesaplayabilmek için fark denklemleri haline getirmek gerekir. Türev ifadesi için Euler ileri yaklaşıklığı kullanılarak fark denklemleri elde edilebilir. Hücre çıkışlarının yakınsadığı değerler sistemin çıkışı olur.

2.3.2 HSA Denkleminin Nümerikleştirilmesi

Diferansiyel denklemlerin sayısal ortamlarda simüle edilebilmesi için nümerik bir yapıda olması gerekir. Türev ifadesi fark denklemine dönüştürüldüğünde ayrık zamandaki fark denklemleri iteratif olarak hesaplanabilir.

Euler ileri yaklaşıklığı integral ifadesinin dikdörtgen yaklaşıklığı ile hesaplanmasından yola çıkarak elde edilmiştir.



Şekil 2.12 İntegrale dikdörtgen yaklaşıklığı

(Şekil 2.12)' de ki taralı alan integral üzerinden hesaplanabilir. Ayrıca Δt yeterince küçük olduğu kabul edilirse taralı alanın, üzerini çevreleyen dikdörtgenin alanına yakın bir değeri vardır. Δt değeri ne kadar küçükse integral ile hesaplanan alan ile dikdörtgenin alanı arasındaki fark o derece azalır. Bu durum altında integral alan hesabı ile dikdörtgen alan hesabı (2.11)' de ki gibi verilebilir.

$$\int_{t_a}^{t_a+\Delta t} \left[\frac{dx(t)}{dt} \right] \cdot dt \cong \Delta t \cdot \left[\frac{dx(t)}{dt} \right] \Big|_{t_a} \quad (2.10)$$

(2.10) düzenlendiğinde türev ifadesi (2.11)' de ki gibi ayırık bir yapıya dönüşür.

$$\frac{dx(t)}{dt} \cong \frac{x(t+\Delta t) - x(t)}{\Delta t} \quad (2.11)$$

(2.11)' de bulunan ayırık türev ifadesi HSA diferansiyel denkleminde (2.12) elde edilir.

$$\begin{aligned} \frac{dx(t)}{dt} \cong \frac{x(t+\Delta t) - x(t)}{\Delta t} = \\ -x_{ij}(t) + \sum_{k=-1}^1 \sum_{l=-1}^1 A(k, l) y_{i+k, j+l}(t) + \sum_{k=-1}^1 \sum_{l=-1}^1 B(k, l) u_{i+k, j+l}(t) + z_{ij} \end{aligned} \quad (2.12)$$

Adım aralığı, sürekli zamanlı işaretleri örnekleme periyodudur. Bu durum altında $\Delta t = T_s$ olarak kabul edilirse (2.13) yazılabilir. ($x'_{ij}(t)$) ifadesi ($x_{ij}(t)$) ifadesinin türevidir.

$$x'_{ij}(t) \equiv x'_{ij}(n \cdot T_s) \Big|_{t=n \cdot T_s} \quad (2.13)$$

Örnekleme süresi belirlendikten sonra (2.12) üzerinden (2.14) elde edilir.

$$\begin{aligned} \frac{x_{ij}((n+1)T_s) - x_{ij}(nT_s)}{T_s} = -x_{ij}(nT_s) + \sum_{k=-1}^1 \sum_{l=-1}^1 A(k, l) y_{i+k, j+l}(nT_s) \\ + \sum_{k=-1}^1 \sum_{l=-1}^1 B(k, l) u_{i+k, j+l}(nT_s) + z_{ij} \end{aligned} \quad (2.14)$$

(2.14) ifadesi, iterasyon ile hesaplamaya uygun hale getirmek için (2.15) şeklinde düzenlenebilir.

$$\begin{aligned} x_{ij}((n+1)T_s) = (1 - T_s) x_{ij}(nT_s) \\ + T_s \left(\sum_{k=-1}^1 \sum_{l=-1}^1 A(k, l) y_{i+k, j+l}(nT_s) + \sum_{k=-1}^1 \sum_{l=-1}^1 B(k, l) u_{i+k, j+l}(nT_s) + z_{ij} \right) \end{aligned} \quad (2.15)$$

2.4 HSA ile Filtreleme

Hücrel sinir ağları, doğrusal olmayan yapısı sayesinde görüntü işleme alanında sıklıkla kullanılan işlemlerin gerçekleşmesinde kullanılmaktadır. Kenar tespiti, köşe tespiti, görüntülerin eşiklenmesinin yanı sıra ikili resimlerde lojik ve/veya, açma/kapama işlemlerinde de kullanılmaktadır. Ayrıca doğrusal HSA yapıları ile alçak, yüksek ve band geçiren filtre yapıları gerçekleştirilebilmektedir. Görüntü işlemede yoğun olarak kullanılan ve band geçiren filtre olan Gabor filtreleri de yine HSA ile gerçekleştirilebilmekte ve VLSI tasarımlara uygun bir hale getirilebilmektedir [28].

2.4.1 HSA ile Filtreleme Örnekleri

Yüksek Geçiren Filtre : A geri besleme klon şablonu ve B ileri besleme klon şablonu ve z bias değeri (2.16)'da ki gibi seçilirse doğrusal HSA yapısı giriş görüntüsü üzerinde kenar belirleme işlemi yapar.

$$A = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad B = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix} \quad z = \boxed{-0.5} \quad (2.16)$$



Şekil 2.13 HSA ile kenar belirleme uygulaması

Alçak Geçiren Filtre : Bir başka doğrusal HSA uygulaması ise alçak geçiren filtrelemedir. A ve B şablonları ile z bias değeri (2.17)'de ki gibi olursa HSA alçak geçiren filtreleme yapar.

$$A = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad B = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad z = \begin{bmatrix} 0 \end{bmatrix} \quad (2.17)$$



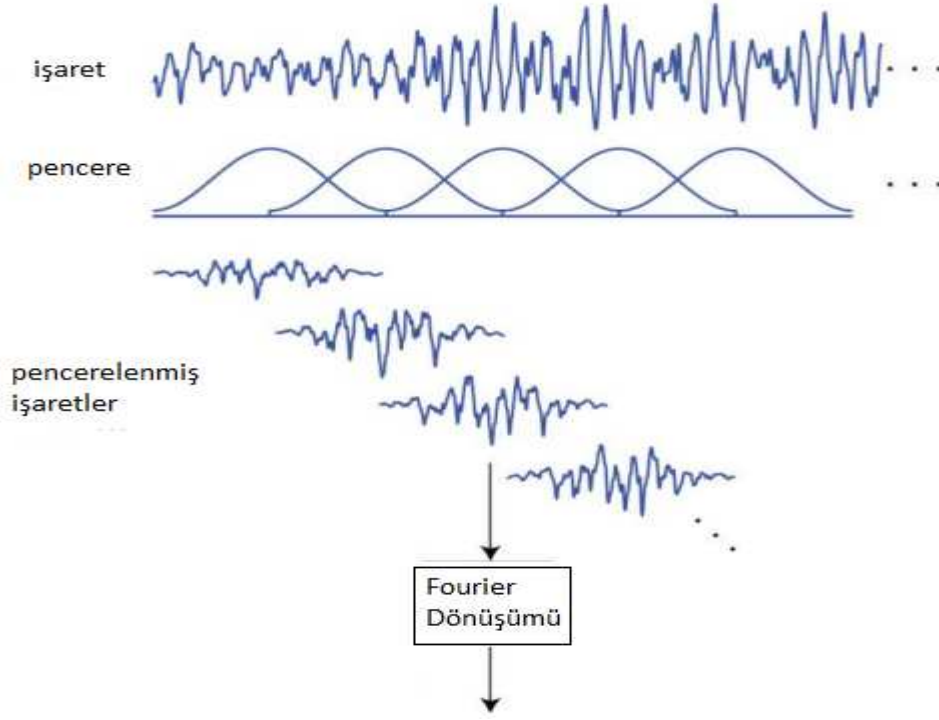
Şekil 2.14 HSA ile alçak geçiren filtreleme uygulaması

2.5 HSA İle Gabor Filtresi Tasarımı

Giriş

2-B işaretlerin analizinde Fourier ve Laplace dönüşümleri gibi birçok dönüşüm kullanılmaktadır. Bu dönüşümler ile işaretlerin bileşenleri incelenebilmekte ve içerdikleri bilgi işlenebilmektedir.

Bir dönüşümden istenen sistemler için genel bilginin çıkartılması olurken işaretler için lokal bilgilerin çıkartılması önem taşımaktadır [20]. İşaret işlemede yaygın olarak kullanılan Fourier dönüşümü global bir operatördür [29]. Bu nedenle işaretlerden lokal bilgilerin çıkartılmasında yetersiz kalmaktadır. Bunun için işaretlerin gösterimi için zaman-frekans analizleri geliştirilmiştir [30]. Zaman-frekans analizleri için Kısa Zamanlı Fourier Dönüşüm (KZFD) ("Short-time Fourier Transform") teknikleri geliştirilmiştir. Bu sayede işaretlerin lokal bilgileri çıkartılabilmıştır.



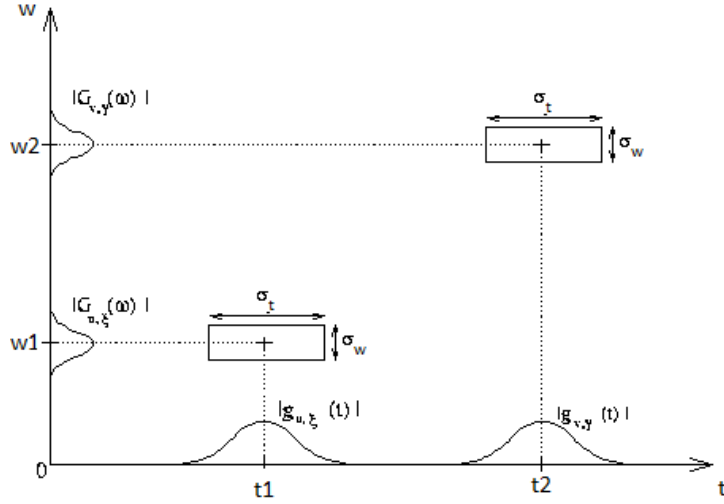
Şekil 2.15 Bir işaretin KZFD ile analizi

Zaman-frekans analizinde işaretin tamamının Fourier dönüşümünü almak yerine, işaret bir pencere fonksiyonunu ile parçalara ayrılarak her bir parçanın ayrı ayrı Fourier dönüşümü alınır. Bu sayede işaret içerisindeki frekans bileşenlerinin hangi zaman aralıklarında var oldukları bilgisi elde edilir (Şekil 2.15).

Zaman-frekans analizlerinde ölçekleme önemli bir problemdir. İşareti zaman düzleminde analiz ederken pencere fonksiyonunun olabildiğince dar olması istenir. Bu sayede daha lokal bilgiler çıkartılabilir. Fakat Fourier dönüşümünün (2.18)' de tanımlanan ölçekleme özelliğinden dolayı zaman düzleminde pencere fonksiyonunun daraltılması ile pencere fonksiyonunun band genişliği artmış olacak ve frekans düzleminde ki lokalliği azalacaktır. Bunun tersine eğer frekans düzleminde daha lokal bir analiz istenirse zaman düzlemindeki lokallikten ödün verilmesi gerekmektedir (Şekil 2.16).

$$f(x) \rightarrow F(\omega)$$

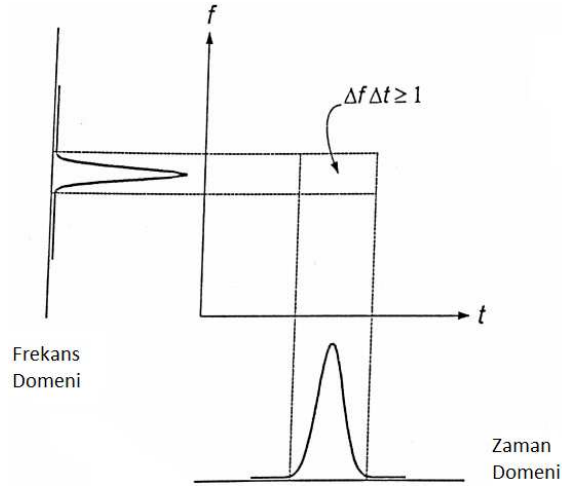
$$f(ax) \rightarrow \left| \frac{1}{a} \right| F\left(\frac{\omega}{a}\right) \quad (2.18)$$



Şekil 2.16 Zaman düzleminde pencerenin genişletilmesi ile Fourier düzlemindeki karşılığının daralması

Dennis Gabor, Gauss fonksiyonunu kullanarak geliştirdiği Gabor filtreleri ile işaretlerin hem zaman hem de frekans düzleminde minimum yerellik (lokallik) ile analiz edebilmiştir [31].

Gauss fonksiyonu fizikte ve işaretlerin analizinde sıklıkla kullanılmaktadır. Bir Gauss işaretinin Fourier dönüşümü yine bir Gauss işaretidir. Bu özelliği ile Fourier dönüşümünün öz fonksiyonu olarak bilinmektedir [32]. Zaman-frekans analizinde minimum belirsizlik Gauss fonksiyonu ile sağlanabilmektedir (Şekil 2.17).



Şekil 2.17 Bir işaretin zaman-frekans düzleminde olabilecek minimum belirsizliği

Gabor filtreleri bir boyutlu (1-B) işaretlerin analizinde güçlü bir matematiksel yöntemdir. İşaretler içerisinde öznitelik çıkartımı ve işaretlerin gürültüden temizlenmesi ve iyileştirilmesi gibi konularda birçok çalışma yapılmıştır.

2.5.1 (2-B) Gabor Filtreleri

Gabor filtrelerinden esinlenerek Daugman, iki boyutlu (2-B) Gabor filtrelerini tasarlamıştır [33]. 2-B Gabor filtreleri, uzamsal düzlemde 2-B Gauss işaretinin belli bir frekansı ve yönelimi olan kompleks sinüzoidal işaret ile modüle edilmesi ile oluşur. Fourier dönüşümünün modülasyon özelliğinden dolayı $g(x,y)$ Gauss işaretinin uzamsal fonksiyonu ve $G(\omega_x, \omega_y)$ Gauss işaretinin 2-B Fourier dönüşümü olmak üzere:

$$g(x, y) \cdot e^{j(\omega_{x0}x + \omega_{y0}y)} \longrightarrow G(\omega_x - \omega_{x0}, \omega_y - \omega_{y0})$$

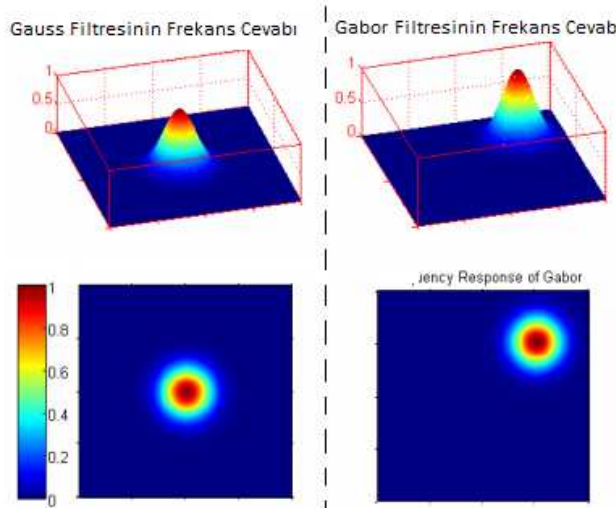
şeklinde olmaktadır.

2-B Gabor filtreler $(\omega_{x0}, \omega_{y0})$ merkez frekanslarına bağlı band geçiren filtrelerdir. Bir Gabor filtresinin uzamsal birim impuls cevabı $g(x,y)$ Gauss fonksiyonunun modüle edilmesi ile elde edilir.

$$h(x, y) = g(x, y) \cdot e^{j(\omega_{x0}x + \omega_{y0}y)}$$

$$g(x, y) = \frac{1}{2\pi\sigma^2} \cdot e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (2.19)$$

(2.19)' te ki ifade de $(\omega_{x0}, \omega_{y0})$ kompleks üsselin uzamsal frekansı ve σ Gauss fonksiyonunun standart sapmasıdır. (Şekil 2.18)' te Gauss ve Gabor filtrelerinin frekans cevapları görülmektedir.



Şekil 2.18 Gauss ve Gabor filtrelerinin frekans cevapları $(\omega_{x0} = \frac{\pi}{2}, \omega_{y0} = \frac{\pi}{2}, \sigma = 2)$

2-B Gabor filtrelerin görüntüler içerisinde yön seçici özelliği vardır. Bu özelliği ile görüntü işleme alanında sıklıkla kullanılmaktadır. Görüntüler içerisinde (2.20)' de belirtilen $\theta_{yön}$ açısı doğrultusunda yönelmiş kenarları seçmek için kullanılırlar.

$$\theta_o = \arctan\left(\frac{\omega_{xo}}{\omega_{yo}}\right) , \theta_{yön} = \theta_o + \frac{\pi}{2} \quad (2.20)$$

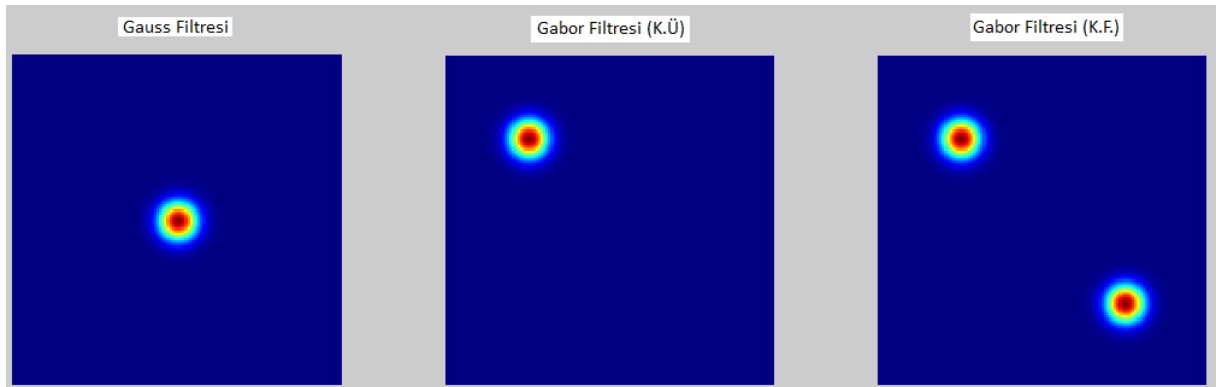
Gabor filtrelerinde kompleks üssel fonksiyon frekansları seçilecek olan yönü; Gauss fonksiyonunun standart sapması filtrenin band genişliğini (seçiciliğini) belirler.

Gauss işaretinin ve Gabor filtresinin Fourier dönüşümleri $G(\omega_x, \omega_y)$ ve $H(\omega_x, \omega_y)$ (2.21) ' de verilmiştir.

$$G(\omega_x, \omega_y) = e^{-\frac{\sigma^2(\omega_x^2 + \omega_y^2)}{2}}$$

$$H(\omega_x, \omega_y) = e^{-\frac{\sigma^2((\omega_x - \omega_{xo})^2 + (\omega_y - \omega_{yo})^2)}{2}} \quad (2.21)$$

Gabor filtreleri Gauss fonksiyonunun kompleks üssel fonksiyonu yerine kosinüs fonksiyonu ile module edilmesi ile de elde edilir (2.22). Kompleks üssel ile modüleden farklı olarak, frekans düzleminde ötelenmiş bir Gauss yerine merkez frekansları orijine göre simetrik olan ötelenmiş iki Gauss işareti bulunmaktadır (Şekil 2.19).



Şekil 2.19 Gauss Filtresi (sol), kompleks üssel (K.Ü) ile elde edilmiş Gabor filtresi (orta), kosinüs fonksiyonu (K.F.) ile elde edilmiş Gabor filtresi (sağ)

$$G(\omega_x, \omega_y) = e^{-\frac{\sigma^2(\omega_x^2 + \omega_y^2)}{2}}$$

$$H(\omega_x, \omega_y) = e^{-\frac{\sigma^2((\omega_x - \omega_{xo})^2 + (\omega_y - \omega_{yo})^2)}{2}} + e^{-\frac{\sigma^2((\omega_x + \omega_{xo})^2 + (\omega_y + \omega_{yo})^2)}{2}} \quad (2.22)$$

Gabor filtreleri (2.20),(2.21)' e göre yön seçici filtreleme yapmaktadır. Bir görüntü içerisinde Gabor filtresinin doğrultu açısında yönelmiş kısımlar filtrelenir. Özellikle örüntülerin yönelmiş kenar bilgisinin çıkartılmasında oldukça başarılıdır. Bu özelliği ile Gabor filtreleri iyi öznelik çıkarıcıdır. (Şekil 2.20)' de görüleceği üzere örüntülerin yönlü kenarları istenen doğrultular için çıkartılabilmektedir.



Şekil 2.20 Giriş görüntüsünden 135° açısında yönelmiş bileşenlerin çıkartılması
 $(\omega_x = \pi/4, \omega_y = \pi/4)$

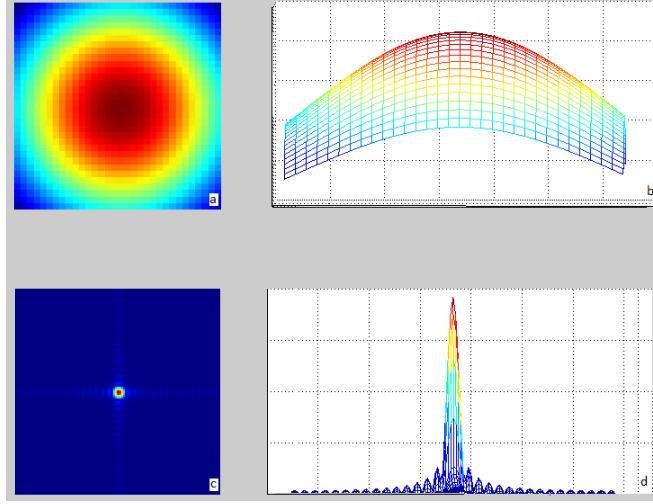
Gabor filtrelerin birçok avantajının yanında hesaplanma maliyeti yönünden büyük bir dezavantajı vardır. 2-B Gabor filtreleri, 2-B Gauss fonksiyonları üzerinden türetilmiştir. Gauss fonksiyonunun standart sapması filtre şablonunun boyutlarını belirlemektedir. Fourier dönüşümünün (2.18)' de verilen ölçekleme özelliği göz önünde bulundurulursa dar bandlı filtreler için uzamsal düzlemde geniş bir standart sapma (σ) kullanılması gerekmektedir.

Dar bandlı filtreler, örüntülerin alçak frekans bölgesindeki bilgilerin çıkartılmasında yoğun olarak kullanılmaktadır. Kullanılan saha için sayısal tasarımlar önemlidir. Fakat uzamsal düzlemde σ ' nın büyük olması filtre şablonunun boyutlar ($W \times W$) değerlerinin de büyük olmasını gerekmektedir (2.23a) (2.23b) [34]. Bu da FIR Gabor filtreleri tasarlanırken daha fazla çarpma ve toplama işleminin yapılacağı anlamına gelmektedir [34].

$$W \geq [6. \sigma + 1] \quad (2.23a)$$

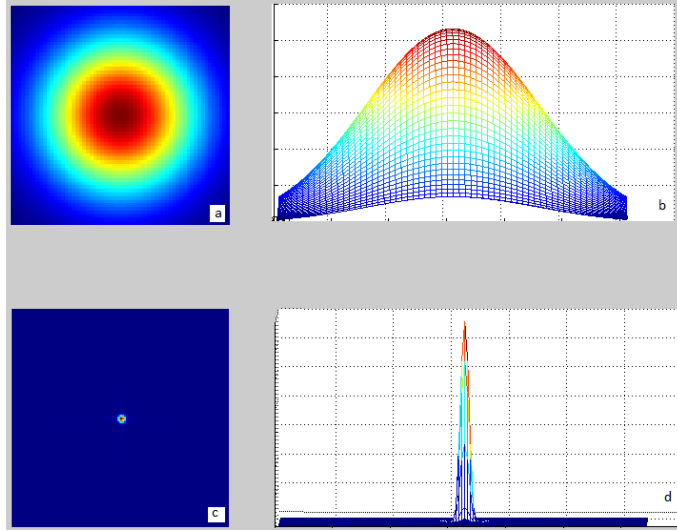
$$W \geq [4. \sigma + 1] \quad (2.23b)$$

Seçilecek olan şablon boyutu Gauss işaretinin %100' e yakın bir enerjisini temsil edebilmesi gerekir. Eğer şablon boyutu küçük seçilirse Gauss işareti bütün olarak şablona sığmaz ve kare bir fonksiyon ile kesilmiş gibi olur. Bu durum frekans bölgesinde pencereleme (windowing) etkisine yol açar (Şekil 2.21). Bu problemin olmaması için W değeri (2.23a)' ya göre olmalıdır [34].



Şekil 2.21 Düşük değerli W için pencereleme etkisi ($\sigma=15, W = 33$)
 a,b) Uzamsal düzlemde Gauss işareti (imshow, mesh)
 c,d) Gauss işaretinin Fourier dönüşümü (imshow, mesh)

[34]' da yapılan çalışma ile W değeri (2.23b)' de verilen ifadeye göre tasarlanırsa Gauss işaretinin %99' dan fazla enerjisini içerdiği gösterilmiştir. Bu sayede pencereleme etkisi olmadan (2.23a)' ya göre maliyeti daha az olan şablonlar tasarlanabilmektedir (Şekil 2.22).



Şekil 2.22 (2.23b)' ye göre tasarlanan şablon ile pencereleme etkisinin giderilmesi
 ($\sigma=15, W = 61$)

a,b) Uzamsal düzlemde Gauss işareti (imshow, mesh)
 c,d) Gauss işaretinin Fourier dönüşümü (imshow, mesh)

Her ne kadar (2.23b), (2.23a)' ya göre maliyeti düşük olsa da şablon boyutları dar bantlı filtrelerin gerçekleştirilmesi için oldukça yüksektir. Filtre şablonunun bir piksel için reel çarpma ve toplama sayıları üzerinden maliyet hesabı (Çizelge 2.1)' de gösterilmiştir.

Çizelge 2.1 Filtre şablonununun 1 piksel için işlem yükü hesabı

Band Genişliği (λ)	Standart Sapma ($\sigma=1/\lambda$)	$W = 4\sigma + 1$	Çarpma Say.	Toplama Say.
0.2	5	21x21	441	440
0.1	10	41x41	1681	1680
0.02	50	201x201	40.401	40.400

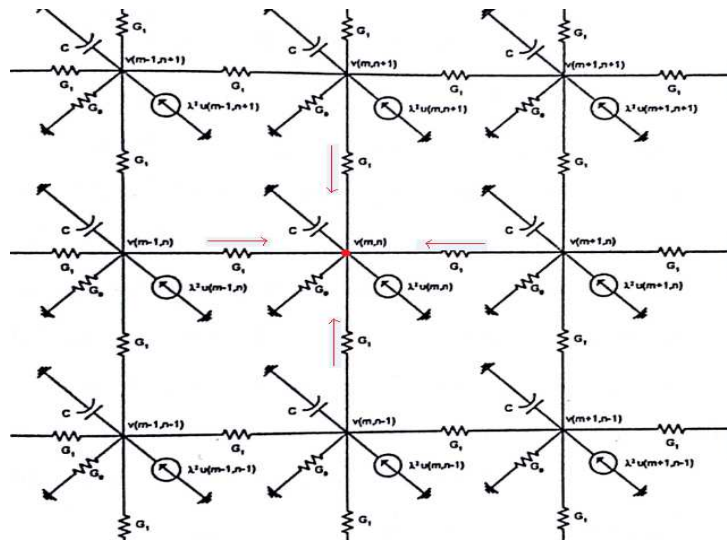
2.5.2 HSA İle Gabor-tipi Filtre Tasarımı

FIR Gabor filtrelerin hesaplama maliyetlerinin yüksek olması sayısal tasarımlarda kullanılan çarpıcı ve blok RAM sayısını arttırmaktadır [34]. Bununla beraber güç tüketim seviyesi de yüksek değerler de olmaktadır.

FIR Gabor filtreler geniş bir uygulama alanına sahip olmasına rağmen tasarımlarının zor olması nedeni ile kullanışlı değildirler. Bu filtreler ile yapılan yön seçici filtreleme özelliği hesap maliyeti çok daha az olan HSA Gabor-tipi filtreler ile yapılabilmektedir. Ayrıca HSA Gabor-tipi filtreler, FPGA gibi donanımlarda, FIR Gabor filtrelere kıyasla daha az çarpıcı ve blok RAM ile tasarlanabilirler [34].

HSA Gabor-tipi Filtrelerin Matematiği

Gabor-tipi filtreler HSA ile tasarlanan alçak geçiren filtrelerden türetilmiştir. HSA alçak geçiren bir filtre (Şekil 2.23)' de ki gibi doğrusal direnç ızgara yapısı ile gerçekleştirilebilir.



Şekil 2.23 Doğrusal dirençli ızgara yapısı ile HSA alçak geçiren filtrenin devre gerçekleştirilmesi

$v(m,n)$ düğümü için Kirchhoff' un akım yasası (KAY) uygulanırsa (2.24) elde edilir.

$$[v(m-1, n) - v(m, n)]G1 + [v(m, n-1) - v(m, n)]G1 + [v(m+1, n) - v(m, n)]G1 + [v(m, n+1) - v(m, n)]G1 - v(m, n)Go - C \frac{dv(m,n)}{dt} + \lambda^2 u(m, n) = 0 \quad (2.24)$$

Devre içerisinde ki iletkenlerin değerleri ($G1 = 1$) , ($Go = \lambda^2$) ve kapasitenin değeri ($C = 1$) olarak alınır (2.24);

$$v'(m, n) = \frac{dv(m,n)}{dt} = -(4 + \lambda^2).v(m, n) + v(m-1, n) + v(m, n-1) + v(m+1, n) + v(m, n+1) + \lambda^2.u(m, n) \quad (2.25)$$

şeklinde olur.

(2.6) ve (2.7)' ya göre A ve B şablonları;

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -(4 + \lambda^2) & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & \lambda^2 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (2.26)$$

olarak bulunur. A ve B şablonları ile işlem şablon çarpımı ile olur. Görüntü işleme için bu işlem, durum görüntüsü ($v(m,n)$) ile giriş görüntüsünün ($u(m,n)$) 3x3 A ve B şablonları ile konvolüsyon edilmesine denk gelir (2.27).

$$\frac{dv(m,n)}{dt} = \sum_{k=-3}^3 \sum_{l=-3}^3 A(k, l).v(m+k, n+l) + \sum_{k=-3}^3 \sum_{l=-3}^3 B(k, l).u(m+k, n+l) \quad (2.27)$$

(2.27)' te ki eşitliğin sol tarafındaki türev ifadesinin değeri ($t \rightarrow \infty$) için sifıra gider. Bunun anlamı durum görüntüsünün piksel değerleri değişmemektedir. Görüntü sabit bir görüntüye yakınsamıştır. $v(m,n)$ ' in değerlerinin değişmediği bu görüntü çıkış görüntüsüdür. Doğrusal HSA için çıkış görüntüsü (2.9)' de belirtildiği gibi durum görüntüsünün kendisine eşit olmaktadır. $x(i,j)$, $v(m,n)$ 'i ve $u(i,j)$, $u(m,n)$ 'i ve \otimes konvolüsyon işlemi temsil etmek üzere ($t > \infty$) için (2.23);

$$0 = A \otimes x_{ij} + B \otimes u_{ij} \quad (2.28)$$

olarak yazılır.

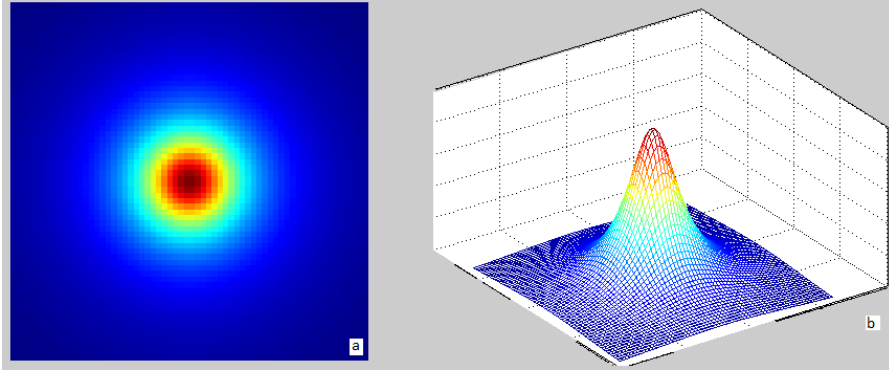
Tasarlanan HSA filtresinin uzamsal düzlemde yaptığı etkiyi görebilmek için frekans karakteristiğini incelemek gerekir. (2.28)' ün 2-B Ayrık Konum Fourier Dönüşümü (AKFD)

alındığında (2.29) elde edilir. (2.29)' te bulunan A_f ve B_f ifadeleri A ve B şablonlarının AKFD ifade eder.

$$0 = (e^{j\omega_x} + e^{-j\omega_x} + e^{j\omega_y} + e^{-j\omega_y} - (4 + \lambda^2)).X(e^{j\omega_x}, e^{j\omega_y}) + \lambda^2.U(e^{j\omega_x}, e^{j\omega_y})$$

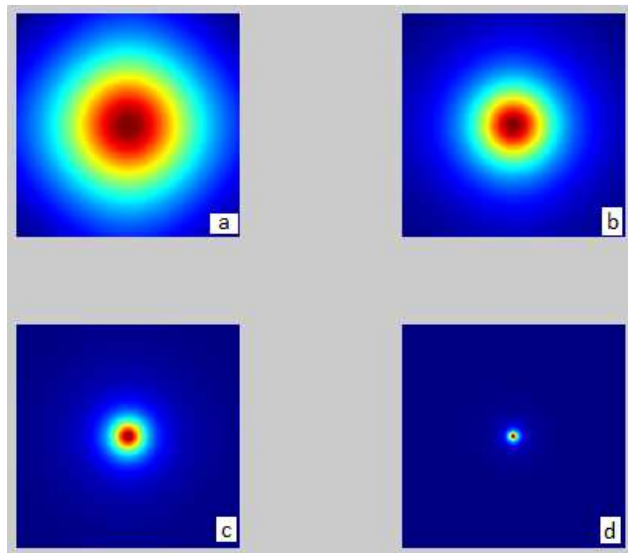
$$H(e^{j\omega_x}, e^{j\omega_y}) = \frac{X(e^{j\omega_x}, e^{j\omega_y})}{U(e^{j\omega_x}, e^{j\omega_y})} = \frac{-B_F(e^{j\omega_x}, e^{j\omega_y})}{A_F(e^{j\omega_x}, e^{j\omega_y})} = \frac{\lambda^2}{4 + \lambda^2 - 2 \cos(\omega_x) - 2 \cos(\omega_y)} \quad (2.29)$$

(2.29) ifadesindeki filtrenin frekans karakteristiği (Şekil 2.24)' de gösterilmiştir.



Şekil 2.24 $H(e^{j\omega_x}, e^{j\omega_y})$ ' nin frekans karakteristiği (a-imshow b-mesh)

(Şekil 2.24)' te görüldüğü gibi HSA filtresi alçak geçiren filtre gibi davranmaktadır. Ayrıca filtreleme karakteristiği Gauss filterisine benzemektedir. Filtre içerisinde ki λ parametresi filtrenin band genişliğini ayarlar (Şekil 2.25).



Şekil 2.25 Farklı band genişliklerindeki Gauss benzeri filtrelerin frekans karakteristikleri
[a) $\lambda = 1$, b) $\lambda = 0.5$, c) $\lambda = 0.2$, d) $\lambda = 0.05$]

Gabor filtresinin Gauss filtresinden türetildiği gibi Gabor-tipi HSA filtresi de Gauss benzeri alçak geçiren HSA filtresinden türetilebilir [35].

Band geçiren bir HSA filtresi alçak geçiren $H_{LP}(e^{j\omega_x}, e^{j\omega_y})$ filtresinin $(\omega_{x0}, \omega_{y0})$ merkez frekansına kaydırılması ile oluşturulabilir. Bir HSA Gabor-tipi filtresinin frekans cevabı;

$$H_{BP}(e^{j\omega_x}, e^{j\omega_y}) = H_{LP}(e^{j(\omega_x - \omega_{x0})}, e^{j(\omega_y - \omega_{y0})}) \quad (2.30)$$

şeklinde elde edilebilir edilebilir.

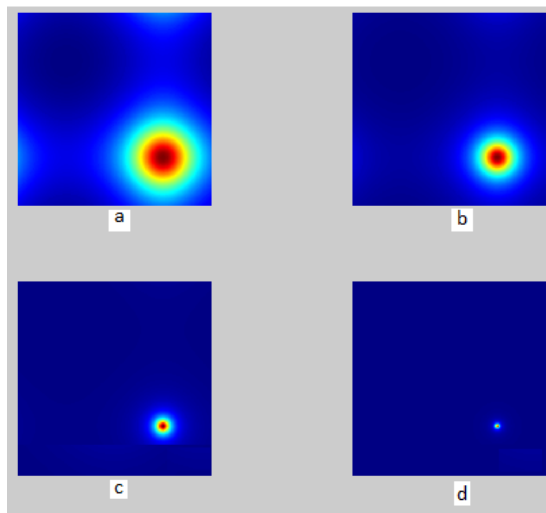
$$H_{BP}(e^{j\omega_x}, e^{j\omega_y}) = \frac{X(e^{j\omega_x}, e^{j\omega_y})}{U(e^{j\omega_x}, e^{j\omega_y})} = \frac{\lambda^2}{4 + \lambda^2 - 2 \cos(\omega_x - \omega_{x0}) - 2 \cos(\omega_y - \omega_{y0})}$$

$$0 = ([e^{-j\omega_{x0}}] \cdot e^{j\omega_x} + [e^{j\omega_{x0}}] \cdot e^{-j\omega_x} + [e^{-j\omega_{y0}}] \cdot e^{j\omega_y} + [e^{j\omega_{y0}}] \cdot e^{-j\omega_y} - (4 + \lambda^2)) \cdot X(e^{j\omega_x}, e^{j\omega_y}) + \lambda^2 \cdot U(e^{j\omega_x}, e^{j\omega_y}) \quad (2.31)$$

(2.30)' da ki band geçiren filtrenin frekans cevabından yola çıkarak (2.31) elde edilmiştir. (2.31)' de ki [.] içerisinde olan ifadeler sabit karmaşık sayılardır. Bu sabit sayılar üzerinden A ve B şablonları yeniden oluşturulursa (2.28) elde edilir.

$$A_{bp} = \begin{bmatrix} 0 & e^{-j\omega_{y0}} & 0 \\ e^{-j\omega_{x0}} & -(4 + \lambda^2) & e^{j\omega_{x0}} \\ 0 & e^{j\omega_{y0}} & 0 \end{bmatrix} \quad B_{bp} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & \lambda^2 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (2.32)$$

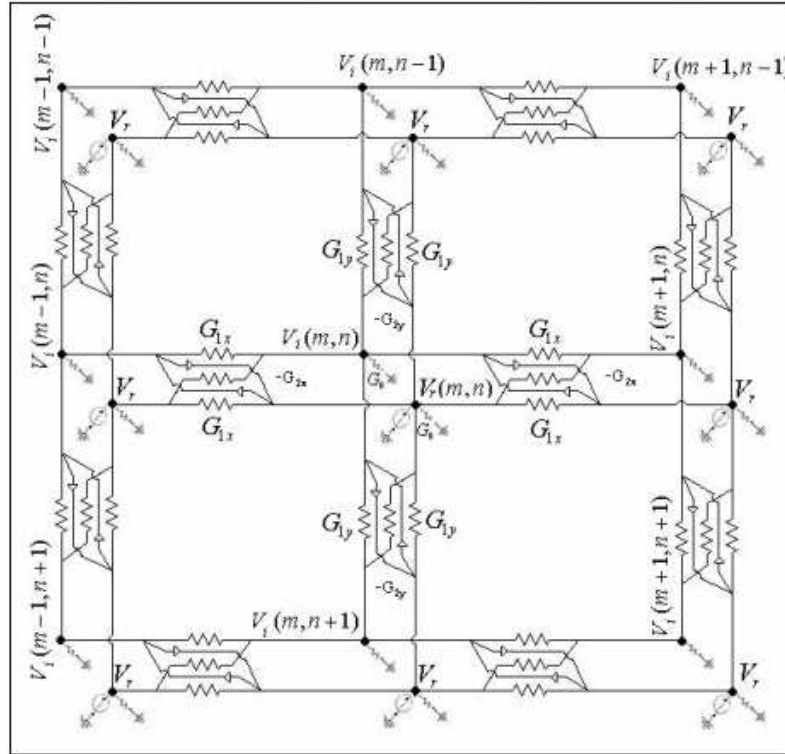
(Şekil 2.26)' de elde edilen A_{bp} ve B_{bp} şablonlarının AKFD' leri alınarak yeni şablonlar ile oluşturulan filtrenin farklı λ değerleri için frekans karakteristiği incelenmiştir.



Şekil 2.26 Farklı band genişliklerindeki Gabor-tipi filtrelerin frekans karakteristikleri
 $(\omega_{x0} = \pi/2, \omega_{y0} = \pi/2, a) \lambda = 1, b) \lambda = 0.5, c) \lambda = 0.2, d) \lambda = 0.05]$

Merkez frekans $(\omega_{x0}, \omega_{y0})$ olan bu filtrenin uzaysal frekans genliği $\omega_o = \sqrt{\omega_{x0}^2 + \omega_{y0}^2}$ ve $\theta_o = \arctan\left(\frac{\omega_{y0}}{\omega_{x0}}\right)$ olmak üzere uzamsal düzlemde filtrelediği yön açısı $\theta = \theta_o + \frac{\pi}{2}$ olarak tanımlanır [20].

Gabor-tipi filtrelerin, direnç ve OTA (Operational Transconductance Amplifier) elemanları kullanılarak iki katmanlı HSA devre gerçeklemelerinin yapılabileceği [35]' de önerilmiştir (Şekil 2.27).



Şekil 2.27 İki boyutlu HSA Gabor-tipi filtrenin analog devresi [3]

Gabor-tipi filtrelerin HSA gerçeklemelerinde A geri besleme şablonu kullanıldığından dolayı hesap yükü FIR Gabor filtrelerine göre ciddi oranda azalmıştır. Simülasyonlarda band genişliği $\lambda = [0.2 \ 0.1 \ 0.02]$ olan Gabor-tipi bir filtre, iterasyona giren iki görüntü arasındaki piksel değerlerinin farkı $\Delta = 0.01$ olacak bir şekilde, maksimum iterasyon sayıları $k = [23 \ 32 \ 88]$ olarak gözlemlenmiştir. Farklı band genişlikleri için iterasyon sayısı ve hesap yükü (Çizelge 2.2)' de verilmiştir.

Çizelge 2.2 HSA Gabor-tipi filtrenin 1 piksel için hesap yükü

Band Genişliği (λ)	İterasyon sayısı	A şablonu Reel çarp./top.	A şablonu İmajiner çarp./top.	B Şablonu çarp./top.	Toplam Çarpma	Toplam Toplama
0.2	23	5/piks. 4/piks	4/piks. 3/piks	1/piks 1/piks	23.(10)= 230	23.(8) = 184
0.1	32				32.(10)= 320	32.(8) = 256
0.02	88				88.(10)= 880	88.(8) = 704

(Çizelge 2.1) ile (Çizelge 2.2) karşılaştırıldığında λ değerleri için sırası ile yaklaşık 2, 5, 45 kat daha az işlem yapıldığı tespit edilmiştir. (Çizelge 2.2)' de ki iterasyon sayıları yapılan denemler de elde edilen maksimum iterasyon sayılarıdır.

Tablo karşıştırmalarından görüleceği üzere Gabor-tipi filtreler FIR Gabor filtrelere göre işlem yükü açısından oldukça verimlidir. Ayrıca FPGA tasarımlarında kullanılan çarpıcılar ve blok RAM açısından da verimlilik ciddi oranda yükseltir [34].

Gabor-tipi filtreler, yön seçici özelliği ve verimliliği ile görüntülerden hızlı bir şekilde öznitelik çıkartmada tercih edilmekte ve FPGA gibi sayısal tasarımlarda eş zamanlı uygulamalar için HSA ile gerçekleştirilebilmektedir.

HSA GABOR-TİPİ FİLTRELER KULLANILARAK EL YAZISI KARAKTER TANIMA

3.1 Giriş

Gabor filtresi örüntü işlemede sıklıkla kullanılan öznitelik çıkartma tekniklerinden biridir [36]. Doku analizleri [36], bölütleme [37], kenar belirleme gibi örüntü işleminin temel problemlerinin yanı sıra iris tanıma [38], parmak izi iyileştirme [29,39] ve tanınması [40], yüz tanıma [41] ve plaka tanıma [42] gibi spesifik problemlerin de çözümünde sıklıkla kullanılmıştır. Ayrıca doku analizleri için uygun olabilecek optimal Gabor filtre bankası seçimi için algoritmalar geliştirilmiştir [37,43].

Literatürdeki çalışmalar gösteriyor ki Gabor filtreleri teorik çalışmalarda ve pratik uygulamalarda üzerine yoğun çalışılan konulardan birdir. Özellikle örüntülerden öznitelik çıkartmada güçlü bir yöntemdir.

Gabor filtrelerinin kullanıldığı bazı uygulama alanları gerçek zamanlı çalışmalardır. Bu sebeple örüntü üzerinden öznitelik çıkartma kısımlarında zaman tüketiminin az olması istenmektedir. (Bölüm 2.5.1)' de gösterildiği üzere Gabor filtrelerinin FIR filtre olarak tasarlanıp kullanılması ciddi bir işlem yükü getirmektedir. Bu işlem yükü, zaman tüketimini arttırmanın yanında sayısal sistemlerde ki güç tüketiminin de fazla olmasına neden olmaktadır. Bu problem HSA ile tasarlanan Gabor-tipi filtreler ile giderilebilmektedir.

Gabor filtrelerinin yön seçici filtremle özelliği, HSA' lar ile gerçekleştirilebilen Gabor-tipi filtreler ile sağlanabilmektedir [35]. Ayrıca Gabor-tipi filtrelerin FPGA gibi VLSI sistemlerde tasarlanabildiği [18]' de gösterilmiş ve [19 ve 44]' de bu filtrelerin etkili bir şekilde FPGA üzerinde gerçekleştirilebilmeleri için yeni yöntemler geliştirilmiştir. [19 ve 44]' deki çalışmalarda Gabor-tipi filtreler ile Full HD görüntülerin bile hızlı bir şekilde

filtrelenebileceği ortaya konmuştur. Bu çalışmalar ile Gabor-tipi filtreleme sayesinde hesap yükü önemli ölçülerde minimize edilebilmiştir.

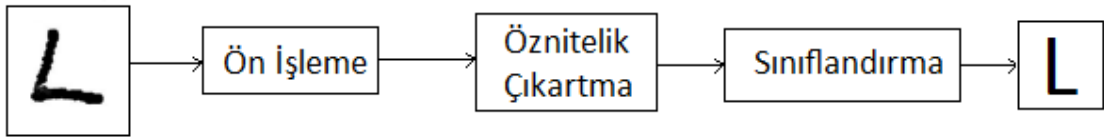
Literatürdeki çalışmalarda görüleceği üzere Gabor filtrelerin yerine Gabor-tipi filtreler kullanılarak daha verimli ve daha hızlı sayısal sistemler tasarlanabilmektedir. El yazısı karakter tanıma bu sistemlere güzel bir örnektir.

Bu bölümün konusu olan Gabor-tipi filtreler ile el yazısı karakter tanıma, literatürde üzerine çalışılan konular arasındadır [16,17]. Buradaki tez çalışması da [16,17]'deki yayınlar ile [29]'da bu konu ile alakalı olan bölümden esinlenerek hazırlanmıştır.

Bu bölümün (3.2). kısmında [29]'da yapılan Gabor-tipi filtreler ile el yazısı karakter tanıma çalışması anlatılmıştır. (3.3). kısım da ise yeni geliştirilen algoritma ve [17] arasında ki farklarından bahsedilmiştir.

3.2 Gabor-tipi Filtreler ile El Yazısı Karakter Tanıma

[29]'de yapılan çalışmada el yazısı karakterler (Şekil 3.1)'de ki gibi üç aşamalı bir sistem ile tanınmıştır.



Şekil 3.1 [29]'de tasarlanan el yazısı karakter tanıma sisteminin genel blok diyagramı

Bu sistemde ilk olarak ön işleme bloğunda boyutları rastgele $I \times J$ olan giriş görüntüleri Doğrusal Olmayan Şekil Normalizasyonu (Nonlinear Shape Normalisation (NSN)) ile sabit $M \times N$ boyutlarına getirilmiştir. Öznitelik çıkartma bloğunda ise Gabor-tipi filtreler kullanılarak öznitelikler çıkartılmıştır. Öznitelik çıkarma işlemi için ilk olarak Tek Ölçekli Tanıma Sistemi (Single Scale Recognition System) tasarlanmış, ardından tanıma oranının daha da artması için tasarım Çok Ölçekli Tanıma Sistemi (Multiscale Recognition System)'ne tasarım genişletilmiştir. Bu sistemler ile el yazısı harfin yönlü bileşenleri filtrelenmiştir. Sınıflandırma bloğunda ise öznitelik çıkartma bloğundan gelen ve yönlü bileşenleri içeren görüntüler ile daha önceden üretilen veri tabanında bulunan şablonların korelasyonu alınmıştır. Burada ki korelasyon katsayısının büyüklüğüne göre tanıma işlemi gerçekleştirilmiştir.

3.2.1 Ön İşleme

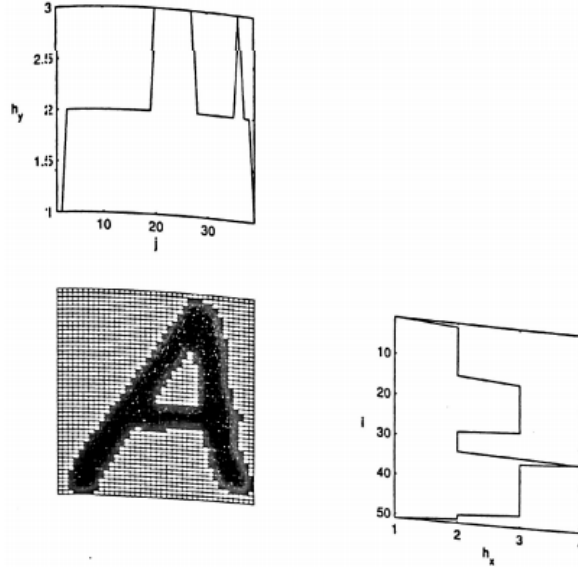
Ön işleme el yazısı karakterlerin büyüklüklerine ve kalınlıklarına göre çeşitlilik göstermektedir. Bu çeşitliliğin tanıma üzerindeki olumsuz etkisini azaltmak için ön işleme basamağında şekil normalizasyonu yapılmaktadır. Bu işlem için NSN dönüşümü kullanılmıştır. NSN dönüşümü ile rastgele boyutlarda olan giriş görüntüsü (MxN)'lik sabit boyutlu bir görüntüye dönüştürülmüştür. Bu sayede boyutlarından bağımsız daha gürbüz bir tanıma sistemi tasarlanmıştır.

NSN dönüşümü rastgele boyutlarda olan bir görüntüyü sabit boyutlara getirmek için kullanılan doğrusal olmayan bir dönüşümdür. Literatürde tanımlanan bütün NSN dönüşümlerin temeli giriş görüntüsünün iz düşümü histogramındaki kesişen çizgilere ya da siyah noktaların sayısına dayalıdır. [29]' da kullanılan NSN dönüşümü Yamada tarafından geliştirilmiştir [44]. Bu NSN dönüşümü giriş görüntüsünde satırda ya da sütunda kesişen çizgilerin yoğunluğu üzerinden işlem yapmaktadır [44].

$h_x(i)$ ve $h_y(j)$ sırasıyla yatay ve dikey doğrultudaki çizgi kesişimlerinin iz düşümü olsun. Çizgi yoğunluklarının iz düşümleri (3.1)' de ki gibi hesaplanmaktadır.

$$\begin{aligned} h_x(i) &= \sum_{j=1}^J [f(i, j) \cdot f(i, j - 1)^*] + \alpha \\ h_y(j) &= \sum_{i=1}^I [f(i, j) \cdot f(i - 1, j)^*] + \alpha \end{aligned} \quad (3.1)$$

Burada $f(i, j)$, $I \times J$ boyutlarındaki giriş görüntüsünün (i, j) ' nci pikseldeki değerini temsil eder. Görüntü ikili (binary) formatta olduğu için bu değer 1 ya da 0 olabilir. $f(i, j)^*$ ise $f(i, j)$ ' de bulunan pikselin tümleyenini temsil eder. α değeri, doğrusal olmayan dönüşüm için doğal sayı sabitidir.



Şekil 3.2 Giriş görüntüsünün yatay (h_x) ve dikey (h_y) iz düşümleri [29]

(Şekil 3.2)' de sistemin girişine gelen "A" harfi görüntüsü içerisindeki kesişen yerlerin çizgi yoğunluk iz düşümleri çıkartılmıştır. α katsayısı 1 olarak alınmıştır. Yatay ve dikey iz düşümleri çıkartılırken görüntü satırda ve sütunda taranarak piksel değerinin 1 değerinden 0 değerine geçiş noktaları hesaplanır. Bu değer her piksel için hesaplanır ve (Eşitlik 3.1)' de ki gibi birikimsel olarak toplanır.

$$\min \left\{ i \mid \sum_{k=1}^i h_x(k) \geq m \left(\frac{\sum_{k=1}^I h_x(k)}{M} \right) \right\}$$

$$\min \left\{ j \mid \sum_{k=1}^j h_y(k) \geq n \left(\frac{\sum_{k=1}^J h_y(k)}{N} \right) \right\} \quad (3.2)$$

Hesaplanan çizgi yoğunluk iz düşümleri Yamada tarafından tanımlanan (3.2)' de ki öznitelik yoğunluk dengeleme eşitlikleri ile $I \times J$ olan giriş görüntüsünden $M \times N$ boyutlarında normalize görüntüsü elde edilmiştir [44]. (m,n) ikilileri $m = 1, \dots, M$ ve $n = 1, \dots, N$ kadar olmak üzere sırası ile verilen i ve j değerleri için eşitsizliği sağlayan ilk (m,n) çiftinin piksel değeri giriş görüntüsünde bulunan (i,j) ' de ki piksel değeri olmaktadır.

3.2.2 Öznitelik Çıkartımı

El yazısı harfler farklı açılarda, yerlerde ve uzunluklardaki doğruların birleşimi şeklinde düşünülebilir. HSA Gabor-tipi filtrelerin yön ve frekans seçici filtre bankaları kullanılarak bu bileşenler harflerden süzülebilir ve harflerin özniteliklerini çıkartmada kullanılabilir.

Öznitelik çıkartma bloğu için [29]' de iki tür sistem anlatılmıştır. Bunlar Tek Ölçekli Tanıma Sistemi ile Çok Ölçekli Tanıma Sistemleridir.

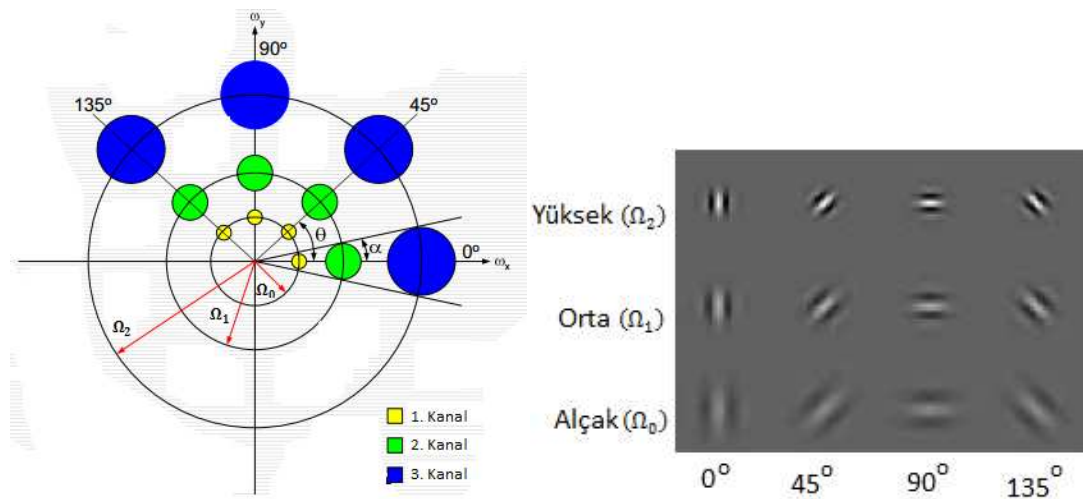
$$\left\{ \omega_{x_0}^k = \Omega_o \cdot \cos(\theta_k), \omega_{y_0}^k = \Omega_o \cdot \sin(\theta_k) \mid \theta_k = \frac{\pi}{2} + \frac{k\pi}{n_f}, k = 0, \dots, (n_f - 1) \right\} \quad (3.3)$$

Bu sistemler HSA Gabor-tipi filtrelerin (3.3)' te belirtilen ω_{x_0} ve ω_{y_0} olan merkez frekansını belirlemede kullanılan Ω_o uzamsal frekans genliğinin sayısına göre tek ve çok ölçekli olmak üzere ikiye ayrılır. Tek bir uzamsal frekans genliği için tasarlanan filtreler tek ölçekli, birden fazla kullanılan sistemler için tasarlanan filtreler ise çok ölçekli tanıma sistemi olarak adlandırılmıştır. (3.3)' te bulunan n_f tasarlanan filtrelerin süzeceği açıları tanımlamak için kullanılmaktadır.

Tanıma oranı daha yüksek olduğu için bu çalışmada Çok Ölçekli Tanıma Sistemi tercih edilmiştir.

Çok Ölçekli Tanıma Sistemi

Harf tanıma sistemlerinde giriş görüntüsünün boyutlarının farklılığı ve görüntü içerisindeki harfin kalınlığının her zaman aynı olmaması önemli problemlerdir. Giriş görüntüsündeki boyut farklılığı NSN dönüşümü ile giderilebilir. Harflerdeki kalınlık problemi ise kullanılacak olan filtrelerin farklı uzamsal frekans genlikleri kullanılarak giderilebilir.

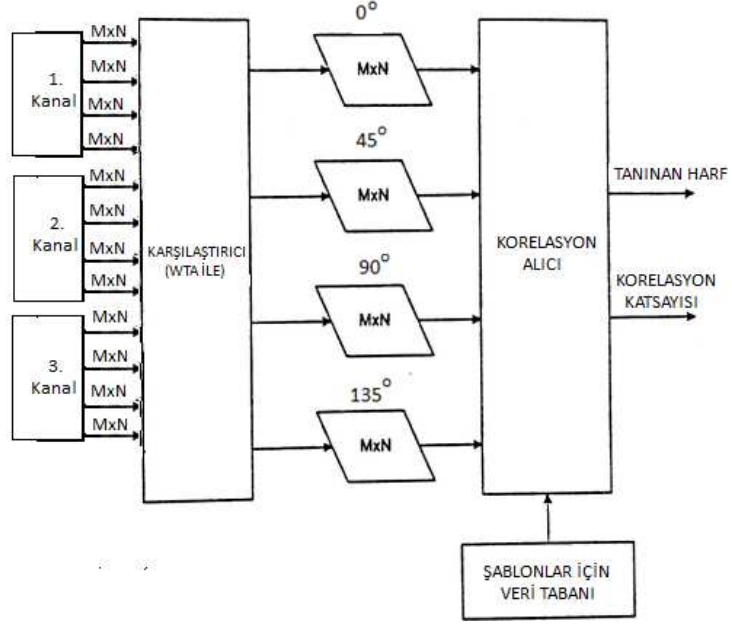


Şekil 3.3 Farklı uzamsal frekans genliklerinde ve 4 farklı açıda bulunan Gabor-tipi filtreler (sol) [8], Filtrelerin birim impuls yanıtları (sağ) [45]

Uzamsal düzlemde harfler üzerindeki kalınlık ve incelik sırası frekans düzlemindeki alçak frekans ve yüksek frekans bölgelerindeki bileşenlere denk gelmektedir. Eğer bir harf kalın bir şekilde yazılırsa harfin yönlü bileşenlerinin frekans içerikleri alçak frekans bölgesine; ince bir şekilde yazılırsa yüksek frekans bölgesine denk gelir. Sistem olarak sadece tek bir uzamsal frekans genliği (Ω) seçilirse 4 farklı açıda fakat sadece bir frekans bölgesinde filtreler oluşturulacaktır. Bu nedenle filtrelerin harfin kalınlık ya da inceliğine göre etkili bir öznelik çıkarmadaki performansı değişecektir. Sistemin harfin kalınlığından bağımsız olabilmesi için 4 farklı açıda ve 3 farklı frekans bölgesinde filtreler tasarlanmalıdır. Bu nedenle Ω_0 , Ω_1 , ve Ω_2 uzamsal frekans genlikleri ile (3.3)' e göre 12 ayrı Gabor-tipi filtre tasarlanmıştır. Alçak, orta ve yüksek frekans bölgelerinde bulunan farklı açıdaki filtreler 1. Kanal, 2. Kanal ve 3. Kanal olarak gruplandırılmıştır. "Winner Take All (WTA)" sinir ağı ile 3 farklı kanaldan (alçak, orta, yüksek frekans bölgeleri) elde edilen filtrelenmiş görüntüler içerisinde aynı açıdaki bileşenleri baskın olan görüntüler seçilmiştir. Aynı açıda farklı frekans bölgesindeki filtrelerden elde edilen çıkışlar WTA ile karşılaştırılıp bileşenlerin daha baskın olduğu görüntüler seçilmiştir. Bu sayede 12 filtre çıkışından 4 farklı açıdaki 4 baskın bileşenli görüntü elde edilmiştir.

3.2.3 Sınıflandırma

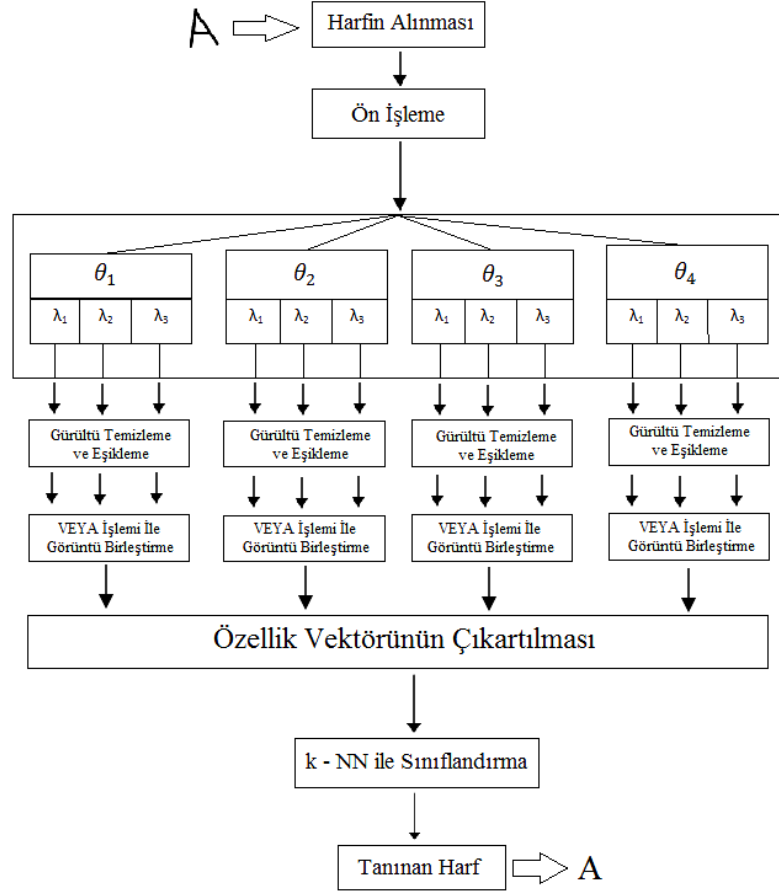
Öznelik çıkartma bloğundan elde edilen, 4 farklı açıdaki baskın bileşenli görüntülerin daha önceden hazırlanan veri setindeki şablonlar ile korelasyonu alınmıştır. Korelasyon katsayısı yüksek olan şablonlar üzerinden tanıma işlemi gerçekleştirilmiştir. Sistemin blok diyagramı (Şekil 3.4)' te gösterilmiştir.



Şekil 3.4 WTA ile karşılaştırıcıdan çıkan 4 farklı açıdaki baskın bileşenli görüntülerin sınıflandırılması

3.3 Tasarlanan Yeni Algoritma

Yeni tasarlanan algoritma [29]' de ve [16]' te anlatılan algorithmandan esinlenerek oluşturulmuştur. Geliştirilen algoritmanın blok diyagramı (Şekil 3.5)' teki gibidir.



Şekil 3.5 Algoritmanın blok diyagramı

Yeni algoritmada performansın daha da artırılması için ön işleme basamağına eğim düzeltme algoritması eklenmiştir. Bu sayede eğik olan harfler düzeltilmiş ve bu harflerin de tanınması sağlanmıştır.

Öznitelik çıkartma yöntemi olarak yine HSA Gabor-tipi filtreler kullanılmıştır. Fakat filtre çıkışlarının WTA ile karşılaştırılması yerine Histograma Dayalı Eşikleme (HDE) ile görüntüler eşiklenmiş, ardından bu görüntülerden aynı açıda olanlar mantıksal “veya” işlemi ile toplanmıştır. Toplanan görüntüler üzerinden öznitelik vektörleri çıkartılmıştır.

Sınıflandırma basamağında korelasyon yerine literatürde yaygın olarak kullanılan k-NN algoritması tercih edilmiştir.

3.3.1 Ön İşleme

Kamera ya da tarayıcı gibi cihazlardan alınan harflerin boyutları ve türleri farklı olmaktadır. Filtrelemeden önce alınan görüntülerin sabit bir boyutta [64x64], ikili görüntü türünde ve görüntü içindeki harfin kalınlığının sabit olması gerekmektedir.

Ayrıca tanıma performansını daha da arttırmak için eğik olan harfler düzeltilmiş ve bu şekilde filtrelenmiştir.

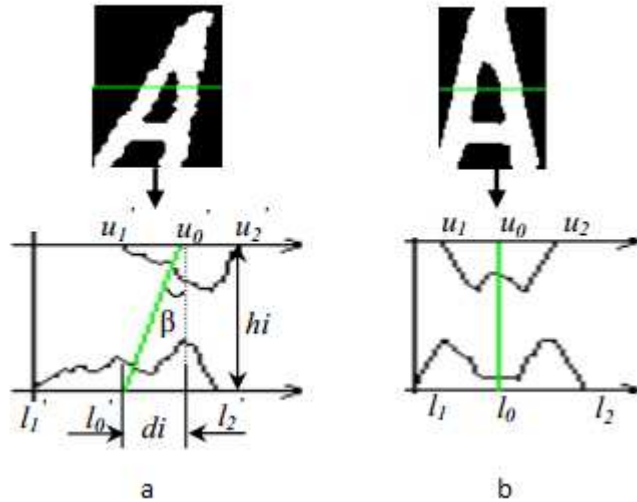
Boyut ve Türlerin Ayarlanması

Dışarıdan alınan rastgele boyutlardaki görüntüler üzerinden doğru bir öznitelik çıkartımı yapabilmek için işlenecek olan görüntülerin sabit boyutlarda olması gerekmektedir. Bunun için (Bölüm 3.2.1)' de anlatılan NSN dönüşümü kullanılmıştır.

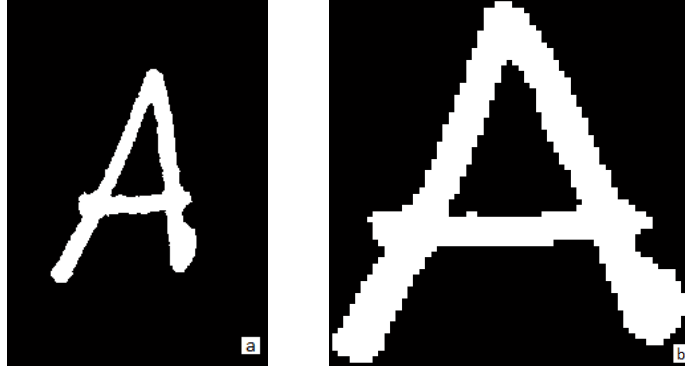
Dikey Eğim Düzeltme

Alınan harfler bazen eğik olabilmekte ve bu durum tanıma işlemini güçleştirmektedir. Bu nedenle ikili görüntü türünde ve 64x64 boyutlarında olan harfin eğimi dikey eğim düzeltme algoritması ile eğimi düzeltilebilmektedir.

Dikey eğim düzeltme algoritmasının çalışması şu şekildedir; 64x64 boyutundaki görüntü yatay olarak ortadan ikiye bölünür. Her iki parçanın ayrı ayrı yatay eksen üzerine iz düşümü alınır ve iz düşümlerin orta noktaları hesaplanır. Alt ve üst iz düşümün orta noktaları arasındaki açı eğim açısını verir. Bu açı kullanılarak görüntünün alt ve üst parçalarındaki piksellerin yerleri eğim açısı olmayacak şekilde tekrar düzenlenir (Şekil 3.6) [46].



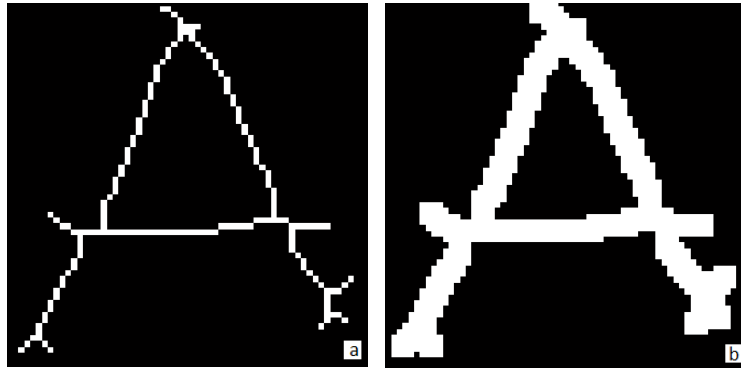
Şekil 3.6 Eğim düzeltme algoritması ile eğik harfin düzeltilmesi [46]



Şekil 3.7 a) Eğik ve boyutu ayarlanmamış harf b) Eğikliği giderilmiş 64x64 boyutlarında olan harf

Harfin Kalınlığının Sabitlemesi

Filtreleme basamağından önce son adım harfin kalınlığının sabitlemesidir. Bunun için yine MATLAB programının ikili görüntülerin iskeletini çıkartmaya yönelik hazır kodu kullanılmıştır. İskeleti çıkartılan görüntüye 3 x 3 boyutlarındaki sabit bir yapısal eleman ile genişletme işlemi (Bölüm 4.2.1) uygulanmış ve harfin kalınlığı sabit hale getirilmiştir. Bu işlem (Şekil 3.8)' da gösterilmiştir.



Şekil 3.8 a) İskeleti çıkarılmış harf b) YE ile genişletilmiş harf

Harfin kalınlığının sabit olması sınıflandırma adımı için oldukça önemlidir. Yapılan denemelerde görüldüğü üzere harfin kalınlığının değişmesi tanıma işlemini zorlaştırmaktadır. Sabit kalınlık ile çalışmak doğruluk oranını arttırmaktadır.

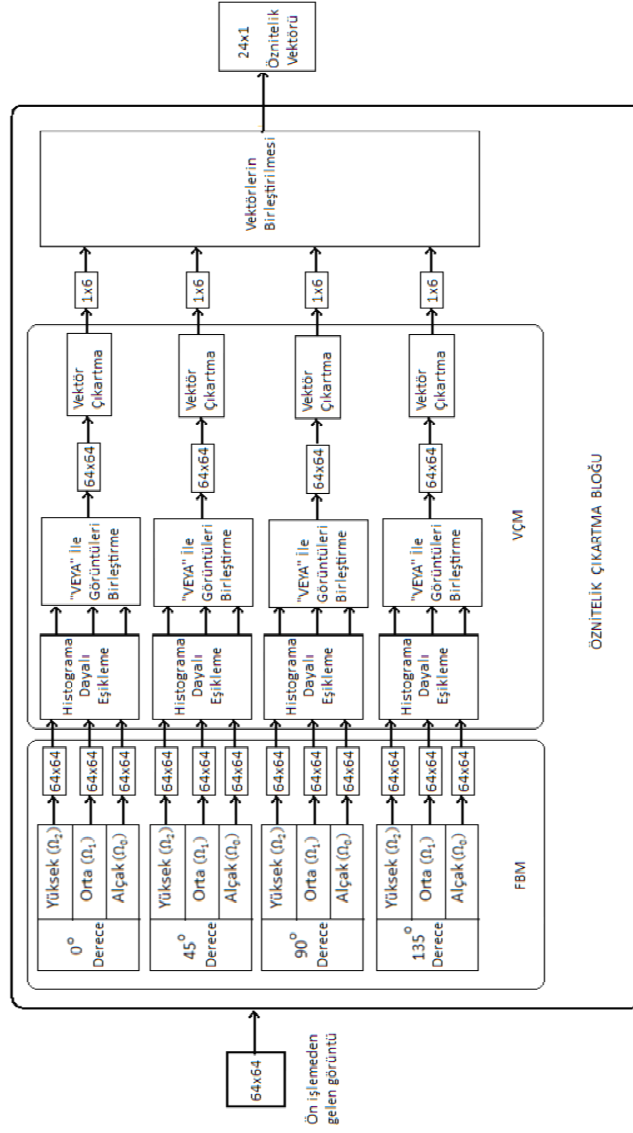
Kalınlığı sabit olan ikili görüntünün "0" ve "1" piksel değerleri, HSA ile filtrelemeye uygun hale getirmek için (3.4)' e göre hesaplanmıştır. HSA' larda işlem yapabilmek için beyaz olan piksel "-1", siyah olan piksel ise "1" olarak alınır. (3.4)' teki $I_{bin}(i,j)$,

içerisindeki harfin kalınlığı sabit olan ikili (binary) görüntüyü ve $I_{HSA}(i,j)$ ise filtrelenecek olan görüntüyü temsil etmektedir.

$$I_{HSA}(i,j) = -2 \cdot (I_{bin}(i,j)) + 1 \quad (3.4)$$

3.3.2 Filtreleme ve Öznitelik Vektörlerinin Çıkartılması

(Şekil 3.9)' da bulunan öznitelik çıkartma bloğunun içerisinde iki ana modül bulunmaktadır. Bunlardan ilki, 0° , 45° , 90° , ve 135° açılarda ve alçak, orta ve yüksek frekans bölgelerine konuşlandırılmış 12 adet HSA Gabor-tipi filtreden oluşan Filtre Bankası Modülü (FBM); ikincisi ise filtrelenmiş görüntülerden öznitelik çıkartan Vektör Çıkartma Modülü (VÇM)' dür.

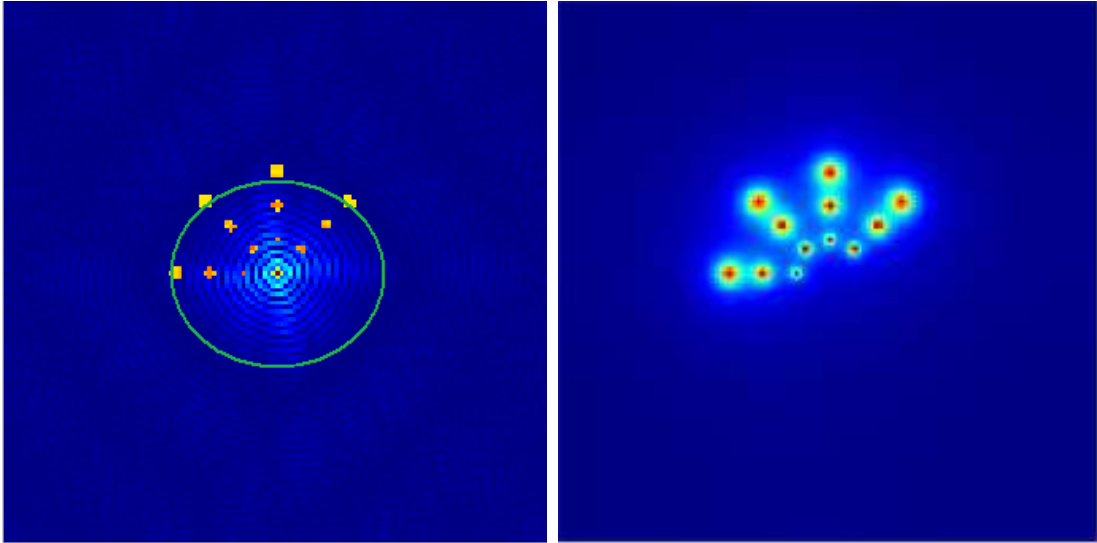


Şekil 3.9 Öznelik çıkartma bloğunun iç yapısı

Filtre Bankası Modülü (FBM)

FBM' de, (Şekil 3.3)' teki filtre bankası tasarlanmıştır. Filtre bankasındaki Ω değerleri sırası ile $(\frac{\pi}{8}, \frac{2\pi}{8}, \frac{3\pi}{8})$ ve Ω değerlerine bağlı filtrelerin band genişlikleri λ değerleri ise $(0.0392, 0.0769, 0.1117)$ ' dir. Tasarlanan filtrelerin frekans karakteristikleri (Şekil 3.10)' da görülmektedir.

Ω değerleri seçilirken "O" harfinin frekans karakteristiğinden yararlanılmıştır. Bunun nedeni "O" harfi bütün yönlerde bileşenler içerdiği için frekans karakteristiği diğer harfleri kapsayıcı niteliktedir. (Şekil 3.10)' da merkezden başlayıp dalgalar şeklinde yayılan örüntü "O" harfinin 2-B Fourier dönüşümüdür. Sarı veya turuncu olan yerler (merkez hariç) ise tasarlanan filtrelerin merkez frekanslarıdır. (Şekil 3.10) üzerinde "O" harfinin içerdiği frekans değerlerinin (daire şeklinde olanlar) yüksek frekanslara gidildikçe azaldığı görülmektedir. Yeşil daire içerisinde kalan alan harf bilgisinin büyük bir kısmını oluşturmaktadır. Bu nedenle maksimum Ω değeri bu dairenin yarıçapına göre alınmıştır. Diğer Ω değerleri de maksimum Ω değerine göre seçilmiştir.

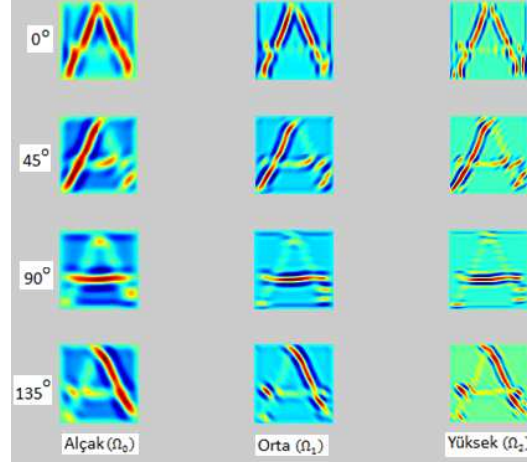


Şekil 3.10 "O" harfinin FFT' si ve filtrelerin merkez frekansları (sol)
FBM içerisindeki filtrelerin frekans karakteristikleri (sağ)

HSA Gabor-tipi filtre bankasında alçak frekans bölgesinde bulunan filtrelerin λ değeri en düşüktür. Bu filtrelerin FIR eş değerindeki uzamsal filtre şablonları ise diğerlerine göre büyüktür (Çizelge 2.1). Filtre şablonu büyük olan FIR filtreler HSA yapıları ile gerçekleştirirken iterasyon sayısı artar (Çizelge 2.2). Deneysel gözlemlerden elde edilen maksimum sonuçlara göre alçak frekans bölgesindeki filtreler 88 iterasyon, orta

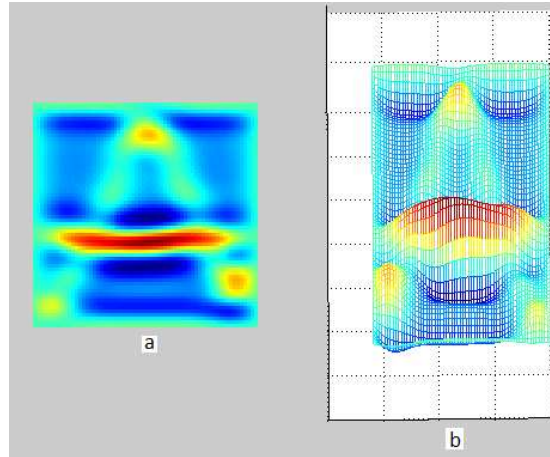
frekans bölgesindeki filtreler 32 iterasyon ve yüksek frekans bölgesindeki filtreler 23 iterasyon ile simüle edilebilmektedir. İterasyon sayıları, iterasyondaki art arda iki görüntünün piksel değerleri arasındaki fark 0.01 değerinden az oluncaya kadar tekrarlanma sayılarıdır.

Filtrelere giriş olarak (Şekil 3.8-b)' deki harf verilmiştir. İterasyonların bitiminde filtrelerin çıkışları (Şekil 3.11)' de görülmektedir.



Şekil 3.11 Filtrelerin çıkışları

Filtreler kendi yönleri doğrultusundaki harf bileşenlerinin değerlerini yükseltmiş, diğer bileşenleri ise bastırmıştır. (Şekil 3.11)' de kırmızı olan yerler filtrelerin geçirdikleri, diğer renklerdeki kısımlar ise bastırdıkları yerlerdir.



Şekil 3.12 90° ve orta frekans bölgesindeki filtre çıkışının "imshow" ve "mesh" komutları ile gösterimi a) "imshow" komutu b) "mesh" komutu

FBM ile harflere ait bileşenler çıkartılmıştır. 12 filtreden elde edilen çıkışlardan Vektör Çıkartma Modülü (VÇM) ile öznitelik vektörleri çıkartılmaktadır.

Vektör Çıkartma Modülü

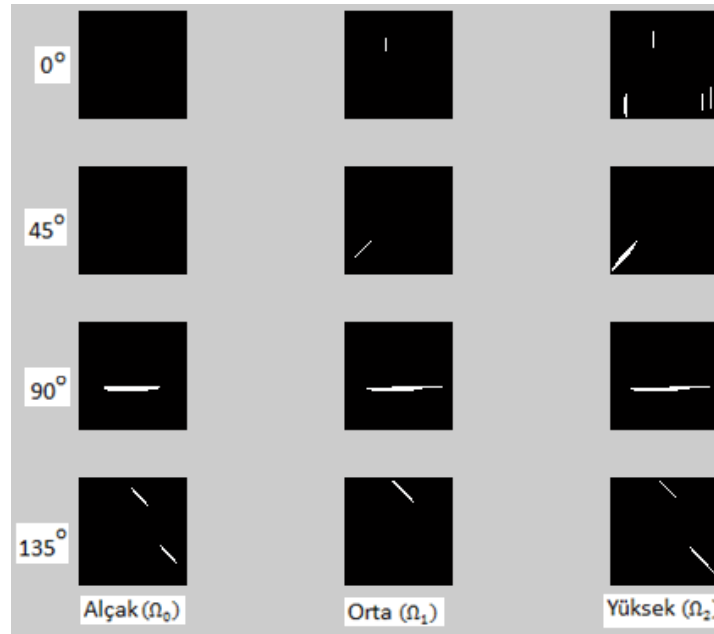
VÇM kendi içerisinde Histograma Dayalı Eşikleme (HDE), “veya” ile görüntü birleştirme ve birleşik ikili görüntülerden vektör çıkartma olmak üzere 3 ayrı bileşenden oluşmaktadır.

HDE kısmında filtre modülünden gelen çıkışlar içerdikleri maksimum ve minimum değere göre [0-255] aralığına ölçeklenir ve tam sayılara çevrilerek $g_{int}(i,j)$ görüntüleri elde edilir. Elde edilen tam sayı formatındaki görüntülerin histogramı çıkartılır. Histogram üzerinden, 255 değerinden itibaren görüntünün enerjisinin %2' sini içerecek olan β sınırı belirlenir.

$$g_{bin}(i,j) = 1, \quad g_{int}(i,j) \geq \beta$$

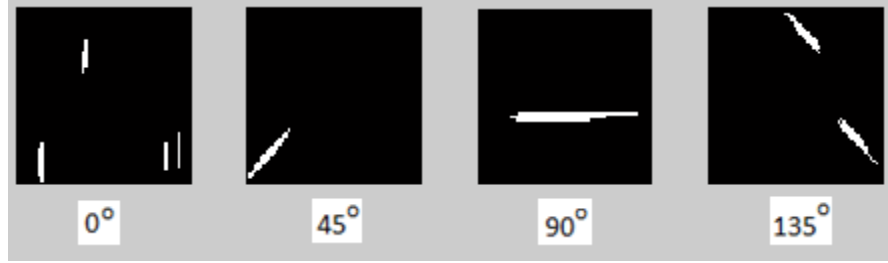
$$g_{bin}(i,j) = 0, \quad \text{diğer} \quad (3.5)$$

(3.5)' e göre $g_{int}(i,j)$ ' nin piksel değeri β ile 255 arasında olanların değeri “1” diğerlerinin değeri “0” olmaktadır. Bu sayede $g_{bin}(i,j)$ görüntüleri elde edilir. Elde edilen ikili değer türündeki görüntüler üzerinde morfolojik açma işlemleri (Bölüm 4.2.3) uygulanarak anlamlı bileşenler çıkartılır.



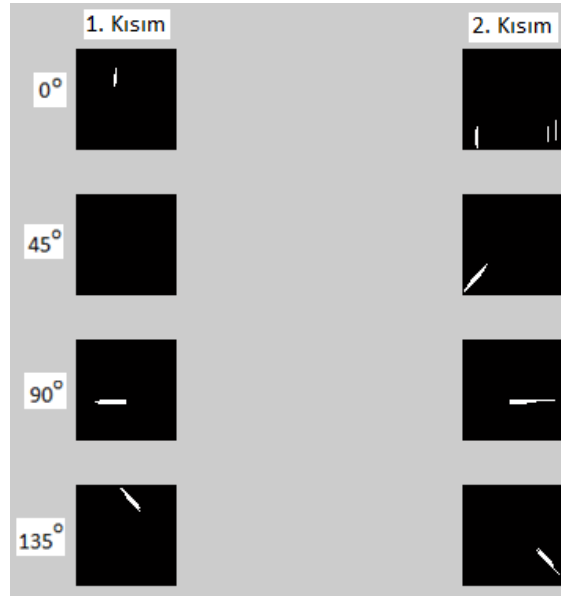
Şekil 3.13 HDE ile elde edilen ikili değer türündeki bileşenler

Elde edilen görüntülerden aynı açılı olanlar mantıksal “veya” işlemi ile toplanırlar. Bu sayede belli bir açığa ait farklı frekans bölgelerindeki bileşenler üst üste gelerek harfin o açı doğrultusundaki baskın bileşenini meydana getirirler.



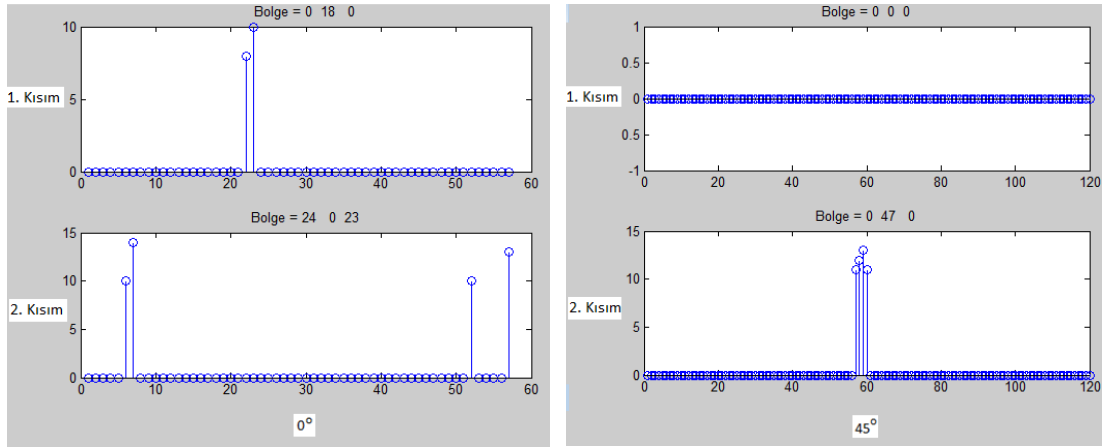
Şekil 3.14 Aynı açıdaki görüntülerin birleştirilmesi ile elde edilen birleşik görüntüler

Filtreler ile elde edile bileşenlerin uzamsal bölgede ki yerlerinin de bilinmesi tanıma işlemi için önem taşımaktadır. Bunun için öznitelik vektörü çıkartılmadan önce birleşik görüntüler, içerdikleri bileşenlere dik ve görüntünün ortasından geçen bir normal doğrusu ile iki kısma ayrılır.

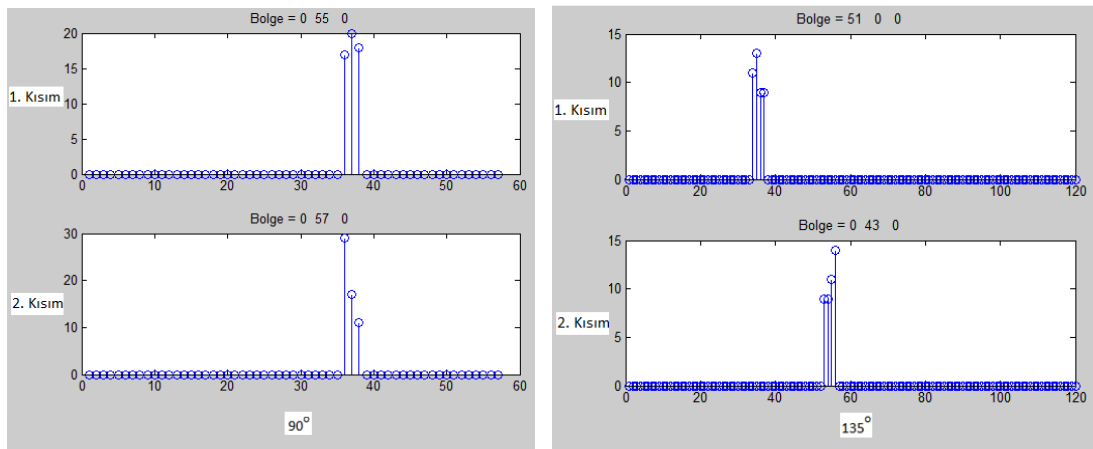


Şekil 3.15 İçerdikleri bileşenlere göre dik olan normal doğrusu ile ayrılmış görüntüler

Ayrılan görüntülerin normal doğrusu üzerine iz düşümlerinin toplamları alınır. İz düşüm toplamları 0° ve 90° için 64; 45° ve 135° için 128 boyutlarında bir vektördür. Elde edilen bu vektör eşit olarak üç parçaya bölünür ve parçalar içerisinde bulunan değerler bölgesel olarak toplanır.



Şekil 3.16 0° ve 45° için iz düşümü vektörleri ve bölgesel toplamları



Şekil 3.17 90° ve 135° için iz düşümü vektörleri ve bölgesel toplamları

Her açı için iki kısım ve her kısım içinde 3 bölgesel toplam bulunmaktadır. Bir açı için elde edilen bölgesel toplamlar hesaplandıktan sonra (3.6)' ya göre θ açısına dair 1x6 uzunluklu öznitelik vektörü çıkartılır.

$$U_{\theta} = [1. \text{Kısım Bölgesel Toplamlar}, 2. \text{Kısım Bölgesel toplamlar}] \quad (3.6)$$

Vektör çıkartma modülünden gelen 4 açı için 1x6 uzunluklu vektörler (3.7)' e göre el yazısı harfin 24x1 uzunluğunda öznitelik vektörünü oluştururlar. Oluşturulan öznitelik vektörünün elemanları, her bir bölgenin toplam değer üzerindeki yüzdesi ile oluşturulmuştur.

$$V = 100. \frac{([U_0, U_{45}, U_{90}, U_{135}]^T)}{\text{Toplam değer}} \quad (3.7)$$

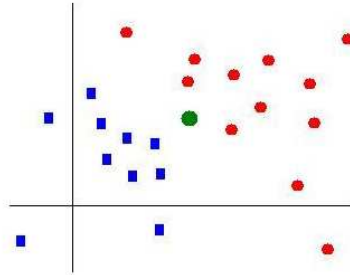
3.3.3 Sınıflandırma

El yazısı harf tanıma algoritmalarında birçok sınıflandırıcı kullanılmıştır. Bunlardan tanıma performansı açısından yüksek olanlar en yakın k komşu (k-NN), destek vektörü makinesi (SVM) ve modifiye edilmiş karesel diskriminant fonksiyonudur (MQDF) [47]. MQDF ve SVM 'in sayısal tasarım maliyeti ve zaman tüketimi içerdikleri ağır matematiksel işlemlerden dolayı oldukça yüksektir. Bu nedenlerden dolayı sınıflandırıcı bloğunda k-NN algoritması kullanılmıştır.

k-NN Algoritması

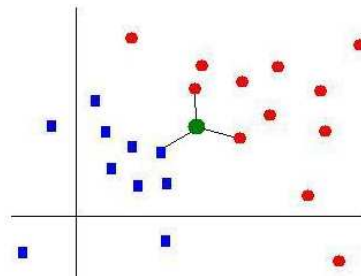
k-NN sınıflandırıcı, sınıflandırılması istenen bir noktanın veri setinde daha önceden var olan noktalardan en yakın k tanesine yakınlığına bakarak sınıflandırma yapar.

(Şekil 3.18) yeşil nokta yeni üyeyi temsil etmektedir. Kırmızı ve mavi noktalar daha önceden sınıflandırılmış olan üyelerdir.



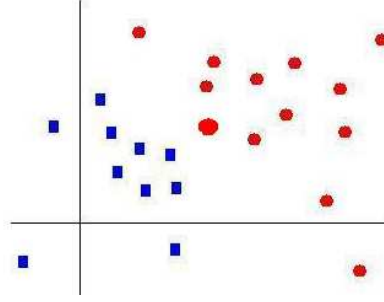
Şekil 3.18 Sınıflandırıcıya yeni üye girişi [48]

k-NN algoritması yeni üyeye en yakın olan noktaları tespit eder. Yakınlık ölçütü için literatürde birçok mesafe tanımı mevcuttur. Bunlar Öklit, Mahalanobis ve Manhattan uzaklık ölçütleridir. Genel olarak uygulamalarda Öklit uzaklık ölçütü kullanılmaktadır.



Şekil 3.19 En yakın üç uzaklığın tespit edilmesi [48]

Uzaklıklar tespit edildikten sonra en yakın üyeler belirlenir. Yeni üye de o sınıfa dahil edilir.



Şekil 3.20 Yeni üyenin bir sınıfa dahil edilmesi [48]

Genel uygulamalarda eşitlik olmaması için k sayısı tek seçilir. Bu sayede sınıflandırma algoritması kararsız kalmaz. Tez için kullanılan k-NN algoritmasında k = 4 olarak seçilmiştir. Bu sayede sistemin tanımada kararsız kaldığı harfler de tespit edilebilmektedir.

Uzaklık ölçütlerinden Öklid ve Mahalanobis sayısal tasarımlar için karmaşık olan karekök ve bölme işlemleri içermektedir. Bu nedenle zaman ve güç tüketimi çoğalmaktadır. Bunun önüne geçmek için uzaklık ölçütü olarak Manhattan uzaklık ölçütü kullanılmıştır. Bu ölçüt (3.8)' de görüldüğü üzere vektörlerin eleman eleman farklarının alınıp mutlak değerlerinin toplanması şeklinde tanımlanmaktadır.

$$d_1(\mathbf{p}, \mathbf{q}) = \|\mathbf{p} - \mathbf{q}\|_1 = \sum_{i=1}^n |p_i - q_i| \quad (3.8)$$

Sınıflandırma İşlemi

Sınıflandırma için (Ek A)' da kullanılan veri seti 7 farklı kişiden alınan harf örnekleri ile oluşturulmuştur. Veri setinin içerisinde 10 farklı takım bulunmaktadır. Her takım İngiliz alfabesinde bulunan 26 harften oluşmaktadır.

Sistemin performansı veri setinden rastgele 3 takım çıkartılıp geriye kalan 7 takım üzerinden sınıflandırma işlemi yapılarak ölçülmüştür. Veri setinden çıkartılan 3 takım test kümesi, kalan 7 takım sınıflandırıcı kümesidir. Performans tablosu (Çizelge 3.1)' de görülmektedir.

3.3.4 Sonu

Test iin kullanılan kmeler veri seti ierisinden rastgele seilmiřtir. Kullanılan 5 kmenin bařarı ortalaması % 81.54' tr. Bařarı oranı, 22.5° ve katları ara alar kullanarak ve k-NN iin k sayısının optimum deęeri bulunarak arttırılabilir.

Çizelge 3.1 Test kümeleri için başarı yüzdeleri

Referans Harfleri	TEST KÜMELERİ														
	Küme 1 [1,5,7]			Küme 2 [2,6,8]			Küme 3 [4,7,9]			Küme 4 [5, 8, 10]			Küme 5 [3, 6, 10]		
A	L+	A+	A+	A-	A-	H+	A-	A+	H+	A+	H+	L+	A-	A-	L+
B	B+	B+	R-	B+	Q+	B-	K+	B+	B-	B+	B-	B+	B+	Q+	B+
C	C+	C+	C+	O+	C+	C+	C+	C+	C+	C+	C+	C+	C+	C+	C+
D	D+	D+	D+	D+	D+	D+	D+	D+	D+	D+	D+	D+	D+	D+	D+
E	E+	E+	E+	E+	E+	E+	E+	E+	E+	E+	E+	E+	E+	E+	E+
F	F+	F+	F+	F+	F+	E-	F+	F+	F+	F+	F-	F-	F+	F+	F+
G	G+	G+	G+	G+	G+	G+	G-	G+	G+	G+	G-	G-	G+	G+	G+
H	H+	H+	H+	H+	H+	H+	H+	H+	H+	H+	H+	H+	H+	H+	H+
I	I+	I+	X+	I+	E+	W+	I+	X+	I+	I+	K+	I+	I+	E+	I+
J	J+	J+	J+	J+	O+	D+	J+	J+	J+	J+	D+	J+	J+	O+	J+
K	K+	R+	X+	K+	K+	X+	K+	H+	K+	E-	X+	K+	Y+	K+	K+
L	L+	L+	L+	L+	L+	L+	L+	L+	L+	L+	L+	L+	L+	L+	L+
M	M+	Y+	X+	M+	J+	M+	M+	X+	M+	M+	M+	M+	M+	J+	M+
N	N+	N+	N+	N+	N+	N+	N+	N+	N+	N+	N+	N+	N+	N+	N+
O	O+	O+	O+	O+	O+	O+	O+	O+	O+	O+	O+	O+	O+	O+	O+
P	P-	F+	Y+	P+	P-	P+	P+	Y+	P-	F+	P+	P-	D-	P+	Y+
Q	Q+	Q+	Q+	Q+	O+	Q+	Q-	Q+	Q+	Q+	Q+	Q+	Q+	O-	Q+
R	R+	R+	R+	R+	R+	R+	R+	R+	R+	R+	R+	R+	R+	R+	R+
S	S+	G+	A+	S+	O+	S+	S-	S+	D+	S-	S+	S+	S+	S+	S+
T	T+	T+	W+	T+	T+	A+	T+	W+	T+	T+	A+	T+	T+	T+	T+
U	U+	U+	U+	U+	U+	U+	U+	U+	X+	U+	U+	U+	U+	U+	U+
V	A+	Y+	V-	V+	U+	V+	T+	V-	V+	Y+	V+	N+	U+	V+	N+
W	I+	V+	W+	I+	A+	U+	N+	W+	N+	N+	U+	X+	W+	A+	X+
X	X+	X+	X+	X+	X+	X+	X+	X+	X+	X+	X+	X+	X+	X+	X+
Y	Y+	Y+	Y+	Y+	Y+	Y+	Y+	Y+	Y+	Y+	Y+	Y+	Y+	Y+	Y+
Z	Z+	K+	Z+	Z+	Z+	Z+	Z+	Z+	Z+	K+	Z-	Z+	Z+	Z+	Z+
Başarı %' si	88.46	73.08	73.08	92.31	73.08	73.08	88.46	80.77	84.62	80.77	76.92	88.46	88.46	76.92	84.62
Yanlış Sayısı	3	7	7	2	7	7	5	5	4	5	6	3	3	6	4
Kararsız Sayısı	1	0	1	0	1	1	1	1	2	1	4	3	0	0	0

MORFOLOJİK İŞLEMLER İLE EL YAZISI TANIMA

4.1 Giriş

Geliştirilen algoritmalar, doğruluk oranını arttırmak için ağır matematiksel temeller üzerine kurulurlar. Bunların FPGA gibi sayısal mimarilerdeki tasarımları oldukça zor ve maliyetlidir. Üçüncü bölümde önerildiği gibi Gabor-tipi filtreler kullanarak hem doğruluk oranı yüksek hem de tasarım maliyeti düşük ve hızlı sayısal sistemler gerçekleştirilmektedir. Gabor filtrelerin uzamsal tanım bölgesinde yön seçici özelliği güçlü bir öznelik çıkarımı sağlamaktadır. Harflerin yönlü bileşenlerinin çıkartılması tanıma algoritması için kullanışlı bir bilgidir. El yazılarının ikili resimler olduğu göz önünde bulundurularak, morfolojik işlemler ile daha hızlı şekilde yönlü bileşenler çıkartılabilir.

Bu bölümde HSA Gabor-tipi filtrelerinin yönlü filtreleme özelliğinden esinlenerek geliştirilen yön seçici morfolojik filtre bankaları anlatılmıştır. Bu filtre bankaları Gabor-tipi filtre bankaları gibi farklı derecelere duyarlı filtreleme yapmaktadır. Filtreleme işleminde sadece morfolojik işlemler kullanıldığı için çok daha hızlı ve tasarım maliyeti düşük bir sistem oluşturulabilmiştir. Hız ve maliyetten elde edilen bu verimlilik göz önünde bulundurularak filtrenin açı duyarlılığı 45° yerine 22.5° yapılmıştır. Bu sayede morfolojik filtre bankasının içerisinde yön çıkartma açıları 0° , 22.5° , 45° , 67.5° , 90° , 115.5° , 135° , 157.5° olmak üzere 8 ayrı açıda ve 3 ayrı uzamsal genişlikte toplam 24 filtre modülü tasarlanmıştır. Bu sayede harf üzerinden alınan farklı açıdaki bileşenlerin sayısı iki katına çıkartılmış ve doğruluk yüzdesi de arttırılmıştır. Ayrıca yazılan tüm MATLAB kodları VHDL diline uygun şekilde, hazır kod kullanılmadan sayısal tasarım mantığı ile yazılmıştır.

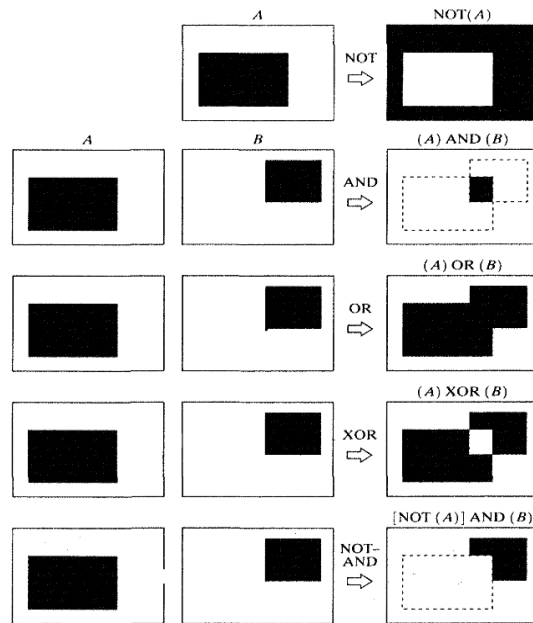
4.2 Morfoloji Matematiği

Morfoloji, kelime anlamı olarak şekil ya da biçim anlamına gelmektedir. Morfoloji daha çok biyoloji bilim dalında bitki ve hayvanların form ve yapıları ile ilgili kullanılan bir terimdir [49]. Matematiksel temeli, kümeler kuramı üzerine kurulan ve matematiksel morfolojik (MM) yöntem terimi şeklinde kullanılmaya başlanan teknik daha sonra geniş bir yöntemsel araç olarak kullanılmaya başlanmıştır. MM yöntemin kullanım alanları sadece matematiksel bir teori olması ile sınırlı kalmamış matematik, elektronik ve haberleşme gibi alanlarda çeşitli uygulamalarda yaygın olarak kullanılmaya başlanmıştır.

Matematiksel morfoloji (MM), küme teorisi üzerinden görüntülerin şekilleri üzerinde değişiklik yapmak ve şekilleri analiz etmek amacıyla kullanılan bir yöntemdir [50].

MM daha çok binary (ikili) görüntüler üzerinde uygulanmakla beraber gri seviye görüntü işleme gibi geniş bir kullanım alanına sahiptir. Yöntem özellikle görüntü işleme alanında üzerinde kullanılabilen güçlü ve bütünsel bir yaklaşım sayesinde çok çeşitli problemler üzerine uygulanmaktadır [49]. İkili görüntü verileri üzerinde seçilen bir ikili şablonun görüntüyle noktasal çarpımı şeklinde gerçekleşen işlem, yapısı çeşitli nedenlere bağlı olarak değişen ve yapısal element (Structural Element) adı verilen ikili şablonlar kullanılarak yapılmaktadır.

(Şekil 4.1)'de gösterilen küme teorisi ikili görüntüler üzerinde temel mantık işlemleri şeklinde uygulanmaktadır.

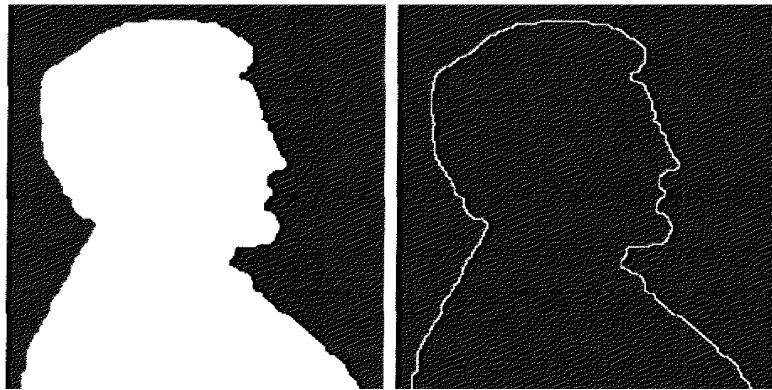


Şekil 4.1 Binary (ikili) görüntülerde temel mantık işlemleri [49]

Görüntü işlemede matematiksel morfolojik (MM) işlemler ön filtreleme, bölütleme, özellik çıkartma ve sınıflandırma olmak üzere çok çeşitli uygulamalarda kullanılmaktadır. Ayrıca morfolojik işlemler ön filtreleme ve görüntünün netleştirilmesi alanlarında; bir görüntü içindeki şekil ve bölgelerin iskelet ile sınırların belirlenmesi ve/veya tespitinde ya da bölge veya şekillerin inceltmesi, kırılması (pruning) gibi temel amaçlar için yaygın olarak kullanılmaktadır. Görüntü işleme sisteminde belirli aşamalardan geçmiş görüntülere MM işlemler kullanılarak görüntülerde hedeflenen bilgilerin (nokta, kenar, kenar birleştirme, sınır tespiti vb.) belirginleşmesi sağlanır. Ayrıca hedef bölgenin ya da nesnenin dışında bulunan nispeten küçük ölçekli, çevresel etkilerden kaynaklı gürültü bölgeleri de temel MM işlemler ile temizlenebilmektedir.

MM yöntemlerin kullanıldığı diğer alanlar arasında; Bölge doldurma (region filling), sınır tespiti (boundary extraction), çizgi takibi, aşınma (thinning), kalınlaştırma, bağlı bileşenlerin elde edilmesi (extraction of connected components), dışbükey örtü uygulaması, görüntünün iskeletini çıkartma (Skeletons), kırılma (Pruning) gibi çok çeşitli uygulama alanları bulunmaktadır.

Diğer yandan MM işlemler, görüntü işleme tabanlı sistemlerde özelliklerin çıkartılmasında kullanılan uygun ve başarılı bir tekniktir. Sınıflandırma aşamasında kullanılmak üzere görüntülere ait kullanılabilir özelliklerin elde edilmesinde MM yöntemler kullanılmaktadır [50]. Özellikle yüksek boyuttaki verilerin MM yöntemlerle işlenerek, veri miktarının indirgenmesi, bütün görüntü yerine sadece görüntünün sınır bölgelerinin kullanılabilmesi (Şekil 4.2) gibi avantajları sayesinde MM yöntemler sistemler açısından işlem yoğunluğunu azaltmakta ve daha etkili tasarımlara olanak sağlamaktadır.



Şekil 4.2 MM yöntemler kullanılarak görüntüye ait sınır bölgelerinin tespiti

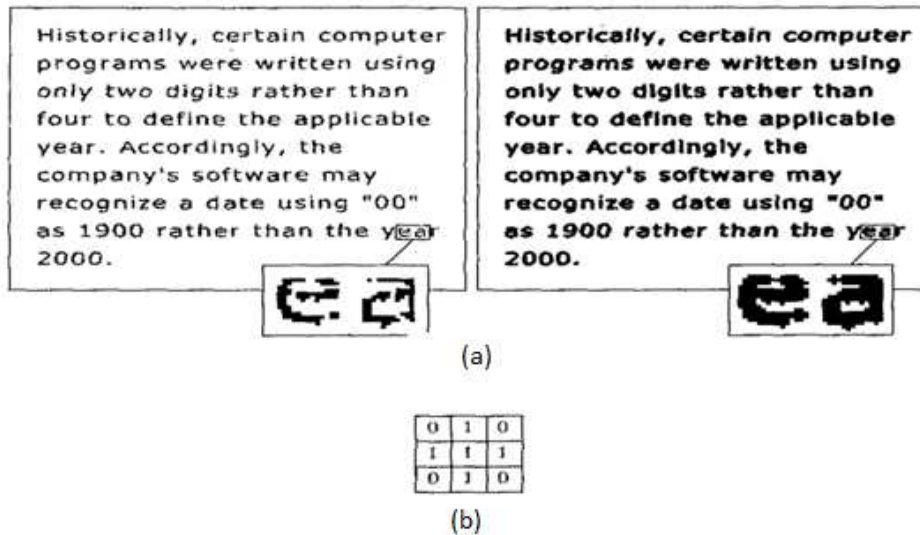
MM görüntü işleme tekniği, ikili seviye görüntülerde yaygın olarak kullanılmaktadır (Şekil 4.1). MM görüntü işleme tekniklerinin kullanım alanları genişlemekte ve son yıllarda gri seviye görüntüler üzerinde de kullanılmaya başlanmıştır [49]. İkili seviye MM’de kullanılan temel fonksiyonlar, genişletme (dilation), aşındırma (erosion), açma (opening) ve kapama (closing)’dır [49]. Diğer yandan Açma ve Kapama işlemleri gibi diğer işlemler, Genişletme ve İnceltme işlemlerinden türetilmektedir [49].

4.2.1 Genişletme İşlemi

Daha önce değinildiği gibi MM görüntü işleme tekniğinde kullanılan temel iki işlem bulunmaktadır. Genişletme (Dilation) ve Aşındırma (Erosion) olarak adlandırılan bu yöntemler çeşitli alanlarında yaygın bir biçimde kullanılan MM’ye ait diğer yöntemlerin temelini oluşturmaktadır. Genişletme ve Aşındırma işlemleri uygun bir yapısal element seçilerek hem gri seviye hem de ikili seviye görüntü matrisleri üzerinde uygulanabilmektedir.

Genişletme işlemi, A ve B iki boyutlu uzay Z^2 'de tanımlı olmak üzere (4.1)' de ki şekilde tanımlanır. Burada A işlenecek görüntü matrisi ve B ise uygun şekilde belirlenmiş bir yapısal elementtir.

$$A \oplus B = \{z | (\hat{B})_z \cap A \neq \emptyset\} \quad (4.1)$$



Şekil 4.3 MM genişletme işlemi uygulama örneği
a) Görüntü matrisleri b) Yapısal element

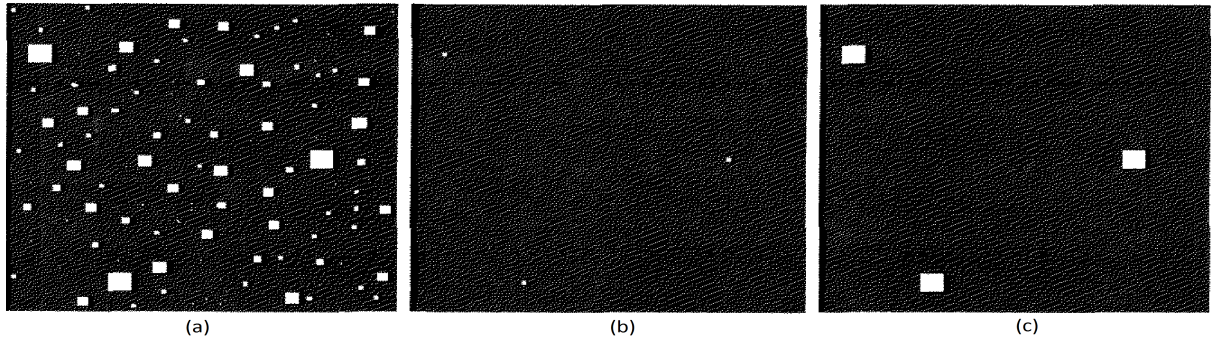
Genişletme işlemi, (Şekil 4.3)'te gösterildiği gibi yaygın olarak görüntünün içindeki kopuk bölge ya da nesnelerin bileştirilmesinde kullanılmaktadır.

4.2.2 Aşındırma

MM yöntemlerin bir diğer temel işlemi olan Aşındırma işlemidir. A görüntü matrisi ve B yapısal elementi Z^2 iki boyutlu uzayında tanımlanan matrisler olmak üzere, A görüntüsünün B yapısal elementi tarafından Aşındırma işlemi,

$$A \ominus B = \{z | (\hat{B})_z \subseteq A\} \quad (4.2)$$

şeklinde uygulanmaktadır. MM işlemler görüntü işleme alanında hem gri seviye hem de ikili seviye görüntülerde kullanıldığından dolayı probleme göre çeşitli yapısal elementler seçilmektedir [49]. Yapısal elementler 3x3, 5x5, 7x7 ... boyutlarında seçilebilmektedir.



Şekil 4.4 a) MM ile 1, 3, 5, 7, 9, 15 boyutundaki karelerden oluşan görüntünün işlenmesi b) 13 piksel boyutunda yapısal element ile Aşındırma işlemi görüntü c) aynı yapısal element ile genişletme işlemi

Morfolojide kullanılan Genişletme ve Aşındırma işlemleri birçok görüntü işleme algoritmasında kullanılmaktadır. Ayrıca bu işlemler kullanılarak açma (opening) ve kapama (closing) işlemleri türetilmiştir.

4.2.3 Türetilen İşlemler

Açma işlemi, aşındırma işleminin ardından genişletme işlemi uygulanarak türetilmiştir. Matematiksel tanımı (Eşitlik 4.3)' te verilmiştir. Birbirine yakın olan nesnelerin arasını açmak ya da yapısal elemanın büyüklüğüne göre görüntü içerisindeki boyutu küçük olan nesnelere temizlemek için kullanılır [10]. (Şekil 4.4) buna güzel bir örnektir.

$$A \circ B = (A \ominus B) \oplus B \quad (4.3)$$

Kapama işlemi, genişletme işleminden sonra aşındırma işleminin uygulanmasıdır. Bu işlem ile görüntü içerisindeki ana hatlar daha belirginleşir. Yakın olan nesnelere birbirine bağlanır.

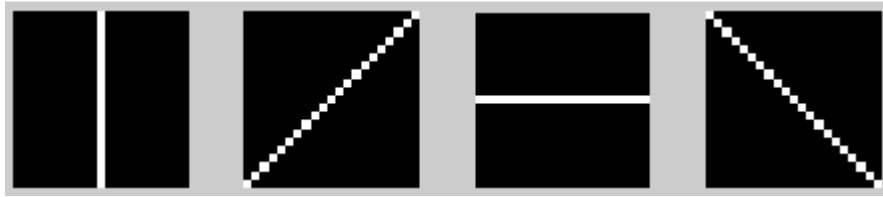
$$A \cdot B = (A \oplus B) \ominus B \quad (4.4)$$

Açma işlemindeki yapısal elemanın şekline göre görüntü içerisindeki küçük nesnelerin temizlenmesinden esinlenerek, bu işlem ile ikili görüntüler üzerindeki nesnelerin yönlü bileşenlerinin çıkartılması fikri geliştirilmiştir. Bu sayede ikili görüntüdeki nesnelerin yönlü bileşenleri mantıksal operatörler ile hızlı bir şekilde çıkartılabilmektedir. Sadece mantıksal işlemler kullanıldığından dolayı sayısal tasarımlar için maliyeti düşük hızlı ve yüksek performansta bir algoritma geliştirilmiştir.

4.3 Morfoloji İle Yönlü Filtreleme

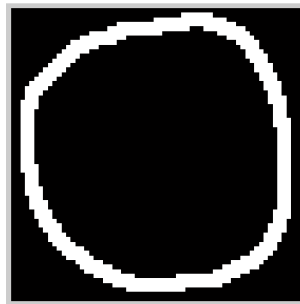
İkili görüntüler için açma işlemi, yapısal elemanın (YE) şekline göre bir filtreleme metodu olarak da düşünülebilir. Açma işleminde ilk aşamada aşındırma işlemi kullanılır. Aşındırma işleminin temel mantığı, görüntü içerisinde YE' nin şeklinde bir yapının olup olmadığıdır. Eğer varsa YE' nin merkez pikseli "1" yoksa "0" olur.

YE' nin şekli eğer yönlü bir biçimde seçilirse açma işlemi ile nesnenin o yöndeki bileşenleri korunacak, diğer yöndeki bileşenleri ise aşındırılacaktır.

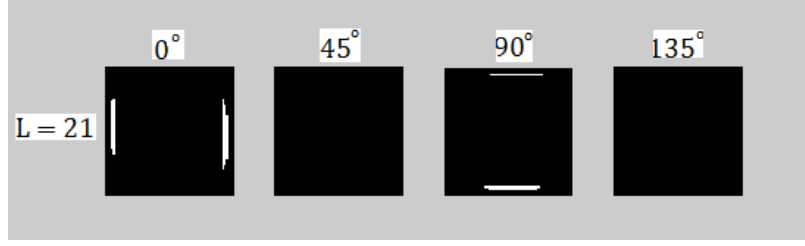


Şekil 4.5 L = 21 uzunluklu YE

(Şekil 4.5)' te L = 21 ve dereceleri 0, 45, 90, 135 olmak üzere dört farklı YE görülmektedir. Bu YE' ler ile (Şekil 4.6)' da ki "O" harfi üzerine açma işlemi uygulandığında (Şekil 4.7)' deki sonuçlar elde edilmektedir. Gabor filtresinin çıkışına benzer şekilde sadece YE' lerin açısındaki bileşenlerin süzülmesi görülmektedir.

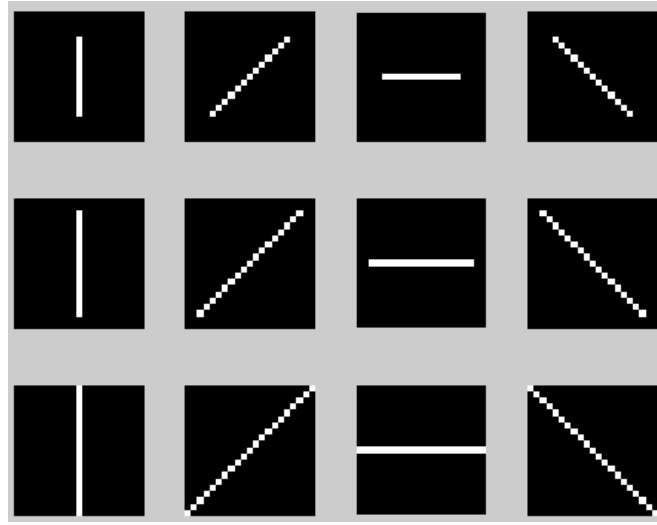


Şekil 4.6 El Yazısı "O" Harfi



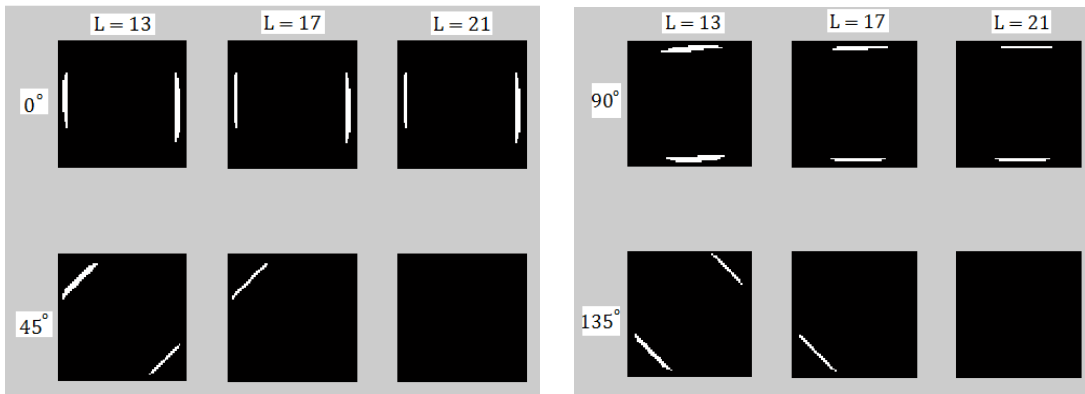
Şekil 4.7 L = 21 olan YE' ler için "O" harfi üzerine uygulanmış açma işlemi

Bu düşünce ile Gabor filtre bankalarına benzer işlem yapan Morfolojik filtre bankaları tasarlanabilir. Örnek olarak (Şekil 4.8)' de L = 13, L = 17 ve L = 21 boyutlarında ve 0°, 45°, 90° ve 135° açılarında 12 ayrı YE oluşturulmuş ve bir morfolojik filtre bankası tasarlanmıştır.



Şekil 4.8 Tasarlanan Morfoloji filtre bankası

"O" harfi, filtre bankasındaki her bir YE ile ayrı ayrı filtrelendikten sonra (Şekil 4.9)' da ki çıkışlar elde edilmiştir.



Şekil 4.9 Filtre bankasının çıkışları

Geliştirilen bu algoritma, yönlü bileşenleri çıkartması açısından harf tanıma için oldukça elverişlidir. Ayrıca tasarım maliyetinin düşük ve hızının yüksek olması göz önünde bulundurulduğunda, algoritmanın tanıma performansını yükseltmek için açı çözünürlüğü artırılabilir. Bu amaç doğrultusunda harf tanıma için tasarlanan filtre bankasında şuan kullanılan açılar yanında 22.5° , 67.5° , 115.5° ve 155.5° açılı filtreler de kullanılmıştır. Harf tanıma için geliştirilen filtre bankasında 3 farklı boyutta ve 8 farklı açılı YE kullanılmak üzere toplam 24 filtre kullanılmıştır.

4.4 Morfolojik Filtreleme İle Harf Tanıma

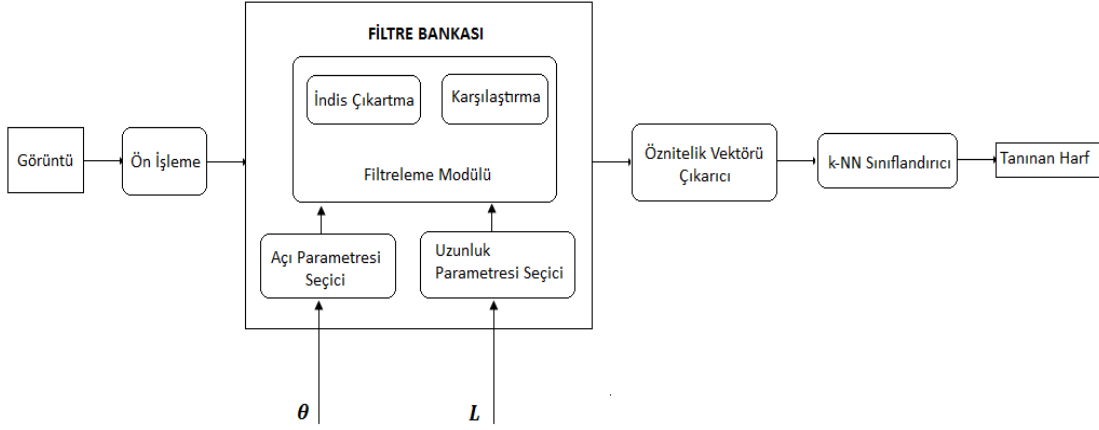
Sayısal sistemler ile insanlar arasında iletişim kurulabilmesinin ve el yazısı metinlerin tanımlanıp sayısal bir ortama aktarılmasının bel kemiğini harf tanıma algoritmaları oluşturmaktadır. Bu algoritmalarından beklenen en önemli performans tanımaya yöneliktir. Tanımının yanında hız performansının da yüksek olması istenilen bir diğer özelliktir.

Gabor filtreler ile tasarlanan algoritmaların doğruluk oranları oldukça yüksektir. Fakat işlem karmaşıklığından dolayı hızları mantıksal işlemlerin yanında ciddi oranda düşüktür. Tasarlanan morfolojik filtreleme algoritmasında Gabor filtrelerinin yönlü bileşenleri tespit ederek öznitelik çıkarmadaki gücü ile mantıksal işlemlerin hızlı işlem gücü birleştirilmiş ve bu yeni algoritma harf tanıma için kullanılmıştır.

Oluşturulan harf tanıma algoritması 4 ana işlem basamağından oluşmaktadır.

- Ön işleme
- Filtreleme
- Öznitelik Vektörünün Çıkartılması
- Sınıflandırma

Ön işleme basamağı haricindeki basamaklarda işlemler, MATLAB' ın konvolüsyon gibi hazır fonksiyonları kullanılmadan VHDL diline ve sayısal tasarım mantığına uygun olarak yazılmıştır.



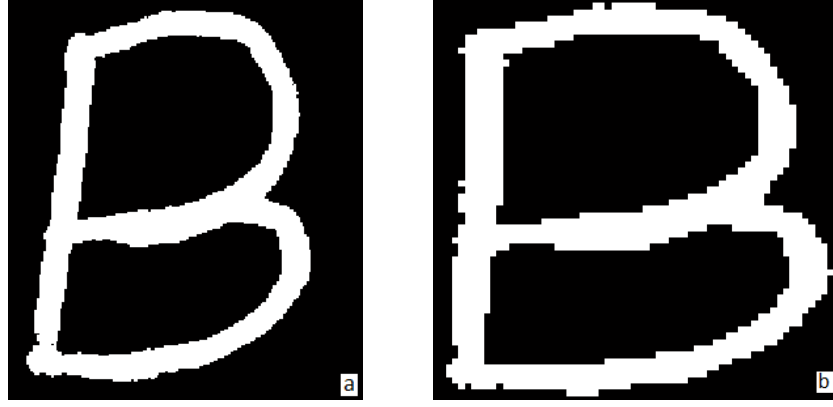
Şekil 4.10 Tasarlanan sistemin blok diyagramı

(Şekil 4.10)' da tasarlanan sistemin blok diyagramı verilmiştir. Dışarıdan alınan görüntü ön işlemeden geçirildikten sonra filtre bankası bloğunda filtrelenir. Burada elde edilen filtrelenmiş görüntülerden, öznitelik çıkartma bloğunda harfin özniteliği çıkartılır. Son aşama olarak sınıflandırma bloğunda ise k-NN sınıflandırıcı ile harf tespiti yapılır.

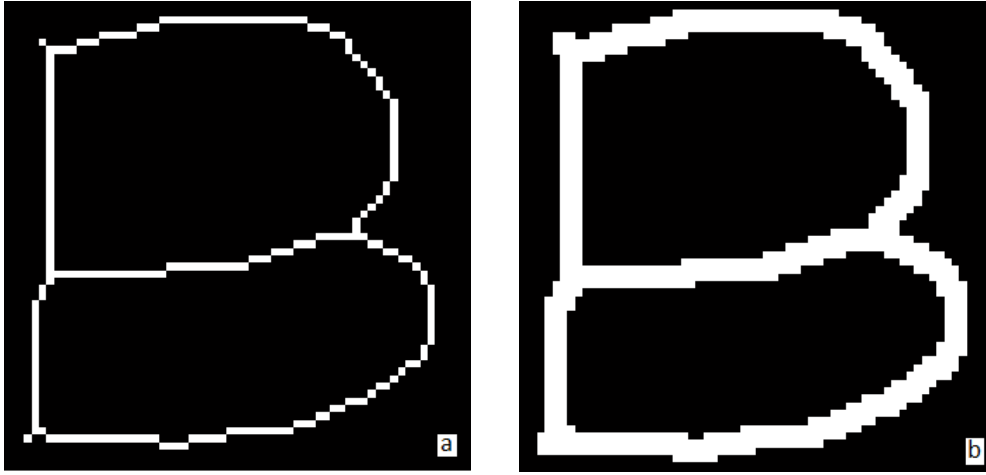
4.4.1 Ön İşleme

(Bölüm 3.3)' te anlatılan HSA Gabor-tipi filtreler ile tasarlanan algoritmanın ön işleme kısmındaki işlemler, morfolojik filtreleme algoritmasının ön işleme kısmında da kullanılmıştır. Fark olarak Gabor-tipi filtrelere giren görüntünün piksel değerleri beyaz ton için "-1" siyah ton için "1" iken morfolojik filtrelemeden önce bu değerler ikili (binary) görüntü türündeki gibi beyaz ton için "1" siyah ton için "0" olarak kalmaktadır. Ayrıca giriş görüntüsünün boyutunu 64x64 sabit boyuta getirmek için NSN dönüşüm yerine MATLAB programının hazır fonksiyonu kullanılmıştır.

Morfolojik filtrelemeden önce ön işleme bloğunun içerisinde giriş görüntüsünün boyut sabit bir boyuta ayarlanmış, görüntü içerisindeki harfin eğimi düzeltilmiş ve iskeleti çıkartılıp harfin kalınlığı sabitlenmiştir. Bu basamaklar (Şekil 4.11) ve (Şekil 4.12)' te gösterilmektedir.



Şekil 4.11 a) Eğik ve boyutu ayarlanmamış harf b) Eğikliği giderilmiş [64 x 64] boyutlarında olan harf

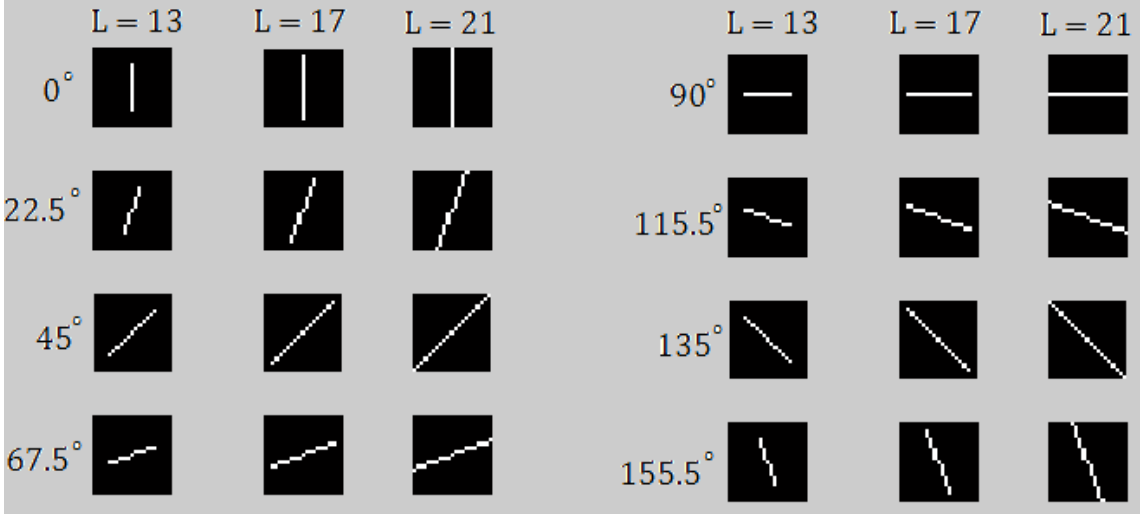


Şekil 4.12 a) İskeleti çıkarılmış harf b) YE ile genişletilmiş harf

Harfin kalınlığının sabit olması sınıflandırma adımı için oldukça önemlidir. Yapılan denemelerde görüldüğü üzere harfin kalınlığının değişmesi tanıma işlemini zorlaştırmaktadır. Sabit kalınlık ile çalışmak doğruluk oranını arttırmaktadır.

4.4.2 Filtreleme

Bu basamakta ön işlemeden geçirilen harfin yönlü bileşenleri çıkartılmaktadır. Filtreleme bloğunun içerisinde 3 farklı boyutta ve 8 farklı açıda 24 YE ile tasarlanmış morfolojik filtreler bulunmaktadır. Bu YE'ler (Şekil 4.13)' te görülmektedir.



Şekil 4.13 Morfolojik filtre bankası için tasarlanan 24 farklı YE
(Açılar YE' nin dikey eksen ile yaptığı açığa göre yazılmıştır)

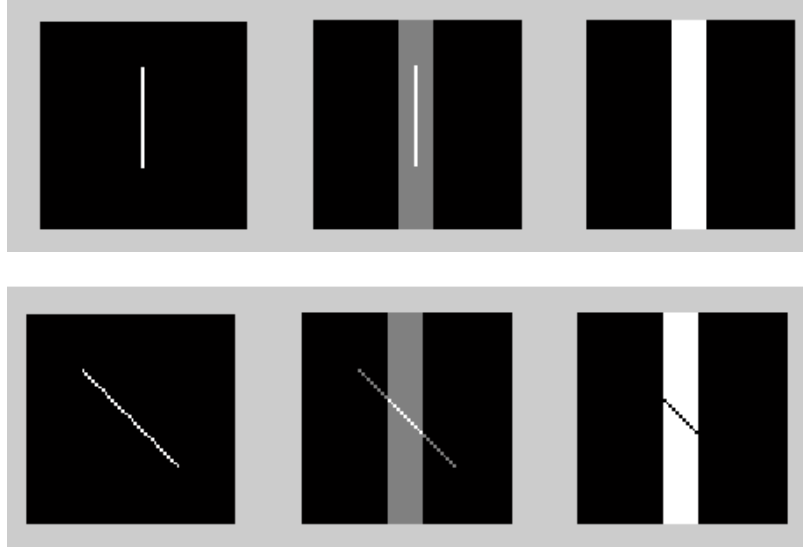
Yapısal elemanların (YE) merkez pikselleri olabilmesi için uzunlukları tek sayı seçilmiştir. Ayrıca ara açılardaki YE' ler tasarlanırken 22.5° ' nin eğimine en yakın değeri verecek olan uzunluk ölçüleri göz önünde bulundurulmuştur. Bu değerler sırası ile ilk değer yatay ikinci değer düşey uzunluk olmak üzere [5,13], [7,17] ve [9,21]' dir. Bu ölçülerdeki YE' ler sabit bir algoritmaya göre üretilebilmektedir. 22.5° için tasarlanan YE'nin, y-eksenine, x-eksenine ve orijine göre simetrisi alınarak 67.5° , 115.5° ve 155.5° açılarındaki YE' ler türetilmiştir.

Filtreleme İşlemi

Filtreleme işlemi (Şekil 4.10)' da görülen filtreleme modülünde gerçekleşir. İlk olarak açı seçici ve uzunluk seçiciden filtremeyi yapacak olan YE' nin özellikleri seçilir. Açık parametrelerine ve uzunluk parametresine göre YE üretilir. İndis çıkartma ve karşılaştırma modülü ile YE görüntü üzerinde gezdirilerek açma işlemi ile filtreleme yapılır.



Şekil 4.14 Filtre için giriş görüntüsü



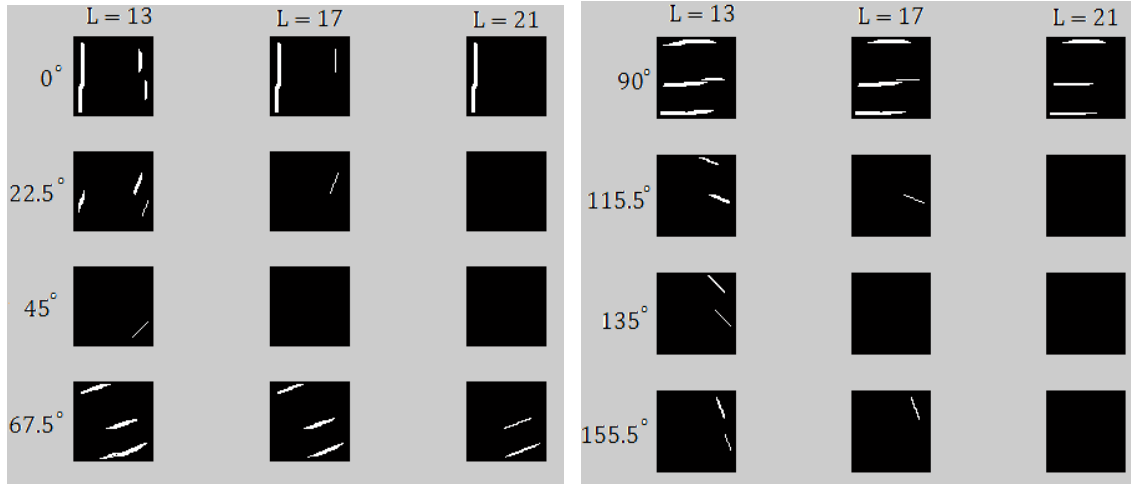
Şekil 4.15 0° (üst) ve 135° (alt) için filtreleme

(Şekil 4.14)' te ki görüntü filtre girişine verilmiş olsun. İlk olarak görüntü üzerinde filtrelenecek istenen yer için merkez piksel üretilir. Burada ki örnek için (32,32) pikseli merkez piksel olarak alınmıştır. İndis çıkartıcı (32,32) pikseline komşu olan ve YE' nin şekline uyan indisleri çıkartır. Çıkarılan indislerin görüntüde karşılık gelen değerleri toplanır. Karşılaştırma modülünde toplanan değer YE' nin boyu ile karşılaştırılır. Eğer bu değer YE' nin boyundan küçükse çıkarılan indislerin görüntü içerisindeki değerleri sıfır, YE boyu kadar ise değerler bir olarak kalır.

(Şekil 4.16)' de ise filtre bankasına giriş olarak "B" harfinin görüntüsü verilmiştir. Bu görüntü 24 ayrı YE ile filtrelendikten sonra (Şekil 4.17)' de ki çıkışlar elde edilmiştir.



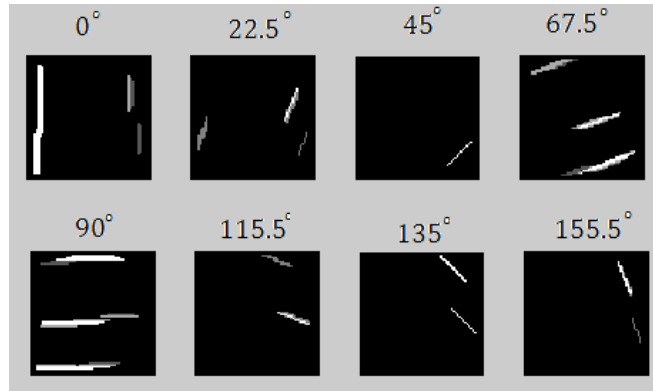
Şekil 4.16 Filtrelenecek olan "B" harfi görüntüsü



Şekil 4.17 “B” harfinin filtre çıktıları

4.4.3 Öznitelik Vektörünün Çıkarımı

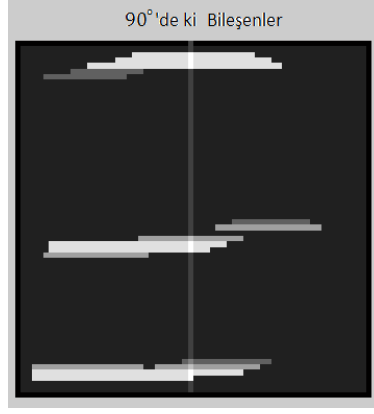
Filtre çıktılarından elde edilen aynı açılı görüntüler birbirleri ile tam sayı formatında toplanır. Bu sayede aynı açıdaki YE' nin üç ayrı uzunluk filtresinden geçebilen bileşenlerin piksel değeri 3, iki uzunluk filtresinden geçenlerinki 2, sadece bir filtreden geçebilenlerin ise 1 olacaktır. Toplanan görüntülerden 8 görüntü elde edilir. Bu görüntüler sadece açı değişkenine göre birbirlerinden farklıdır.



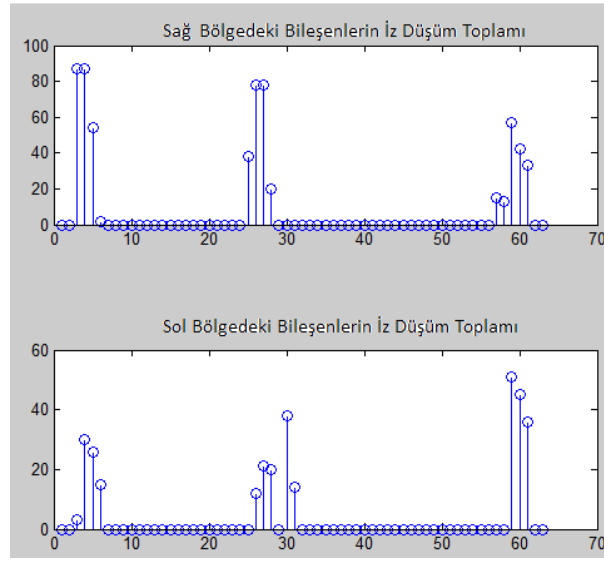
Şekil 4.18 Aynı açıdaki filtre çıktılarının tam sayı formatında toplanması ile oluşan görüntüler

Elde edilen 8 görüntü, 8 farklı açıdan elde edilen bileşenlerden oluşmaktadır. Bu görüntüler ikili görüntü formatında olmayıp tam sayı formatındadır.

Görüntüler içerdikleri bileşenlere dik bir eksen ile ikiye ayırılır. Ayrılan görüntüler eksen üzerine ayrı ayrı iz düşümü toplamı ile toplanır.



Şekil 4.19 İz düşüm eksenini ve 90° de ki bileşenler



Şekil 4.20 Ayrılan sağ ve sol bölgelerin iz düşüm toplamları

Elde edilen iz düşüm vektörlerinin uzunlukları sabit 64 birimdir. İz düşüm vektörleri [1,21], [22, 42] ve [42,64] olmak üzere üç bölgeye ayrılmıştır. Bu bölgelerdeki değerlerde (bileşen öbeği) kendi içlerinde toplanıp 64 uzunluğundaki vektörlerden 3 uzunluklu vektörler elde edilmiştir. Toplama işlemi sınırlar göz önünde bulundurularak adaptif olarak yapılmıştır. Örneğin bir bileşen öbeğinin bir kısmı bir bölgede bir kısmı diğer bölgede kalacak şekilde bir yere gelmiş olabilir. Böyle bir durumda öbeğin bölgelerdeki kısımları ayrı ayrı toplanır. Toplamın küçük olduğu bölgedeki öbek diğer bölgeye dahil edilir.

8 görüntüden toplam 16 iz düşümü vektörü elde edilir. İz düşüm vektörleri de kendi içlerinde üç ana bölgede toplanarak bir harften uzunluğu 48 olan öznitelik vektörü çıkartılır. Çıkartılan öznitelik vektöründeki değerlerin, filtre modülünün girişine gelen görüntü içerisinde ki değeri "1" olan piksellerin toplam sayısı üzerinden yüzdeleri hesaplanır. Bu

sayede bileşenlerin harf içerisindeki ağırlıkları çıkartılmış olur. Elde edilen bu ağırlık vektörü sınıflandırma için kullanılır.

4.4.4 Sınıflandırma

(Bölüm 3.3.3)' te anlatılan k-NN sınıflandırma algoritması $k = 4$ olarak bu algoritmada da kullanılmıştır.

Sınıflandırma İşlemi

Sınıflandırma için kullanılan veri seti 7 farklı kişiden alınan harf örnekleri ile oluşturulmuştur. Veri setinin içerisinde 10 farklı takım bulunmaktadır. Her takım İngiliz alfabesinde bulunan 26 harften oluşmaktadır.

Sistemin performansı veri setinden rastgele 3 takım çıkartılıp geriye kalan 7 takım üzerinden sınıflandırma işlemi yapılarak ölçülmüştür. Veri setinden çıkartılan 3 takım test kümesi, kalan 7 takım sınıflandırıcı kümesidir. Performans tablosu (Çizelge 4.1)' de görülmektedir.

4.4.5 Sonuç

Test için kullanılan kümeler veri seti içerisinde rastgele seçilmiştir. Kullanılan 5 kümenin başarı ortalaması % 91.40' tır. Başarı oranı, k-NN algoritmasında optimum k sayısı bulunarak ve farklı uzaklık ölçütleri kullanılarak arttırılabilir. Bu çalışmada morfolojik işlemler ile ikili görüntüler için sayısal tasarımlara uygun yön seçici filtreler geliştirilmiştir. Bu filtrelerin tasarım maliyeti oldukça düşük bunun yanında hızlarının yüksek olacağı öngörülmüştür.

Çizelge 4.1 Test kümeleri ve doğruluk yüzdeleri

Referans Harfleri	TEST KÜMELERİ														
	Küme 1 [1,5,7]			Küme 2 [2,6,8]			Küme 3 [4,7,9]			Küme 4 [5, 8, 10]			Küme 5 [3, 6, 10]		
A	A+	A+	A-	A+	A-	A+	H+	A-	H+	A+	A+	A+	A+	A+	A+
B	B-	B+	B+	B+	B+	G+	S+	B+	B+	B+	B+	G+	B+	C+	G-
C	C+	C+	C+	O+	O+	C+	C+	C+	C+	C+	C+	C+	C+	C-	C+
D	D+	D-	S+	D+	D+	D+	D+	I+	D+	P-	D+	D+	D+	D-	D+
E	E+	E+	E+	E+	E+	E+	E+	E+	E+	E+	E+	E+	E+	E+	E+
F	F+	F+	F+	F+	F-	F-	F+	F+	F+	F+	E+	E+	F+	F-	F-
G	G+	G+	G+	G+	G+	G+	G+	G+	G+	G+	G+	G+	G+	G+	G+
H	H+	H+	H+	H+	H+	H+	H+	H+	H+	H+	H-	H+	H+	H+	H+
I	I+	I+	I+	I+	I+	I+	I+	I+	I+	I+	I+	I+	I+	I+	I+
J	J+	J+	J+	J+	J+	J+	J+	J+	J+	J+	J+	J+	J+	J+	J+
K	K+	K+	R+	K+	K+	V+	K+	A+	K+	K+	V+	K-	A+	K+	K+
L	L+	L+	L+	L+	L+	L+	L+	L+	L+	L+	L+	L+	L+	L+	L+
M	M+	M+	M+	M+	M+	M+	M+	M-	M+	M+	M+	M+	M+	M+	M+
N	N+	N+	N+	N+	N+	N+	N+	N+	N+	N+	N+	N+	N+	N+	N+
O	O+	O+	O+	O+	O+	O+	O+	O+	O+	O+	O+	O+	O+	O+	O+
P	P+	P+	A+	P+	P+	P+	P+	A+	P+	P+	P+	P+	P+	P+	P+
Q	Q-	O+	O-	O+	Q+	Q+	Q+	Q+	Q+	Q+	Q+	Q+	Q+	Q+	Q+
R	R+	R+	R-	R+	R+	R+	R+	R-	R+	R+	R+	R+	R+	R+	R+
S	S+	S+	S+	S+	S+	S+	S+	S+	S+	S+	S+	S+	S+	S+	S+
T	T+	T+	A+	T+	T+	T+	T+	A+	T+	T+	T+	T+	T+	T+	T+
U	U+	U+	U-	U+	U+	U+	U+	U-	A+	U+	U+	U+	U+	U+	U+
V	V+	V+	V+	V+	V+	V+	V+	V+	V+	V+	V+	V+	V+	V+	V+
W	W+	W+	W+	W+	W+	W+	W+	W+	W+	W+	W+	X+	W+	W+	X+
X	X+	X+	X+	X+	X+	X+	X+	X+	X+	X+	X+	X+	X+	X+	X+
Y	Y+	Y+	Y+	Y+	Y-	Y-	Y+	A+	V+	Y+	Y+	Z+	Y+	Y-	Z+
Z	Z+	Z-	Z+	Z+	Z+	Z+	Z+	Z+	Z+	Z+	Z+	Z+	Z+	Z+	Z+
Başarı %' si	100.00	96.15	80.77	92.30	96.15	92.30	92.30	80.77	88.46	96.15	92.30	82.62	96.15	96.15	88.46
Yanlış Sayısı	0	1	5	2	1	2	2	5	3	1	2	4	1	1	3
Kararsız Sayısı	2	2	3	0	3	2	0	4	0	0	1	1	0	4	1

SONUÇ VE ÖNERİLER

Bu tezde görüntü işlemenin en temel problemlerinden biri olan el yazısı karakter tanıma için hızlı ve VLSI tasarımlara uygun iki algoritma önerilmiştir. 7 farklı kişiden alınan 10 set halinde bulunan harfler üzerinden algoritmalar test edilmiştir.

El yazısı karakter tanıma kendine özgü birçok zorlukları olmasından dolayı üzerinde yoğun olarak çalışılan konuların başında gelir. Bunun için literatürde doğruluk oranları yüksek matematiksel dönüşümleri temel alarak kurulmuş birçok algoritma mevcuttur. Bu algoritmaların her ne kadar tanıma oranları yüksek olsa da hız ve sayısal sistemlerde tasarımları açısından oldukça yavaş ve karmaşıktırlar.

El yazısı tanımaya yönelik bir diğer algoritma çeşidi ise harfin şekilsel özelliklerine bakılarak öznitelik çıkartma yöntemidir. Bu algoritmalar harflerin istatistiksel özelliklerine bakarak öznitelik çıkartırlar. Bu nedenle sayısal tasarımlar için oldukça uygundur. Ayrıca işlem yükü açısından matematiksel dönüşüm temelli algoritmalara göre oldukça düşüktür. Tanıma oranlarının alt seviyelerde kalması bu tür algoritmaların başlıca problemidir.

Bu çalışmada şekilsel algoritmaların hızlı ve sayısal tasarıma elverişli yapısı HSA Gabor-tipi filtrelerinin güçlü öznitelik çıkartma becerisi ile birleştirilmiştir. Bu sayede FPGA gibi donanımlarda rahatlıkla gerçekleştirilebilen, hızlı ve doğruluk oranı %80' leri bulan bir algoritma geliştirilmiştir. %80 harf tanıma için orta seviyede bir başarıdır. Eğer kullanılan algoritmada açısı sayısı 4' ten 8' e çıkartılırsa bu oranın artacağı öngörülmektedir. Ayrıca sınıflandırma bloğunda kullanılan k-NN algoritmasında ki farklı uzaklık ölçütleri de kullanılarak harf tanıma algoritmasının başarı oranı artırılabilir.

Önerilen diğer şekilsel algoritmada ise HSA Gabor-tipi filtrelerin yön seçici özelliğinden esinlenerek geliştirilen Morfolojik yönlü filtreleme ile el yazısı harflerin açılı bileşenleri HSA Gabor-tipi filtrelere göre çok daha az işlem yükü ile çıkartılabilmektedir. Bu algoritmanın temel mantığında işlenen görüntülerin ikili (binary) türde olmasıdır. İşlem yükünün düşmesi ile algoritmaya 22.5° nin katları olan ara açılar eklenmiş ve doğruluk oranının %91.40 seviyesine kadar yükseldiği görülmüştür. Ayrıca MATLAB ortamında yazılan programın önemli bir bölümü VHDL diline uygun hazır kod kullanılmadan yazılmıştır.

Bu çalışmaya ek olarak tasarlanan algoritmalar FPGA donanımında gerçekleştirilebilir. Bu sayede performansları daha doğru ölçülür. Bununla beraber k-NN haricinde FPGA donanımına uygun farklı sınıflandırıcılar da denenecek başarı oranları yukarı seviyeye çekilebilir.

KAYNAKLAR

- [1] Kırılı Ö., (2006). El Yazısı Karakterlerden Karakter ve Yazıcı Tanıma, Yüksek Lisans Tezi, ESOGÜ Fen Bilimleri Enstitüsü, Eskişehir
- [2] Arıca N., (1998). Serbest Stil El Yazısı İçin Çevrim Dışı Karakter Tanıma Sistemi, Yüksek Lisans Tezi, ODTÜ Fen Bilimleri Enstitüsü, Ankara.
- [3] Bhattacharya, U., Vajda, S., Mallik, A., Chaudhuri, B. B. ve Belaid A., (2004), "On the Choice of Training Set, Architecture and Combination Rule of Multiple (MLP) Classifiers for Multiresolution Recognition of Handwritten Characters", IEEE Proceedings of Workshop on Frontiers in Handwriting Recognition (IWFHR-9).
- [4] Bilişim Ansiklopedisi, El Yazısı Tanıma, <http://www.cs.bilkent.edu.tr/~duygulu/papers/BilisimAnsiklopedisi.pdf> ", 2012
- [5] Fu, H-C. ve Chen, C. M., (1996), "Recognition of Handwritten Similar Chinese Characters by Neural Networks", IEEE Signal Processing Society Workshop, 320-329.
- [6] Çapar A., Taşdemir K., Kılıç Ö. ve Gökmen M., (2002)," Türkçe El Yazısı Tanıma Sistemlerinde Öznitelik Çıkarma ve Sınıflandırma Yöntemlerinin Karşılaştırılması", Uluslar arası Radyo Bilimleri Birliği (URSI)
- [7] Kumar, M., Jindal, M.K. ve Sharma, R.K., (2011) "k-nearest neighbor based offline handwritten Gurmukhi character recognition", Image Information Processing (ICIIP), 2011 International Conference
- [8] Ye X., Wei Z., (2010). "On a Clustering Method for Handwritten Digit Recognition", 2010 Third International Conference on Intelligent Networks and Intelligent Systems
- [9] Chen Y., Liang C., Peng L. ve Zhong X., (2010). "A Handwritten Character Recognition Algorithm based on Artificial Immune", 2010 International Conference on Computer Application and System Modeling (ICCASM 2010)
- [10] Liu H. ve Ding X., (2005)," Handwritten Character Recognition Using Gradient Feature and Quadratic Classifier with Multiple Discrimination Schemes", Proceedings of the 2005 Eight International Conference on Document Analysis and Recognition (ICDAR'05)
- [11] Pal U., Wakabayashi T. ve Kimura F., (2009)." Comparative Study of Devnagari Handwritten Character Recognition using Different Feature and Classifiers", 2009 10th International Conference on Document Analysis and Recognition

- [12] Panggabean M. ve Ronningen L. A., (2009).” Character Recognition of the Batak Toba Alphabet Using Signatures and Simplified Chain Code”, 2009 IEEE International Conference on Signal and Image Processing Applications
- [13] Yuen H., (1996).” A Chain Coding Approach For Real-Time Recognition of On-line Handwritten Characters”, Acoustics, Speech, and Signal Processing, 1996. ICASSP-96.
- [14] Kim M. ve Kwon Y., (1996).” Multi-Font and Multi-Size Character Recognition based on the Sampling and Quantization of an Unwrapped Contour”, Pattern Recognition, 1996., Proceedings of the 13th International Conference
- [15] Liu L., Xia W., Jin L., Mao H. ve Tian F., (2010).” A Kai Style Contour Beautification Method for Chinese Handwriting Characters”, Systems Man and Cybernetics (SMC), 2010 IEEE International Conference
- [16] Saatçi E. ve Tavşanoğlu V., (2002). “Multiscale Handwritten Character Recognition Using CNN Image Filters”, Neural Networks, 2002. IJCNN '02. Proceedings of the 2002 International Joint Conference
- [17] Tavşanoğlu V. ve Saatçi E., (2006).” Feature Extraction for Character Recognition Using Gabor-type Filters Implemented by Cellular Neural Networks”, 2000 6m IEEE International Workshop on Cellular Neural Networks and Their Applications proceedings
- [18] Saatçi E., Cesur E., Tavşanoğlu V. ve Kale I., (2007). “An FPGA Implementation Of 2-D CNN Gabor-Type Filter”, Circuit Theory and Design, 2007. ECCTD 2007. 18th European Conference
- [19] Cesur E., Yıldız N. ve Tavşanoğlu V., (2011). “An Improved FPGA Implementation of CNN Gabor-type Filters”, Circuits and Systems (ISCAS), 2011 IEEE International Symposium
- [20] Polat A. (2007), Hücresel Sinir Ağları ve Yön Seçici Gabor Süzgeçleri, Yüksek Lisans Tezi, YTU Fen Bilimleri Enstitüsü, İstanbul
- [21] Sampaio W.B., Diniz E.M., Silva A.C. ve de Paiva A.C., (2009).”Detection of Masses in Mammograms Using Cellular Neural Networks, Hidden Markov Models and Ripley's K Function”, Systems, Signals and Image Processing,. 16th International Conference
- [22] Oniga S., Tisan A., Mic D., Buchman A. ve Vida-Ratiu A., (2008). “Optimizing FPGA implementation of Feed-Forward Neural Networks”, Optimization of Electrical and Electronic Equipment,. 11th International Conference
- [23] Kayabol K., (2002), Hücresel Sinir Ağları ve Görüntü İşlemedeki Uygulamaları, Yüksek Lisans Tezi, IU Fen Bilimleri Enstitüsü, İstanbul
- [24] Chua L. O. ve Roska T., (2004). Cellular Neural Networks and Visual Computing, second edition, Cambridge University Press, Cambridge
- [25] Tavşanoğlu V., (2000).” Cellular Neural Networks and Applications to Image Processing”, IEEE Sinyal İşleme ve İletişim Uygulamaları Konferansı (SIU) 2000.
- [26] Tavşanoğlu V., (2009).” Spatio-Temporal Filters”, PHYSCON 2009.

- [27] Chua L. O. ve Yang L. (1988). "Cellular Neural Networks : Theory", Circuits and Systems, IEEE Transactions, 35(10):1257-1272
- [28] Moradi M., Poormina M.A. ve Razzazi F.,(2009). "FPGA Implementation of Feature Extraction and MLP Neural Network Classifier for Farsi Handwritten Digit Recognition", Computer Modeling and Simulation, EMS '09. Third UKSim European Symposium
- [29] Saatçi E., (2003), Image Processing Using Cellular Neural Networks, Doktora Tezi, London South Bank Universty, Londra
- [30] Qian S. ve Chen D., (1999). "Joint Analysis", IEEE Signal Processing Magazine
- [31] Gabor D., (1946). "Theory of Communication", J. Inst. Elect. Eng. London, 93:429 - 457
- [32] Vaidyanathan P. P., (2008). "Eigenfunctions of the Fourier Transfor", IEEE Journal of Education
- [33] Daugman J. G., (1985). "Uncertainty Relation for Resolution in Space, Spatial Frequency and Orientation Optimized by Two-Dimensional Visual Cortical Filters", Journal of the Optical Society of America, 2:1160 - 1169
- [34] Cesur E., Yıldız N. ve Tavşanoğlu V., (2012). "On an Improved FPGA Implementation of CNN-Based Gabor-Type Filters", IEEE Transactions on Circuits and Systems 59(11):815-819
- [35] Shi B. E., (1998). "Gabor-Type Filtering in Space and Time with Cellular Neural Networks", Circuits and Systems I: Fundamental Theory and Applications, IEEE Transactions , 45(2):121-132
- [36] Li W., Mao K., Zhang H. ve Chai T. (2010). "Selection of Gabor Filters for Improved Texture Feature Extraction", Proceedings of 2010 IEEE 17th International Conference on Image Processing
- [37] Dunn D. ve Higgins W. E., (1995). "Optimal Gabor Filters for Texture Segmentation", IEEE Transactions on Image Processing, 4(7): 947 - 964
- [38] Zhou Z., Wu H. ve Lv Q, (2008)." A New Iris Recognition Method Based on Gabor Wavelet Neural Network", International Conference on Intelligent Information Hiding and Multimedia Signal Processing
- [39] Areekul V., Watchareeruetai U., Suppasriwasuseth K. ve Tantaratana S., (2005). "Separable Gabor Filter Realization for Fast Fingerprint Enhancement", Image Processing, 2005. ICIP 2005. IEEE International Conference
- [40] Nemati J. R. ve Javed M. Y., (2008)."Fingerprint Verification using Filter-bank of Gabor and Log Gabor Filters", Systems, Signals and Image Processing, 2008. IWSSIP 2008. 15th International Conference
- [41] Chung K. C., Kee S. C. ve Kim S. R., (1999)."Face Recognition using Principal Component Analysis of Gabor Filter Responses", Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems, 1999. Proceedings. International Workshop

- [42] Caner H., (2006), FPGA Donanımı Üzerinde Araç Plakası Tanıma Sistemi, Yüksek Lisans Tezi, Hacettepe Üniversitesi Fen Bilimleri Enstitüsü, Ankara
- [43] Afshang M., Helfroush M. S. ve Zahernia A., (2009). "Gabor Filter Parameters Optimization for Texture Classification Based on Genetic Algorithm", 2009 Second International Conference on Machine Vision
- [44] Horiuchi, T., Haruki, R., Yamada, H. ve Yamamoto, K., (1997). "Two-dimensional Extension of Nonlinear Normalization Method using Line Density for Character Recognition", Document Analysis and Recognition, 1997., Proceedings of the Fourth International Conference
- [45] Cheng-Lin K. M. ve Fujisawa H., (2005). "Gabor Feature Extraction for Character Recognition Comparison with Gradient Feature", Document Analysis and Recognition. Proceedings Eighth International Conference, vol. 1, pp. 121-125
- [46] Pan R., Ma X. ve Wang L., (2010). "An Efficient Method for Skew Correction of License Plate", 2010 Second International Workshop on Education Technology and Computer Science, 2:90-93
- [47] Calik N., Cesur E. ve Tavsanoğlu V., (2013). "Handwritten character recognition application by using Cellular Neural Network", Signal Processing and Communications Applications Conference (SIU)
- [48] Bilgisayar Kavramları, KNN (K nearest neighborhood, en yakın k komşu), <http://www.bilgisayarkavramlari.com/2008/11/17/knn-k-nearest-neighborhood-en-yakin-k-komsu/>, 2013
- [49] Gonzalez R. C., (2001). Digital Image Processing, second edition, Prentice Hall Co, New Jersey
- [50] Jahne B., (2005). Digital Image Processing, sixth edition, Springer, Heidelberg
- [51] Karhan M., Oktay M. O., Karhan Z. ve Demir H., (2011). "Morfolojik Görüntü İşleme Yöntemleri ile Kayıslarda Yaprak Delen (Çil) Hastalığı Sonucu Oluşan Lekelerin Tespiti", 6th International Advanced Technologies Symposium (IATS'11)

EL YAZISI HARF TABLOSU

Çizelge 5.1 7 farklı kullanıcı ile oluşturulmuş 10 farklı harf setleri

1	2	3	4	5	6	7	8	9	10
			A	A	A	A	A	A	A
A	A	A	B	B	B	B	B	B	B
B	B	B	C	C	C	C	C	C	C
C	C	C	D	D	D	D	D	D	D
D	D	D	E	E	E	E	E	E	E
E	E	E	F	F	F	F	F	F	F
F	F	F	G	G	G	G	G	G	G
G	G	G	H	H	H	H	H	H	H
H	H	H	I	I	I	I	I	I	I
I	I	I	J	J	J	J	J	J	J
J	J	J	K	K	K	K	K	K	K
K	K	K	L	L	L	L	L	L	L
L	L	L	M	M	M	M	M	M	M
M	M	M	N	N	N	N	N	N	N
N	N	N	O	O	O	O	O	O	O
O	O	O	P	P	P	P	P	P	P
P	P	P	Q	Q	Q	Q	Q	Q	Q
Q	Q	Q	R	R	R	R	R	R	R
R	R	R	S	S	S	S	S	S	S
S	S	S	T	T	T	T	T	T	T
T	T	T	U	U	U	U	U	U	U
U	U	U	V	V	V	V	V	V	V
V	V	V	W	W	W	W	W	W	W
W	W	W	X	X	X	X	X	X	X
X	X	X	Y	Y	Y	Y	Y	Y	Y
Y	Y	Y	Z	Z	Z	Z	Z	Z	Z
Z	Z	Z							

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı : Nurullah ÇALIK
Doğum Tarihi ve Yeri : 29.10.1987 / Tekirdağ
Yabancı Dili : İngilizce
E-posta : nurullah.calik@gmail.com

ÖĞRENİM DURUMU

Derece	Alan	Okul/Üniversite	Mezuniyet Yılı
Lisans	Elektronik ve Hab. Mühendisliği	Yıldız Teknik Üniversitesi	2010
Lise	Fen	Tekirdağ Anadolu Lisesi	2005

Bildiri

1. SIU 2013 "Hücrel Sinir Ağları Kullanılarak El Yazısı Tanıma Uygulaması"