

**T.C.
SÜLEYMAN DEMİREL ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**PARALEL PROGRAMLAMA İLE EL YAZISI RAKAMLARININ
TANINMASI**

Ali GÜNEŞ

Danışman: Doç. Dr. Tuncay YİĞİT

**YÜKSEK LİSANS TEZİ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİMDALI
ISPARTA – 2011**

TEZ ONAYI

Ali GÜNEŞ tarafından hazırlanan “**Paralel Programlama ile El Yazısı Rakamlarının Tanınması**” adlı tez çalışması aşağıdaki jüri tarafından oy birliği / oy çokluğu ile Süleyman Demirel Üniversitesi Mühendislik-Mimarlık Fakültesi Bilgisayar Mühendisliği Anabilim Dalı’nda **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Danışman: Doç. Dr. Tuncay YİĞİT (İmza)
Süleyman Demirel Üniversitesi Bilgisayar Mühendisliği Anabilim Dalı

Jüri Üyeleri :
Yrd. Doç. Dr. Bayram CETİŞLİ (İmza)
Süleyman Demirel Üniversitesi Bilgisayar Mühendisliği Anabilim Dalı

Yrd. Doç. Dr. Ecir Uğur KÜÇÜKSİLLE (İmza)
Süleyman Demirel Üniversitesi Bilgisayar Mühendisliği Anabilim Dalı

Prof. Dr. Mehmet Cengiz KAYACAN
Enstitü Müdürü

Not: Bu tezde kullanılan özgün ve başka kaynaktan yapılan bildirişlerin, çizelge, şekil ve fotoğrafların kaynak gösterilmeden kullanımı, 5846 sayılı Fikir ve Sanat Eserleri Kanunundaki hükümlere tabidir.

İÇİNDEKİLER

	Sayfa
İÇİNDEKİLER	i
ÖZET.....	iii
ABSTRACT	iv
TEŞEKKÜR.....	v
ŞEKİLLER DİZİNİ.....	vi
ÇİZELGELER DİZİNİ	viii
SİMGELER VE KISALTMALAR DİZİNİ	ix
1. GİRİŞ	1
1.1. El Yazısı Karakterlerinin Tanınması.....	2
2. KAYNAK ÖZETLERİ	6
2.1. Öznitelik Çıkarma ile İlgili Yapılmış Çalışmalar	6
2.2. Sınıflandırma ile İlgili Yapılmış Çalışmalar	9
3. MATERYAL VE METOD	12
3.1. Materyal	12
3.2. Yöntem.....	14
3.2.1.Paralel programlama	14
3.2.1.1. Paralel bilgisayarlar.....	16
3.2.1.2. Paralel bilgisayar bellek mimarileri	17
3.2.1.3. Paralel program tasarımı	20
3.2.1.4. Amdahl kanunu, paralel hızlanma ve performans.....	22
3.2.1.5. OpenMP	24
3.2.2. Dalgacık dönüşümü.....	25
3.2.2.2. Ayrık dalgacık dönüşümü	27
3.2.3. Kenar çıkarma ve eğim	33
3.2.4. Destek vektör makineleri	34
3.2.4.1. Lineer destek vektör makineleri.....	35
3.2.4.2. Lineer olmayan destek vektör makinesi sınıflandırıcıları.....	40
4. ARAŞTIRMA BULGULARI VE TARTIŞMA	43
4.1 Uygulama Platformu ve Ön hazırlıklar	43

4.2. Karakter İmgelerinin Ön İşlemden Geçirilmesi.....	44
4.3. Öznitelik Çıkarma Aşamasında Yapılan Çalışmalar	44
4.3.1. İki boyutlu ayırık dalgacık dönüşümü ile özniteliklerin elde edilmesi	44
4.3.2. Kenar çıkarma ve eğim öznitelikleri	45
4.4. Sınıflandırıcılar ve Tanıma Oranları	46
4.5 Seri ve Paralel Kodlara Ait Performans Analizleri	58
5. SONUÇ	61
6. KAYNAKLAR	63
ÖZGEÇMİŞ	81

ÖZET

Yüksek Lisans Tezi

PARALEL PROGRAMLAMA İLE EL YAZISI RAKAMLARININ TANINMASI

Ali GÜNEŞ

Süleyman Demirel Üniversitesi
Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

Danışman: Doç. Dr. Tuncay YİĞİT

Bu tez çalışmasının amacı, büyük ölçekli veri setleri üzerinde çalışabilen, hızlı ve tanıma oranı yüksek bir el yazısı karakter tanıma sisteminin geliştirilmesidir. Bu amaç doğrultusunda çalışma; ön işlemler, öznitelik çıkarma, sınıflandırma ve hız artırımı olmak üzere dört aşamada gerçekleştirilmiştir. Öznitelik çıkarma işlemi için, iki boyutlu dalgacık dönüşüm yöntemi ve sobel tabanlı gradient yöntemi kullanılmıştır.

Gerçekleştirilen çalışmanın sınıflandırma aşamasında üç adet destek vektör makinesi kullanılmıştır. İlk destek vektör makinesi iki boyutlu dalgacık yöntemi ile elde edilen öznitelikleri sınıflandırırken, ikincisi gradient yöntemi ile elde edilen öznitelikleri, üçüncüsü ise her iki yöntemle elde edilen öznitelikleri kullanarak sınıflandırma yapmıştır.

Çalışmanın hız artırımı bölümünde ise, paralel programlama mimarilerinden birisi olan paylaşımlı bellekli mimari kullanılmıştır. Bu doğrultuda, seri kodlar analiz edilmiş ve paralel çalışmaya uygun olan kod bölümleri paralelleştirilerek çok işlemcili bir bilgisayarda koşturulmuştur.

Deneysel çalışmalarda kullanılan el yazısı rakamlar bu alanda en çok bilinen MNIST veri tabanından elde edilmiştir. Elde edilen %96-98 arasındaki tanıma oranları ve 1.3 ile 2 kat arasındaki hız artışları genel anlamda yeterli sayılabilir.

Anahtar Kelimeler: Rakam tanıma, karakter tanıma, paralel programlama, dalgacık dönüşümü, destek vektör makineleri, paylaşımlı bellekli mimari

2011, 69 sayfa

ABSTRACT

M.Sc. Thesis

HANDWRITTEN DIGITS RECOGNITION WITH PARALLEL PROGRAMMING

Ali GÜNEŞ

**Suleyman Demirel University
Graduate School of Applied and Natural Sciences
Department of Computer Engineering**

Supervisor: Assoc. Professor. Dr. Tuncay YİĞİT

The aim of this thesis is to develop handwritten character recognition system which can run on the large scale data sets with fast and have a high recognition rate. For this purpose this thesis is performed throughout four stages which are pre-processing, feature selection, classification and speedup. Two-dimensional wavelet-based method and the Sobel gradient method are used for the feature extraction process.

Three support vector machines were used in the classification stage of thesis. While the first support vector machine was classified features which extract with two-dimensional wavelet method, the second one was classified features which extract with gradient method and the last one was classified to combine features which extract with both feature extraction methods.

In the speedup stage of the study, shared memory architecture was used which one of the parallel programming architectures. In this aspect, serial codes were analyzed and the code blocks which were suitable parallel works run with a multicore computer.

Handwritten numbers used in experimental studies, have been obtained from the MNIST database which is most well known in this field. The obtained %96-98 recognition rates and 1.3 with 2 time speed increase can be sufficient generally.

Keywords : Digit Recognition, character recognition, parallel programming, wavelet transformation, support vector machines, shared memory architecture

2011, 69 pages

TEŐEKKÖR

Tez alıőmam boyunca verdiđi destek ve ynlendirmeleri iin danıőman hocam Do. Dr. Tuncay YIĐIT' e teőekkrlerimi sunarım.

Ayrıca bu alıőmada ilgisi ve desteđiyle yardımlarını esirgemeyen Yrd. Do. Dr. Bayram CETIŐLI hocama teőekkr ederim.

2920-YL-11 No'lu Proje ile tezimi maddi olarak destekleyen Sleyman Demirel niversitesi Bilimsel Araőtırma Projeleri Ynetim Birimi Baőkanlıđına teőekkr ederim.

Tezimin her aőamasında beni yalnız bırakmayan aileme sonsuz sevgi ve saygılarımı sunarım.

Ali GNEŐ
ISPARTA, 2011

ŞEKİLLER DİZİNİ

Şekil 3.1. MNIST verilerinin genel görünümü	13
Şekil 3.2. MNIST veri tabanındaki tanınması zor olan bozuk rakamlar.....	14
Şekil 3.3. Seri hesaplama	15
Şekil 3.4. Paralel hesaplama	16
Şekil 3.5. Flynn taksonomisi.....	16
Şekil 3.6. Paylaşımlı bellekli mimari	18
Şekil 3.7. Dağıtık bellekli mimari.....	19
Şekil 3.8. Dağıtık paylaşımlı bellek.....	20
Şekil 3.9. Veri ayrışımı	21
Şekil 3.10. Fonksiyonel ayrışım.....	22
Şekil 3.11. f ve $(1-f)$ ' nin hızlanma ile değişimi (Lawrence Livermore National Laboratory, 2007)..	23
Şekil 3.12. Kodun paralellik oranı ile hızlanma arasındaki ilişki (Lawrence Livermore National Laboratory, 2007)	23
Şekil 3.13 İşlemci sayısı ile hızlanma arasındaki ilişki (Lawrence Livermore National Laboratory, 2007)	24
Şekil 3.14. OpenMP çatallaş-birleş metodu.....	25
Şekil 3.15. Fourier dönüşümü (Misiti et al, 1996).....	26
Şekil 3.16. Bir sinyalin alçak ve yüksek geçiren filtrelerden geçirilmesi (Misiti et al, 1996).....	27
Şekil 3.17. Bir sinyalin bir seviyeli çözünürlükte bileşenlerine ayrıştırılması	28
Şekil 3.18. Bir sinyalin çok seviyeli ayrıştırılması(Misiti et al, 1996)	29
Şekil 3.19. Bir sinyalin geri çatılması (Misiti et al, 1996).....	30
Şekil 3.20. Haar ölçekleme ve dalgacık fonksiyonu.....	32
Şekil 3.21. Haar alçak ve yüksek geçiren filtre katsayıları	32
Şekil 3.22. Daubchies 4 ölçekleme ve dalgacık fonksiyonu.....	33
Şekil 3.23. Daubchies 4 alçak ve yüksek geçiren filtre katsayıları.....	33
Şekil 3.24. Sobel kenar çıkarma operatörü yatay ve düşey matrisleri	34
Şekil 3.25. Optimum ayırıcı düzlem	36
Şekil 3.26. Lineer olmama durumunda optimal ayırıcı düzlem.....	38
Şekil 3.27. Doğrusal olmayan bir destek vektör makinesi.....	40
Şekil 3.28. Giriş uzayını özellik uzayına eşleme	40
Şekil 4.1. Sobel kenar çıkarma operatörü yatay ve düşey matrisleri	45
Şekil 4.2. Kenar yön açlarına göre yön kodlaması	46
Şekil 4.3. Öznitelik setleri ve sınıflayıcılar	48
Şekil 4.4. Çekirdek fonksiyonlarına göre doğru tanıma oranları	49
Şekil 4.5. Çekirdek fonksiyonlarına göre eğitim süreleri	49
Şekil 4.6. Çekirdek fonksiyonlarına göre test süreleri	49
Şekil 4.7. RBF çekirdek fonksiyonu hata dağılımı	50
Şekil 4.8. Lineer çekirdek fonksiyonu hata dağılımı	50

Şekil 4.9. Polinom çekirdek fonksiyonu hata dağılımı	50
Şekil 4.10. Çekirdek fonksiyonlarına göre başarı oranları.....	51
Şekil 4.11. Çekirdek fonksiyonlarına göre eğitim süreleri	52
Şekil 4.12. Çekirdek fonksiyonlarına göre test süreleri	52
Şekil 4.13. Lineer çekirdek hata dağılımı	53
Şekil 4.14. Polinom çekirdek hata dağılımı	53
Şekil 4.15. Radyal tabanlı fonksiyon çekirdek hata dağılımı	53
Şekil 4.16. Çekirdek fonksiyonlarına göre başarı oranları.....	55
Şekil 4.17. Çekirdek fonksiyonlarına göre eğitim süreleri	55
Şekil 4.18. Çekirdek fonksiyonlarına göre test süreleri	56
Şekil 4.19. Lineer çekirdek hata dağılımı	56
Şekil 4.20. Polinom çekirdek hata dağılımı	57
Şekil 4.21. Radyal tabanlı fonksiyon hata dağılımı	57
Şekil 4.22. İlk öznitelik seti için seri ve paralel kod analizleri	59
Şekil 4.23. İkinci öznitelik seti için seri ve paralel kod analizleri	59
Şekil 4.24. Üçüncü öznitelik seti için seri ve paralel kod analizleri	60

ÇİZELGELER DİZİNİ

Çizelge 2.1. Yapılan diğer çalışmalarda elde edilen sonuçlar	11
Çizelge 3.1. Eğitim ve test setlerindeki karakter dağılımları.....	13
Çizelge 4.1 Kenar yön açı aralıkları ve yön kodlaması.	46
Çizelge 4.2. Piksel yönlerini içeren öznitelik seti ile elde edilen sonuçlar.....	48
Çizelge 4.3. Coiflet2 dalgacık seti ile elde edilen sonuçlar..	51
Çizelge 4.4. Dalgacık filtrelerinin tanıma oranına etkileri.....	54
Çizelge 4.5. Dalgacık katsayıları ve piksellere ait yön kodları ile elde edilen sonuçlar	55
Çizelge 4.6. Elde edilen en iyi tanıma oranları.....	57
Çizelge 4.7. Seri kod bloklarının performans sonuçları.....	58

SİMGELER VE KISALTMALAR DİZİNİ

KISALTMALAR

CPU	Central Processing Unit
UMA	Uniform Memory Access
NUMA	Non-Uniform Memory Access
SVM	Support Vector Machine
KKT	Karush-Kuhn-Tucker
GPU	Graphics Processing Unit
RBF	Radial Basis Function
SISD	Single Instruction Single Data
SIMD	Single Instruction Multiple Data
MISD	Multiple Instruction Single Data
MIMD	Multiple Instruction Multiple Data
ANN	Artificial Neural Network

1. GİRİŞ

Örüntü tanıma, tanımlanmamış örüntülerin çeşitli işlemlerden geçirilmesi sonucu bilinen gruplara ayrılması yani sınıflandırılması veya tanımlanmasıdır. Örüntü tanıma çalışmaları geçmişten günümüze kadar bilimsel bir disiplin olarak medikal, askeri, sanayi ve robotik gibi birçok alanda uygulanmıştır. Bunlara ses, imge, optik karakter, yüz, parmak izi, belge ve doku tanıma, hedef takibi gibi çeşitli uygulamalar örnek gösterilebilir (Tou and Gonzalez, 1977; Schalkoff, 1992; Bishop, 1995; Gose et al., 1996; Theodoridis and Koutroumbas, 1999). Örüntü tanıma uygulamalarını insanların yaşantısında da görebiliriz: hava değişiminin algılanması, çiçek, bitki ve hayvan türlerinin tanımlanması, kitap okuma, yüz ve ses tanıma gibi bulanık sınırlara sahip birçok etkinlikte örüntü tanıma kullanılır. İnsan örüntü tanınması, geçmiş tecrübelerle dayalı öğrenme esaslıdır. Böylece, insanlar pratikte karşılaştığı örüntü tanıma olaylarını tecrübeleri ışığında değerlendirebilme yeteneğine sahiptirler. Belirli bir sesi tanımak için kullanılan kuralları tanımlamak mümkün değildir. İnsanlar bu işlemlerin birçoğunu oldukça iyi yapmalarına rağmen, bu işlemlerin daha ucuz, daha iyi, daha hızlı ve otomatik olarak makineler tarafından yapılmasını arzularlar. Örüntü tanıma, bu nitelikteki akıllı ve öğrenebilen makineleri gerçekleştirmek için, çok boyutlu bir mühendislik disiplindir (Hazar, 2007). Örüntü tanıma sistemleri temel olarak üç aşamadan oluşmaktadır:

- Veri Önleme
- Öznitelik Çıkarımı
- Sınıflandırma

Veri önleme aşamasında verileri kullanıma daha uygun hale getirebilmek için çeşitli yöntemlerle iyileştirmeler yapılabilir. Örneğin; bir resim veya sinyal içerisinde yer alan gereksiz bilgiler temizlenerek sistemin daha iyi sonuçlar üretmesi sağlanabilir. Öznitelik çıkarımı aşamasında ise orijinal verileri temsil eden veri grupları oluşturulur. Böylece daha az ancak temsil gücü daha yüksek verilerle daha iyi sonuçlar elde edilebilir. Sınıflandırma aşamasında ise, çıkarılan öznitelikler belirlenen kurallara göre sınıflandırılarak örüntü tanımlanır.

1.1. El Yazısı Karakterlerinin Tanınması

Örüntü tanıma disiplininin bir alt çalışma alanı da karakter tanıma uygulamalarıdır. Karakter tanıma uygulamaları son elli yıl içerisinde birçok bilimsel çalışmaya konu olmuş ve birçok alanda uygulamaları yapılmıştır. Bu uygulamalar ile yapılması gereken işler bir otomasyon sistemi tarafından yapılarak insan kaynaklı hataların ortadan kaldırılmasının yanında zamandan da kazanç sağlanmıştır. Karakter tanıma, sayısal görüntüler içerisindeki harf, rakam veya sembollerin dijital ortamda kullanılabilir hale getirilmesidir. Karakter tanıma sistemleri de kendi içerisinde el yazısı karakter tanıma ve optik karakter tanıma olarak ikiye ayrılmaktadır. Optik karakterlerin tanınması, el yazısına göre çok daha basittir. Çünkü optik karakterlerde tek tip yazım şekli vardır ve karakterler el yazısına göre daha düzgün bir yapıdadır.

El yazıları ise kişiden kişiye farklılık göstermekle beraber bazı durumlarda kişi kendi el yazısını bile okuyamamaktadır. Bunun yanında kullanılan kalem, kâğıt, tarayıcı gibi çevresel etmenlerde tanıma oranını etkilemektedir. Bu açıdan el yazısı tanıma sistemleri çok daha karmaşık bir yapıdadır. Hem optik hem de el yazısı karakter tanıma sistemleri günümüzde birçok alanda kullanılmaktadır. Masaüstü yayıncılık, mahkemeler, vergi ve tahsilat büroları, personel kayıtlarının yönetimi, bankacılıkta özel olarak doldurulan formların takibi ve daha birçok alanda kullanılmaktadırlar.

Bankalarda kredi kartı için doldurulan başvuru formlarının bir personel tarafından bilgisayar sistemine aktarılması saatleri bulacaktır. Oysa bir karakter tanıma yazılımı ile bu işlem oldukça kısa bir sürede tamamlanacaktır. Ayrıca günümüzde el yazısı ile işlem yapan elektronik cihazların sayılarında da oldukça büyük bir artış olmuştur ve bu tip cihazlar içinde karakter tanıma yazılımlarının başarısı ve hızı çok önemli bir çıkış nedeni olmuştur (Demirci, 2007).

El yazısı karakterlerini tanıma problemi için, birçok proje ve algoritma geliştirilmiştir (Bhattacharya et al., 2004). Karakter tanıma sistemleri de, tüm örüntü tanıma sistemlerinde olduğu gibi önışleme, öznitelik çıkarımı ve sınıflama aşamalarından meydana gelmektedir. Önışleme aşamasında karakter imgeleri üzerinde yer alan gereksiz bilgilerin temizlenmesi, ölçeklendirme, ikili biçime dönüştürme, bölümlenme gibi ön işlemler yapılır. Kullanılmaya daha uygun hale

getirilen karakterler ayırt edici özelliklerinin çıkarılması için öznitelik çıkarma aşamasına gönderilir.

Elde edilen öznitelikler sınıflandırma aşamasında orijinal karakterleri temsil eden veriler olarak kullanılır. Bu özniteliklerin kalitesi bir anlamda örüntülerin iyileştirilmesi ve gürültülerden arındırılmasına bağlıdır (Schalkoff, 1989; Lim, 1990; Awcock and Thomas, 1996; Pitas, 1999). İyileştirilmiş ve gürültüden arındırılmış veri setlerinden elde edilen özniteliklerin ayırt edici özellikleri güçlü olmakla beraber, bu durumdan sınıflayıcılarda pozitif yönde etkilenmektedir. Karakter tanıma uygulamalarında kullanılan öznitelikler, istatistiksel ve yapısal olmak üzere iki temel başlık altında toplanabilir. İstatistiksel özellik çıkarmada Geometrik, Legendre, Zernike, Fourier-Mellin, Fourier ve dalgacık gibi moment temelli yöntemler kullanılmaktadır (Khotanzad and Hong, 1990; Shenand Ip, 1999; Kan and Srinath, 2001; Kan and Srinath, 2002). Bu yöntemler ile karakterler üzerinde daha çok bölgesel analiz yapılmaktadır. Yapısal özellik çıkarmada ise zincir kodu, histogram, şablon eşleme, karakterlerin parça analizi, dış hat takibi, eğri uydurma, topolojik ve geometrik analiz gibi yöntemler kullanılabilir. Ancak yapısal özellik çıkarma yöntemleri kullanılarak elde edilen öznitelikler genelde değişkenlik göstermektedir. Karakterlerin büyük ya da küçük, yatık, ince, kalın olması, karakter parçalarının kopuk ya da eksik olması öznitelikleri değişken kılmaktadır (Cetişli, 2005). Zincir kodu yöntemini ele alırsak, birçok parçadan oluşan “4” rakamı ile halkalı bir yapısı olan “9” rakamından elde edilen zincir kod uzunlukları farklı çıkmaktadır. Bu şekilde farklı uzunluktaki zincir kodlarının sınıflama aşamasından önce aynı uzunluğa sabitlemesi gerekmektedir. Bu uzunluk sabitleme işlemi zincir kodlarını bozmadan gerçekleştirmek oldukça zor yapılmakta ve fazladan işlem gerektirmektedir. Yapısal sorunların yanı sıra, 2-boyutlu örüntülerin imge içindeki yerleşimleri, ölçekleri ve yönleri de sorun olmaktadır. Eğer karakterler imgenin merkezinden farklı bir yerde bulunuyorsa, belli bir açıyla dönmüşse veya kapladığı alan bakımından küçük ya da büyükse tanınması zorlaşmaktadır. Çünkü örüntünün beklenen yerde ve şekilde bulunamaması, çıkarılan bölgesel öznitelikleri oldukça etkilemektedir. Bu üç istenmeyen fiziksel durum için örüntülerin sabitlemesi gerekmektedir. Eğer örüntüler sabitlememişse üretilen öznitelikler de sabit değildir. Dönmüş, yer değiştirmiş ve küçültülmüş bir karakterin herhangi bir yöntemdeki

öznitelikleri, özgün karakter özniteliklerinden farklı çıkmaktadır. Bu nedenle mutlaka yer değiştirme, dönme ve ölçeklemeye göre öznitelikler değişmez olmalıdır (Cetişli, 2005). Bu amaçla yer değiştirme, dönme ve ölçekleme gibi olası durumlarda oldukça başarılı sonuçlar veren Fourier ve Dalgacık momentleri yaygın olarak kullanılmaktadır.

Fourier dönüşümü; herhangi bir sinyali, ortak bir fonksiyonun farklı frekanslarıyla ifade etmek amacıyla ortaya konmuştur. Ancak bu dönüşümde, frekans analizi yapılırken zaman bilgisi kaybedilmektedir. Bundan dolayı frekans-zaman düzleminde herhangi bir anda meydana gelen olaylar gözlemlenememektedir. Bu sorun Fourier dönüşümünün sabit olmayan sinyallerde kullanılmasını engellemektedir. Dalgacık dönüşüm yöntemiyle bu sorun giderilmiştir.

Bu çalışmada, el yazısı karakterleri olarak, MNIST (Modified National Institute of Standards and Technology) veri tabanı kullanılmıştır. Bu veri tabanında yer alan tüm resimler 28x28 piksel boyutlarında gri seviye görüntülerden oluşmaktadır.

Bu tez çalışmasında iki farklı öznitelik çıkarma yöntemi kullanılmıştır. İlk yöntemde, hem bölgesel hem de genel özniteliklerin elde edildiği ve karakterlerin bulunduğu yere, yönüne ve kapladığı alana göre gösterdiği değişimlere Fourier dönüşümünden daha duyarlı olan 2-boyutlu dalgacık dönüşümü kullanılmıştır. 2-boyutlu dalgacık katsayılarını elde ederken ana dalgacık olarak Daubechies, Coiflet, Biorthogonal ve Haar dalgacık filtreleri ayrı ayrı test edilmiştir. Pratikte dalgacık seçiminde kullanılan bir algoritma bulunmamaktadır. Dalgacık seçiminde istenilen durum, dalgacığın çalışılan sinyale zaman düzleminde olabildiğince benzemesidir.

İkinci öznitelik çıkarma yöntemi olarak; ikili biçime dönüştürülen her bir karakter için Sobel operatörü uygulanarak kenarlar tespit edilmiş ve karakteri oluşturan her bir pikselin mutlak eğim büyüklüğü ve yönü bulunmuştur.

Çalışmanın sınıflandırma aşamasında ise üç ayrı destek vektör makinesi sınıflayıcı olarak kullanılmıştır. Destek vektör makinelerinin ilki dalgacık özniteliklerini (256 adet), ikincisi piksel eğim özniteliklerini (784 adet) sınıflandırırken, üçüncüsü ise bu

iki 6znitelik y6ntemiyle elde edilen yeni 6znitelik grubunu (1040 adet) sınıflandırmaktadır.

Çalıřmada performans artırmak amacıyla paralel programlama mimarilerinden birisi olan paylařımlı bellekli mimari kullanılmıřtır. Bu amaçla çalıřmanın seri kodunda yer alan bařta d6ng6ler olmak 6zere sistemin en fazla zaman kaybettiđi b6lgeler ve paralel çalıřmaya uygun olan kod b6l6mleri çok kanallı olarak çalıřtırılmıřtır.

2. KAYNAK ÖZETLERİ

2.1. Öznitelik Çıkarma ile İlgili Yapılmış Çalışmalar

El yazısı rakam tanıma çalışmaları uzun yıllar önce başlamış olan ve günümüzde de halen üzerinde çalışmalar yapılan, popülerliğini yitirmeyen bir örüntü tanıma disiplini.

Brown vd.(1988), karakterlerin topolojik ve geometrik özelliklerini çıkararak bir karakter tanıma sistemi geliştirmişler ve bu çalışmada %97 tanıma oranı elde etmişlerdir.

S.W Lee (1996) yapmış olduğu çalışmada, Kirsch maskesini kullanarak her bir karakter için analitik düzlem üzerinde yer alan 4 farklı yönde yerel ve global öznitelikler elde etmiştir. Her bir karakter için toplamda 80 adet öznitelik elde edilmiş ve bu öznitelikler 3 katmanlı yapay sinir ağları ile sınıflandırılmıştır. Yapılan çalışma CONCORDIA, ETL-1 ve ETRI veri tabanları üzerinde test edilmiş ve sırasıyla %97.10, %99.12 ve %99.40 tanıma oranları elde edilmiştir.

Trier vd. (1996), istatistiksel ve yapısal öznitelik çıkarma yöntemlerinden oluşan 11 farklı öznitelik çıkarma yöntemini farklı formattaki karakter imgeleri üzerinde test ederek oldukça kapsamlı bir çalışma yapmışlardır.

Oh ve Suen (1998), yapmış oldukları çalışmada beyaz ve siyah pikseller arasındaki uzaklıklardan yola çıkarak iki yeni öznitelik çıkarma yöntemi geliştirmişlerdir. Geliştirilen yöntemler CENPARMI, NIST ve PE92 veri tabanları üzerinde test edilmiş ve %80.08 ile % 98.90 arasında tanıma oranları elde edilmiştir.

Gao ve Ding (2000) yapmış oldukları çalışmada iki yeni öznitelik çıkarma yöntemi geliştirmişlerdir. Önerilen yöntemler farklı örüntüler üzerinde de kararlı sonuçlar vermiştir. Yapılan çalışma el yazısı rakamlardan oluşan NIST veri tabanı ve Çince karakterler üzerinde test edilmiştir.

Shi vd.(2002), tarafından yapılan çalışmada karakterlerin eğimlerinden bir öznitelik vektörü elde edilmiş ve bu öznitelikler içerisinde öznitelik uzayını temsil eden en

iyi temel bileşenler seçilerek sınıflandırılmıştır. Bu yöntem IPTP CDROM1, NIST SD3 ve SD7 veri tabanları kullanılarak test edilmiş ve % 98.25 ile % 99.49 seviyelerinde tanıma oranları elde edilmiştir.

Teow ve Loe (2002) MNIST veri tabanında yer alan karakterlerden biyolojik vizyon model adını verdikleri yöntemle çıkarttıkları öznitelikler ile % 0.59 gibi oldukça küçük hata oranına sahip bir çalışma yapmışlardır.

Bir başka çalışmada Liu vd.(2002), CENPARMI, CEDAR ve MNIST veri tabanlarını kullanarak bir çalışma yapmışlardır. Bu çalışmada 8 farklı öznitelik çıkarma yöntemiyle farklı uzunluklara sahip 8 adet öznitelik vektörü elde edilmiş olup, elde edilen öznitelikler birleştirilerek 7 farklı sınıflayıcı ile sınıflandırılmıştır. Bu kapsamlı çalışmanın öznitelik çıkarma aşamasında 4 ve 8 yönlü Zincir Kodları, farklı doğrultularda uygulanan Sobel ve Kirsh kenar çıkarma yöntemleri ve karakterlere ait profil çıkarma yöntemleri kullanılmıştır. Bu öznitelik setlerinin en küçüğü 100, en büyüğü ise 233 adet öznitelikten oluşmaktadır. Sınıflandırma aşamasında ise çok katmanlı yapay sinir ağları, polinomsal sınıflandırıcı ve farklı çekirdek fonksiyonlarına sahip destek vektör makineleri kullanılmıştır. Yapılan testlerde MNIST veri tabanında yer alan karakterlerden elde edilen öznitelikler temel bileşenler analizi ile 80 elemanlı bir öznitelik vektörüne dönüştürülmüş ve en yüksek tanıma oranı % 99.58 olarak elde edilmiştir.

Oliveira vd. (2002), yapmış oldukları çalışmada karakter imgelerinden 8 doğrultulu zincir kodlama ile karakter öznitelikleri çıkarılmıştır. Her bir karakter 132 öznitelikle temsil edilmiş ve sistem yapay sinir ağları ile eğitilmiştir. Yapılan testler %99.13 gibi yüksek bir tanıma oranı elde edilmiştir.

Mowlai vd. (2002), bu çalışmada Farsça harf ve rakamlardan oluşan bir adres veri tabanında yer alan karakterleri ön-işlemlerden geçirerek 64 x 64 piksel boyutlarına ölçeklemiştirlerdir. Ölçeklenen bu karakterlere 3 seviye Haar dalgacık dönüşüm yöntemi uygulanarak 8 x 8 piksel boyutunda; yani 64 elemanlı bir öznitelik vektörü elde edilmiştir. Yapay sinir ağları ile eğitilen sistemde harfler için tanıma oranı, eğitim setinde % 98.97, test setinde % 92.33 ve rakamlar için tanıma oranı, eğitim setinde %99.29, test setinde ise %91.81 çıkmıştır.

G.Y. Chen vd. (2003), ortonormal çoklu dalgacık dönüşüm yöntemi ve yapay sinir ağlarını kullanarak bir el yazısı rakam tanıma sistemi geliştirmişlerdir. Bu çalışmada ilk olarak karakterlerin kenar özelliklerini içeren bir özellik vektörü elde edilmiştir. Daha sonra bu özellik vektörüne ortonormal çoklu dalgacık dönüşüm yöntemi 3 seviye uygulanarak, dalgacık katsayıları elde edilmiştir. Yapay sinir ağları kullanılarak yapılan sınıflandırma işlemi sonucu % 92.20 tanıma oranı elde edilmiştir.

L.Yang vd. (2005), el yazısı karakter tanıma problemlerinde 4 ve 9 gibi benzerlikleri sebebiyle ayırt etmekte zorlanılan karakterler üzerine bir çalışma yapmışlardır. Bu çalışmada 32x32 piksel boyutlarındaki karakterlerin kenar özellikleri çıkarılmış ve bu kenar özelliklerinin eğrilikleri hesaplanarak 128 adet bir öznitelik vektörü elde edilmiştir. Daha sonra bu öznitelik vektörünün boyutu Daubechies 4 filtresine sahip 2 boyutlu dalgacık dönüşüm yöntemiyle düşürülerek yeni bir öznitelik vektörü elde edilmiştir. Yapılan testlerde MNIST veri tabanında yer alan 4 ve 9 sayılarından oluşan eğitim ve test setleri kullanılmıştır. Sınıflandırma aşamasında yapay sinir ağları kullanıldığında %97.93, destek vektör makineleri kullanıldığında ise %99.06 tanıma oranı elde edilmiştir.

Son yıllarda yapılan öznitelik çıkarma çalışmalarında oldukça sık olarak kullanılan yöntemlerden biriside dalgacık dönüşümleri olmuştur. P. Zhang vd.(2006), yapmış oldukları çalışmada 2 farklı dalgacık dönüşüm yöntemi kullanmışlardır. Bunlardan ilkinde Kirsch kenar çıkarma yöntemi ile elde edilen öznitelikler üzerinde 2 boyutlu dalgacık dönüşüm yöntemi uygulanarak 112 adet yeni öznitelik elde edilmiştir. İkincisinde ise 2 boyutlu karmaşık dalgacık dönüşüm yöntemi kullanılarak 160 adet yeni öznitelik elde edilmiştir. Son olarak her bir karakterden 20 adet geometrik özellikler çıkarılarak birinci ve ikinci öznitelik setlerine eklenmiştir. Sınıflandırma aşamasında ise geri beslemeli yapay sinir ağları kullanılmıştır. İlk öznitelik seti ile %98.55 ve ikinci öznitelik seti ile %98.30 tanıma oranı elde edilmiştir.

E. Yen ve I-W.M. Chu (2007), 0-9 arası el yazısı rakamlardan oluşan bir veri tabanındaki karakterlerden Daubechies filtresini kullanarak 3 seviye dalgacık dönüşümü ile bir öznitelik vektörü elde etmişlerdir. Bu öznitelik vektörü Bayes sınıflayıcı ile kullanılmış ve rakam bazında en yüksek tanıma oranı % 96.23 ile 0 rakamında, en düşük tanıma oranı % 90.24 ile 5 rakamında çıkmıştır.

M.E. Gumah vd. (2008), yapmış oldukları çalışmada, hızlı dalgacık dönüşüm yöntemi ile Arap alfabesindeki harflerden oluşan bir veri tabanındaki el yazısı karakterlerden hızlı dalgacık dönüşüm yöntemiyle dalgacık katsayıları elde etmişlerdir. Çalışmada öznitelik vektörünü oluşturan bu dalgacık katsayıları, herhangi bir sınıflayıcıyla sınıflandırılmadan direkt olarak karakterlerin tanınmasında kullanılmıştır. Yapılan testlerde %94.18 tanıma oranı elde edilmiştir.

G.Raju (2008) yapmış olduğu çalışmada, Malayalam karakterlerinden oluşan veri tabanını kullanmıştır. Her bir Malayalam karakteri 32x64 piksel boyutlarına ölçeklenmiş ve her resmin düşey ve yatay projeksiyon profilleri elde edilmiştir. Elde edilen bu yatay ve düşey öznitelik vektörlerinden, yatay profilleri içeren vektör 2 seviye, düşey profilleri içeren vektör 3 seviye, 1 boyutlu dalgacık dönüşüm işlemine tabi tutulmuştur. Çeşitli dalgacık filtreleri kullanılarak elde edilen dalgacık katsayıları yapay sinir ağları ile sınıflandırılmış ve ortalama %76.8 tanıma oranı elde edilmiştir.

2.2. Sınıflandırma ile İlgili Yapılmış Çalışmalar

Karakter tanıma sistemlerinde öznitelik çıkarma yönteminin belirlenmesinden sonra en önemli aşama sınıflandırmadır. Elde edilen özniteliklere uygun sınıflandırma algoritmasının seçilmesi tanıma oranını birebir etkilemektedir. İstatistiksel sınıflandırma ve makine öğrenmesi algoritmaları yaygın olarak kullanılan sınıflandırma yöntemleridir.

Y. Le Cun vd. (1998), MNIST veri seti ile birçok istatistiksel ve makine öğrenmesi sınıflandırma algoritmasını kullanarak geniş çaplı bir çalışma yapmışlardır. Bu çalışmada kullanılan sınıflandırma algoritmalarından k-en yakın komşu algoritmasıyla en düşük %2.4 hata oranı, temel bileşenler analizi ve polinomsal sınıflandırma

yöntemiyle %3.3 hata oranı, destek vektör makineleri ile yapılan sınıflandırmalarda en düşük %0.8 elde edilmiştir.

D. Decoste ve B. Scholkophf (2002) ,MNIST veri tabanını kullanarak yapmış oldukları el yazısı karakter tanıma çalışmasında, destek vektör makinelerini kullanmış ve % 0.56 gibi düşük bir hata oranı elde etmişlerdir.

C-L. Liu vd. (2003), yapmış oldukları bu çalışmada CENPARMI, CEDAR ve MNIST veri setlerini kullanarak 10 farklı öznitelik vektörü elde etmiş ve sınıflandırma aşamasında 8 adet sınıflayıcıdan oluşan bir çoklu sınıflandırma yöntemi kullanmışlardır. Bu sınıflayıcılardan k-en yakın komşu algoritmasında k değerlerini 1,3,5,7,9 olarak seçmişler ve en yüksek tanıma oranlarını k=3 ve k=5 olduğu durumlarda elde etmişlerdir. CENPARMI veri setinde tüm öznitelik vektörleri k-en yakın komşu algoritmasıyla sınıflandırıldığında ortalama %2.63 hata oranı, CEDAR veri setinde ortalama %1.31 hata oranı, MNIST veri setinde ise ortalama % 1.15 hata oranı elde edilmiştir.

B. Zhang vd. (2004), NIST ve MNIST veri setleri üzerinde k-en yakın komşu istatistiksel sınıflandırma algoritmasını kullanarak bir çalışma yapmışlardır. Bu çalışma da her bir karakter için 512 adet öznitelik elde edilmiştir. Sınıflandırma sonucunda NIST veri setinde % 99.25 tanıma oranı elde edilirken, MNIST veri setinde ise % 98.24 tanıma oranını elde etmişlerdir.

Örüntü tanıma problemlerinde destek vektör makineleri çok yaygın olarak kullanılmaktadır. J.-X. Dong vd. (2005), tarafından gerçekleştirilen çalışmada, büyük ölçekli veri tabanları üzerinde sistemin eğitim süresinin çok daha kısa sürdüğü bir destek vektör makinesi algoritması geliştirilmiştir. Geliştirilen yöntem birçok çalışmayla karşılaştırılmıştır. Yaptıkları ilk testte 5000 adet karakter kullanılmış ve sistem her bir karakterden elde edilen 784 adet öznitelik ile test edilmiştir. Bu testte 5000 karakter eğitim amaçlı kullanılmış ve toplam süre 2.37 saat olarak ölçülürken, Keerthi vd., yaptığı benzer bir çalışmada ise toplam süre 21.44 saat olarak ölçülmüştür. Her iki çalışmada da destek vektör makinesi polinom çekirdek kullanılarak eğitilmiştir. Aynı çalışmada yapılan bir başka testte, 28 x 28 piksel boyutlarında olan karakterlerden temel bileşenler analizi ile öncelikle 120 özellik

çıkarılmış ve geliştirilen hızlı destek vektör sınıflayıcı ile sınıflandırılmıştır. Toplamda 5000 karakterin eğitim amaçlı olarak kullanıldığı bu test sonucunda ise sistemin toplam süresi 0.77 saat olarak ölçülmüştür. J.-X. Dong vd., yapmış oldukları bu çalışmada benzer çalışmalardan ortalama olarak 9 kat daha hızlı bir sınıflayıcı geliştirmişlerdir.

A.M. Sarhan ve O. I. Al Helalat (2007) Arap alfabesi için karakter tanıma sistemi geliştirmişlerdir. Bu çalışmada her bir karakter için istatistiksel yöntemleri kullanarak 36 adet öznitelik elde edilmiştir. Çalışmanın sınıflandırma aşamasında ise iki katmanlı bir yapay sinir ağı modeli çıkarılmıştır. Bu modelde, ilk katman 10 nörondan ikinci katman ise alfabedeki toplam harf sayısı olan 28 nörondan oluşmuştur. Modelde transfer fonksiyonu olarak sigmoid transfer fonksiyonu kullanılmıştır.

Bu bölümde anlatılan öznitelik çıkarma ve sınıflandırma çalışmalarına ek olarak Çizelge 2.1 'de CENPARMI, CEDAR ve MNIST veri setlerinin kullanıldığı diğer çalışmalar sunulmuştur.

Çizelge 2.1. Yapılan diğer çalışmalarda elde edilen sonuçlar

Veri Seti	Tanım Oranı (%)	Hata Oranı (%)	Tanınmayan (%)	Referanslar
CENPARMI	93.05	0	6.95	C.Y. Suen vd. (1992)
CENPARMI	98.50	1.50	0	J.Franke vd. (1993)
CENPARMI	97.80	2.20	0	S.-W. Lee (1995)
CENPARMI	98.30	1.70	0	P.D.Gader vd. (1996)
CENPARMI	98.00	2.00	0	C.-L. Liu vd. (1997)
CENPARMI	97.90	2.10	0	Y.-S. Hwang vd. (1997)
CENPARMI	97.60	2.40	0	J. Franke (1997)
CENPARMI	98.85	1.15	0	C.Y. Suen vd. (1999)
CENPARMI	97.85	2.15	0	I.-S. Oh vd. (1999)
CENPARMI	98.45	1.55	0	C.-L. Liu vd. (1999)
CEDAR	98.87	1.13	0	D.-S. Lee vd. (1993)
CEDAR	99.09	0.91	0	T. Ha vd. (1997)
CEDAR	99.77	0.23	0	C.Y. Suen vd. (1999)
CEDAR	99.54	0.46	0	A. Filatov vd. (1999)
CEDAR	98.40	1.60	0	J.-H. Cai vd. (1999)
CEDAR	98.73	1.27	0	I.-S. Oh vd. (1999)
MNIST	98.90	1.10	0	Y. LeCun vd. (1995)
MNIST	99.05	0.95	0	Y. LeCun vd. (1995)
MNIST	99.30	0.70	0	Y. LeCun vd. (1995)
MNIST	98.60	1.40	0	C.J.C. Burges vd. (1997)
MNIST	99.00	1.00	0	C.J.C. Burges vd. (1997)

3. MATERYAL VE METOD

3.1. Materyal

Bu çalışmada, sistemin eğitilmesi ve test işlemleri için uluslararası alanda çok yaygın olarak kullanılan ve el yazısı rakamlardan oluşan Modified National Institute of Standards and Technology (MNIST) veri tabanı kullanılmıştır. MNIST veri tabanı NIST veri tabanından oluşturulmuştur.

NIST veri tabanı SD-3 eğitim kümesi ve SD-1 test kümesinden oluşmaktadır. SD3 veri kümesi nüfus memurlarının yazmış oldukları karakterlerden oluşurken, SD1 ise lise öğrencilerinin yazmış oldukları karakterlerden oluşmaktadır. Bundan dolayı SD3 veri kümesi SD1 kümesine göre daha okunaklı ve temizdir. NIST veri tabanının eğitim kümesinde test kümesine göre daha temiz verilerin bulunması ve veri tabanının çok büyük olmasından dolayı daha kullanışlı olan MNIST veri tabanı oluşturulmuştur.

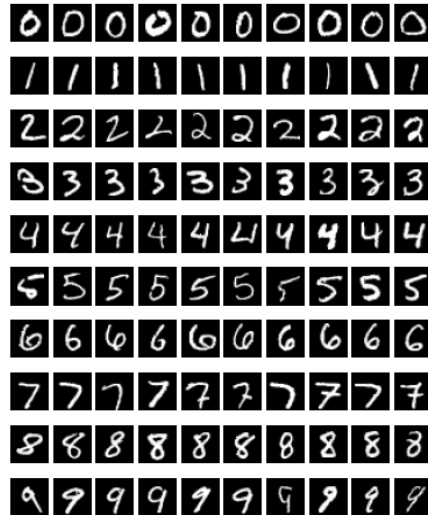
MNIST veri tabanının eğitim seti, NIST veri tabanının eğitim setinden 30000 ve test setinden 30000 yani toplamda 60000 örnek alınarak, test seti ise NIST veri tabanının eğitim setinden 5000 ve test setinden 5000 olmak üzere, toplamda 10000 örnek alınarak oluşturulmuştur.

MNIST veri tabanının eğitim setinde 60.000 ve test setinde 10.000 adet el yazısı ile yazılmış rakamlar mevcuttur (Lecun et al, 1998). Eğitim seti ve test seti içerisindeki tüm rakamlar 20x20 piksel ölçülerinde olup, 28x28 piksel ölçüsündeki bir çerçeve üzerinde yer almaktadır. Tablo 1 'de veri tabanı içerisindeki eğitim ve test setlerinde yer alan her sınıfa ait rakam sayıları gösterilmiştir.

Çizelge 3.1. Eğitim ve test setlerindeki karakter dağılımları

Numara	Eğitim	Test
0	5.923	980
1	6.742	1.135
2	5.958	1.032
3	6.131	1.010
4	5.842	982
5	5.421	892
6	5.918	958
7	6.265	1.028
8	5.851	974
9	5.949	1.009
Toplam	60.000	10.000

Çizelge 3.1.'de görüldüğü gibi eğitim ve test setlerindeki karakter sayıları kendi içlerinde birbirine çok yakın değerlerden oluşmaktadır. Sistemin eğitim ve test aşamalarında her bir rakam için eşit miktarda örnek kullanılması nedeniyle sistemin sınıf bazında tanıma oranının ve performans değerlerinin orantılı olacağı görülebilmektedir. Şekil 3.1.'de MNIST veri tabanı içinde bulunan rakamların genel görünümü yer almaktadır.



Şekil 3.1. MNIST verilerinin genel görünümü

MNIST veri tabanı içerisinde bulunan rakamların tamamı Şekil 3.1.'de görüldüğü gibi doğrudan kullanıma uygun değildir. Bazı rakamların etrafında gürültü adı verilen istenmeyen tarayıcı ya da mürekkep lekeleri bulunmakla birlikte, tanınması güç olan rakamlar da bulunabilmektedir. Şekil 3.2.'de veri tabanı içerisinde yer alan tanınması zor olan bozuk rakam örnekleri gösterilmiştir.



Şekil 3.2. MNIST veri tabanındaki tanınması zor olan bozuk rakamlar

3.2. Yöntem

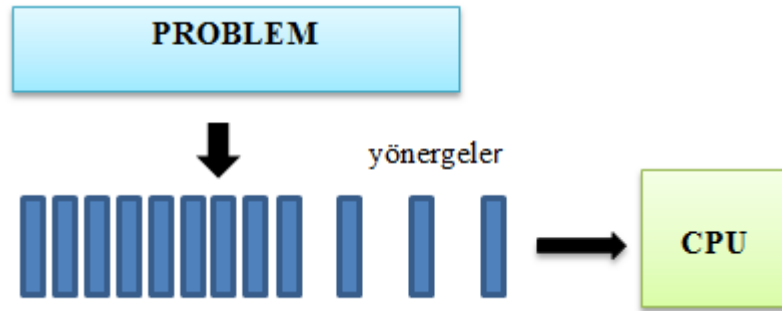
3.2.1.Paralel programlama

Bilişim sektörünün geldiği nokta ve buna bağlı olarak ihtiyaçların her geçen gün hızla artması birçok problemi de beraberinde getirmektedir. Bilgi teknolojileri sektörü de bu konuda pastadan en fazla dilimi alarak ilk sıraya oturmaktadır. Günümüzde gerek kurumsal, gerekse akademik çalışmalarda kullanılan veri boyutlarının artması son kullanıcılardaki performans olgusunu ilk sıraya yerleştirmiştir. Artık geliştirilen bir sistemin kararlı ve hatasız çalışmasının yanında en kısa sürede en iyi sonucu elde etmek de isteklerin başında gelmektedir. Hız konusu, üretim sektörü, mühendislik hesapları, askeri simülasyon projeleri ve hava tahmin hesaplamaları için hayati öneme sahiptir. Paralel hesaplamaların kullanıldığı birçok alan vardır. Bunlara örnek olarak;

- Hava tahmini
- Fizik-Nükleer hesaplamalar, Parçacıklar, Yüksek Basınç
- Uzay Kara-deliklerin modellenmesi
- Askeri Simülasyonlar
- Kimya-Moleküller
- Elektrik-Elektronik-Devre Tasarımı, Mikro-elektronik
- Veri tabanları - veri madenciliği

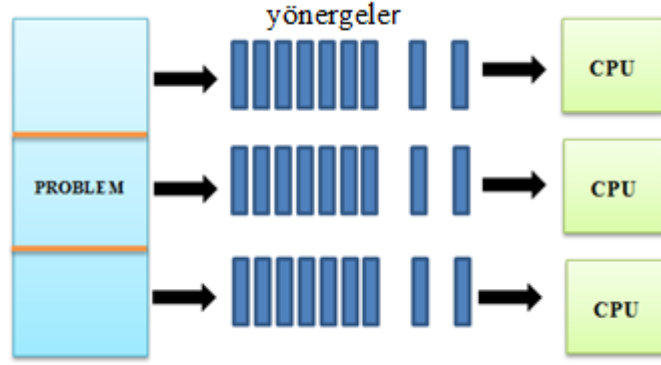
alanları verilebilir.

Gündelik hayatta kullanılan yazılımların çok büyük bir çoğunluğu seri olarak çalışmaktadır. Bu yazılımlar, bir bilgisayarda birden çok işlemci olsa bile tek işlemciyi kullanırlar. Seri sistemlerde her işlem bir diğerinin ardından gerçekleştirilir. Yani bir işlem süresinde sadece bir işlem gerçekleştirilmektedir. Şekil 3.3.'de yer alan görsel bu durumu özetlemektedir.



Şekil 3.3. Seri hesaplama

Paralel hesaplamada ise mevcut hesaplama kaynaklarının yani bir bilgisayarda yer alan tüm işlemcilerin aynı anda kullanılabilmesi söz konusudur. Paralel sistemlerde bir problem için geliştirilen yönergeler eş zamanlı olarak farklı işlemciler üzerinde çalıştırılmaktadır. Şekil 3.4.'de yer alan görsel bu durumu özetlemektedir.

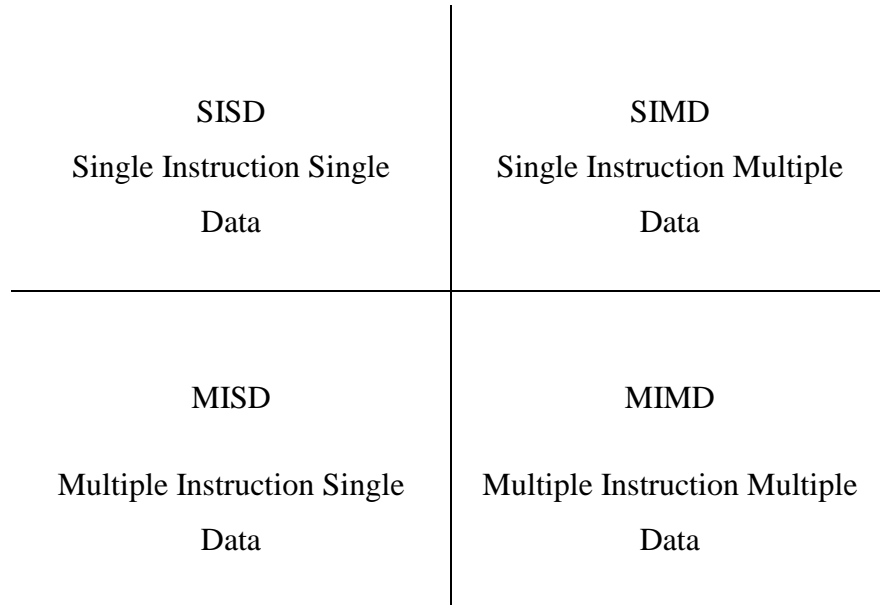


Şekil 3.4. Paralel hesaplama

Paralel hesaplamadaki en önemli kavramlar iplikler ve süreçlerdir. Süreç, çalışan bir sistem üzerinde koştan programın kendisidir. Her süreç belirli bir adres uzayına sahiptir. Farklı süreçler farklı adres blokları üzerinde yürütülmekte ve sistem kaynaklarını kullanabilmektedirler. İplikler ise süreçler içerisinde yer alırlar ve yer aldıkları sürecin adres havuzu içerisinde hareket ederler. İplikler adres uzayı, bellek ve açık dosyalar gibi sistem üzerindeki kaynakları kullanabilirler.

3.2.1.1. Paralel bilgisayarlar

Paralel bilgisayarların sınıflandırılmasında en çok bilinen sınıflandırma Flynn taksonomisidir. Flynn taksonomisi Şekil 3.5.' de görselleştirilmiştir.



Şekil 3.5. Flynn taksonomisi

SISD (Tek komut, tek veri)

Geleneksel tek işlemcili Von Neumann mimarisine sahip paralel bilgisayarlar bu gruba girmektedir. Bu mimariye göre tek bir kontrol ünitesine bağlı tüm işlemciler (çekirdek) aynı komutu paylaşımlı bellek üzerinde yer alan aynı veri setleri için seri olarak işletirler. Günümüzde kullanılan kişisel bilgisayarlar ve iş istasyonları bu gruba girmektedir.

SIMD(Tek komut, çoklu veri)

Bu mimari merkezi bir kontrol ünitesi ve işlemci birimlerinden meydana gelmektedir. Bir saat çevriminde aynı komutlar tüm işlemciler tarafından farklı veri setleri üzerinde işletilmektedir. Bu mimariye sahip bilgisayarlar ile veri seviyesinde paralelleşme yapılmaktadır.

MISD(Çoklu komut, tek veri)

MISD paralel bilgisayar mimarisinde, farklı komut setleri farklı işlemciler ile aynı veri seti için işletilmektedir. Bu mimari, karşılaşılan birçok problem için uygun değildir ve bu mimariye sahip çok az makina üretilmektedir.

MIMD(Çoklu komut, çoklu veri)

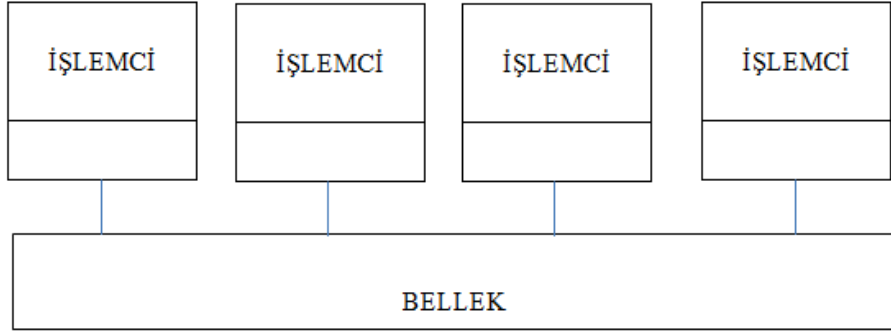
Bu mimaride ise, herhangi bir zaman diliminde farklı işlemciler farklı komutları bir veri setinin farklı parçaları üzerinde işletirler. Paralel bilgisayarlar olarak adlandırılan günümüz süper bilgisayarları bu gruba girmektedir. MIMD mimarisine sahip bilgisayarlar kendi içerisinde bellek bölgesine erişim yöntemlerine bağlı olarak paylaşımlı-bellekli MIMD veya dağıtık-bellekli MIMD olarak sınıflandırılırlar.

3.2.1.2. Paralel bilgisayar bellek mimarileri

Paralel bilgisayar yapıları; paylaşımlı bellek, dağıtık bellek ve dağıtık paylaşımlı bellek olmak üzere 3 gruba ayrılır. Problemin büyüklüğü ve hâlihazırdaki donanıma göre uygun mimariler tercih edilmektedir.

Paylaşımli bellek

Bu mimariye sahip bilgisayarlar da tüm işlemciler ortak kullanılan bellek alanına global adres uzayı üzerinden erişim sağlarlar. Tüm işlemciler birbirinden bağımsız olarak işlem yapabilirler ancak aynı bellek alanını paylaşırlar. Hafıza üzerindeki bir adres bölgesinde yapılan değişiklikler diğer işlemcilerin yürüttüğü işlemleri de etkilemektedir. Ortak bir bellek alanı kullanıldığı için işlemciler arası haberleşme hızlıdır. Bu modelde aktif olan bir süreç içerisindeki ipliklerin sayısı artırılarak iş yükü dağıtılır. Böylece aynı adres uzayında işlem yapan iplik sayısı ile orantılı olarak programın hızı artacaktır. Paylaşımli bellek mimarisi Şekil 3.6.'da görselleştirilmiştir.



Şekil 3.6. Paylaşımli bellekli mimari

Bu mimariye sahip bilgisayarlar UMA ve NUMA olmak üzere iki gruba ayrılmaktadır.

UMA(Tek düze bellek erişimi)

Bu tür makinelerde yer alan işlemciler eşit miktarda bellek bölgelerini kullanırlar ve belleğe erişim süreleri eşittir. Genelde bu makineler CC-NUMA(Uyumlu Ön Bellek-Tek Düze Erişim) olarak adlandırılırlar. Uyumlu ön bellek yapısı sayesinde, bir işlemci hafızanın belirli bir bölgesindeki veri üzerinde değişiklik yaptığında, diğer işlemcilerde bu değişiklikten haberdar edilmektedir. Bu tür makinelere örnek olarak Simetrik Çoklu İşlemcilere(SMP) sahip olan bilgisayarlar verilebilir.

NUMA(Tek düze olmayan bellek erişimi)

İki ya da daha fazla SMP makinenin fiziksel olarak birbirlerine bağlanmasıyla oluşur. Bir SMP makine direkt olarak bir başka SMP makinenin hafızasına erişim sağlayabilir. Tüm işlemcilerin belleğe erişim süreleri ve bellek kullanım miktarları farklıdır (Lawrence Livermore National Laboratory, 2007).

Dağıtık bellek

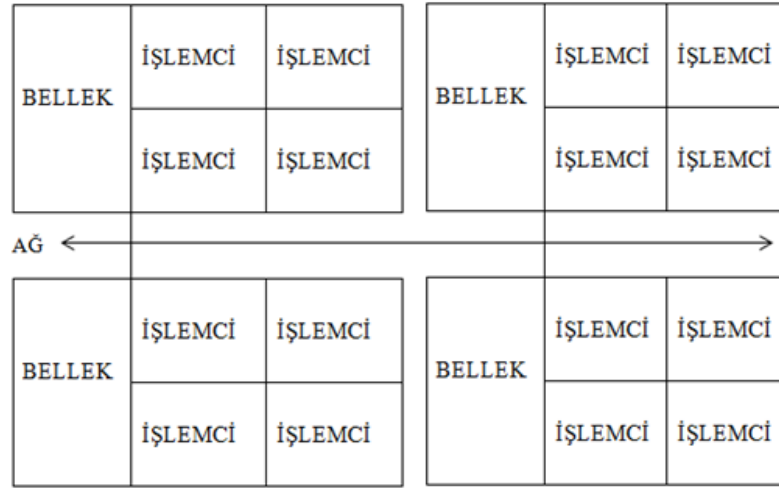
Bu modelde bilgisayarlar bir ağ hattı üzerinden mesajlaşarak haberleşirler. Her işlemci yer aldığı makinenin yerel belleğini kullanır. İşlemciler farklı bellek birimlerini kullandıkları için bellek adres uzayları tamamen farklıdır. İşlemciler birbirlerinden bağımsız olarak çalışırlar ve bir işlemcinin kendi hafıza bölgesinde yaptığı değişiklikler diğer işlemcilere yansımaz. İşlemciler veri ve iş paylaşımı için belirli kurallar çerçevesinde ağ hattı üzerinden mesajlaşırlar. Bu haberleşmeler aslında süreçler arasında gerçekleştirilir. Süreçler sistem kaynaklarını kullanabildikleri için, bir makinede aktif olan bir süreç, ihtiyaç dâhilinde diğer bir makinedeki sürece ağ hattı üzerinden mesaj göndererek iş yükünü dağıtabilir. Makineler arası iletişimin detayları programcı tarafından belirlenir. Bu sistemde çoklu süreç yaklaşımı söz konusudur. Dağıtık bellekli mimari Şekil 3.7.'de görselleştirilmiştir.



Şekil 3.7. Dağıtık bellekli mimari

Dağıtık paylaşımlı bellek

Bu model, genelde paylaşımlı bellekli mimariye sahip olan SMP makinelerin bir ağ hattı üzerinden birleştirilmesiyle oluşturulur. Genelde çok büyük ölçekteki problemlerin çözümünde kullanılmaktadır. Makineler kendi içerisinde paylaşımlı bellekli mimaride olduğu gibi yerel bellekleri kullanırlar ve bir başka SMP makineyle iş yükünü paylaşmak için dağıtık bellekte olduğu gibi ağ hattı üzerinden mesajlaşırlar. Yani sistem kendi içerisinde çoklu iş parçacıklarını aynı anda işlerken, ağ ortamında da süreçler arası haberleşmeler söz konusudur. Şekil 3.8.'de bu durum görselleştirilmiştir.



Şekil 3.8. Dağıtık paylaşımlı bellek

3.2.1.3. Paralel program tasarımı

Bir problemi paralel olarak çözebilmenin ilk şartı problemin paralel yapıya uygun olup olmadığını tespit etmektir (Lawrence Livermore National Laboratory, 2007).

Paralleleştirilebilir bir problemin en belirgin özelliği, işlem adımları arasında veri bağımlılıklarının olmamasıdır. Örneğin, bir molekülün birkaç bin yapısının her birisi için potansiyel enerjinin hesaplanması ve her bir yapının hesabı ayrı ayrı tamamlandıktan sonra tüm yapının minimum enerjisinin bulunması. Bu problem için, moleküle ait yapılar birbirinden bağımsız olarak hesaplanabildiği için paralelleştirilmeye uygundur diyebiliriz.

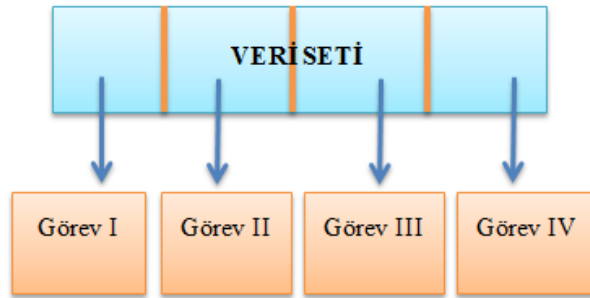
Paralleştirilemeyen bir problemde ise işlem adımları arasında bağımlılık söz konusudur. Bu duruma örnek olarak Fibonacci serisi verilebilir. Fibonacci serisi Denklem 3.1’ de ifade edilmiştir. Bu denklemde k , seri içerisindeki sayı indislerini ifade etmektedir.

$$F(k + 2) = F(k + 1) + F(k) \quad (3.1)$$

Bu problemde iterasyonlar arası bağımlılık söz konusu olduğu için, Fibonacci serisi paraleleştirilmeye uygun değildir. Bir problemin tamamı paralel çalışabildiği gibi sadece bazı bölgeleri paralel çalışıp kalan bölgeler seri olarak çalışabilir. Bu durumda kodun paralelliği arttıkça performansta aynı oranda artacaktır. Paralel bir program tasarlanırken ilk olarak problemin önemli bölgeleri çok iyi analiz edilmelidir. Problem içerisinde paralelleşmeyen yani veri bağımlılıklarının olduğu bölgeler tespit edilmelidir. Problem analiz edildikten sonra, probleme uygun bir ayrışım yöntemi seçilir. Temel olarak iki adet ayrışım yöntemi vardır. Bunlar;

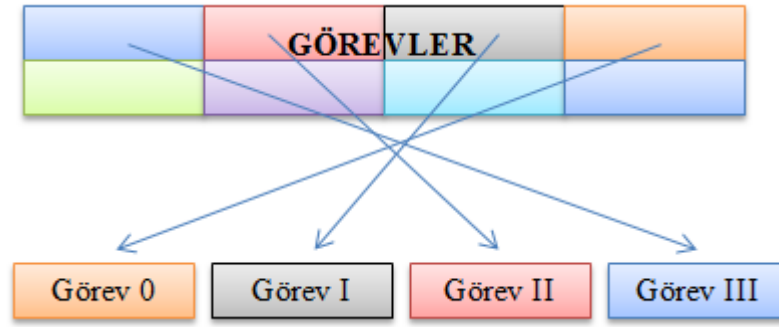
- Veri Ayrışımı (Domain Decomposition)
- Fonksiyonel Ayrışım (Functional Decomposition)

Veri ayrışımı yönteminde görevler arasında veri paylaşımı vardır. Her paralel görev kendisine verilen veri parçası üzerinde çalışır.



Şekil 3.9. Veri ayrışımı

Fonksiyonel ayrışımında ise her paralel görev farklı bir komut setini icra eder. Yani görevler farklı komutları icra etmektedirler. Bu duruma örnek olarak, bir sinyal verisinin üç farklı filtreden geçirilmesi verilebilir.



Şekil 3.10. Fonksiyonel ayrışım

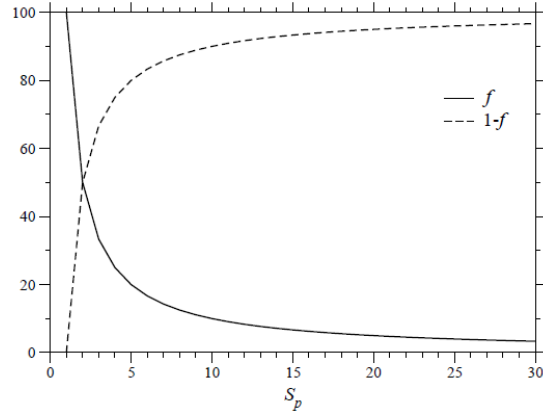
3.2.1.4. Amdahl kanunu, paralel hızlanma ve performans

Performans: Bir kullanıcı için performans, herhangi bir görevin başlama ve bitiş zamanı arasındaki geçen süredir. Bu süre cevap zamanı ya da çalışma zamanı olarak da bilinir. Performans artışı ise bu sürenin kısalmasına bağlıdır. Bir başka açıdan bakılacak olursa amaç belirene kadar bir zaman diliminde yapılacak iş sayısının artırılmasıdır. Örneğin bir bilgisayarın işlemcisinin daha hızlı bir işlemciyle değiştirilmesiyle sistemin cevap zamanı azalacak ve aynı zamanda birim zaman diliminde daha fazla iş yapılacaktır yani performans artışı sağlanmış olacaktır.

CPU çalışma zamanı: İşlemcide yürütülen bir görevin tamamlanma süresidir. Bu süre giriş/çıkış zamanını ve diğer programları beklemek için geçen süreleri içermez.

Toplam süre: Bir programın yürütülmesiyle geçen toplam süre, CPU çalışma süresi ve bekleme süresinin toplamıdır. Buradaki bekleme süresi program içerisindeki giriş/çıkış işlemleri ve diğer bekleme sürelerinin toplamıdır.

Paralel hesaplama ile ilgili en yaygın gözlem, her bir algoritmanın paralel bir bilgisayarda elde edilebilecek hızlanmayı sınırlayan sıralı bir kısmının olduğudur. Bu durum “**Amdahl**” kanunu olarak adlandırılır ve şu şekilde ifade edilir. Eğer bir algoritmanın sıralı bileşeni programın toplam yürütme zamanının f kadarı ise o zaman paralel bir hesaplamada elde edilebilecek maksimum hızlanma $1/f$ kadardır. Örneğin sıralı bileşen %5 ise, elde edilebilecek maksimum hızlanma 20’dir. Şekil 3.11’ de yer alan grafikte, hızlanmanın yaklaşık 2 olduğu durumda hesabın yaklaşık olarak %50’sinin paralelleştirilebildiği görülmüştür (Karasulu, 2006).

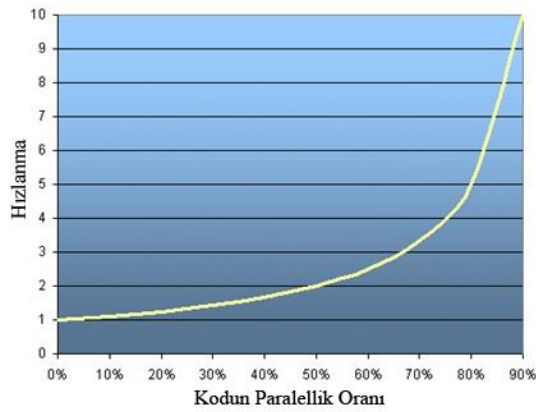


Şekil 3.11. f ve $(1-f)$ ' nin hızlanma ile değişimi (Lawrence Livermore National Laboratory, 2007)

Amdahl kanununa göre paralel hızlanma, Denklem 3.2 ve 3.3' de ifade edildiği gibidir.

$$Hızlanma = \frac{1}{1-P} \quad (3.2)$$

Denklem 3.2' ye göre seri kodun hiçbir bölgesi paralelleşmiyorsa yani $P=0$ ise hızlanma 1' dir yani hızlanma yoktur. Eğer seri kodun tamamı paralelleşebiliyorsa yani $P=1$ ise hızlanma teorik olarak sonsuzdur. Eğer seri kodun % 50' si paralelleşebiliyorsa maksimum hızlanma 2 kat olur. Bu durum Şekil 3.12.' de görselleştirilmiştir.

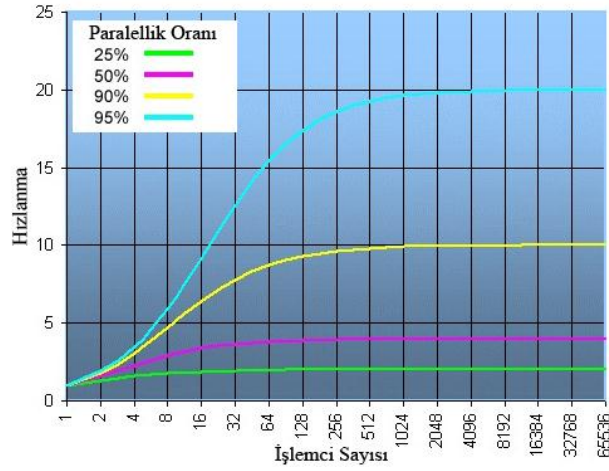


Şekil 3.12. Kodun paralellik oranı ile hızlanma arasındaki ilişki (Lawrence Livermore National Laboratory, 2007)

Denklem 3.3' te ise kodun paralelleşmesiyle birlikte işlemci sayısının artışının hızlanma üzerindeki etkisi ifade edilmiştir.

$$Hızlanma = \frac{1}{\frac{P}{N} + S} \quad (3.3)$$

Denklem 3.3' te P kodun paralel bölümü, N işlemci sayısı ve S ise kodun seri bölümünü ifade etmektedir.



Şekil 3.13 İşlemci sayısı ile hızlanma arasındaki ilişki (Lawrence Livermore National Laboratory, 2007)

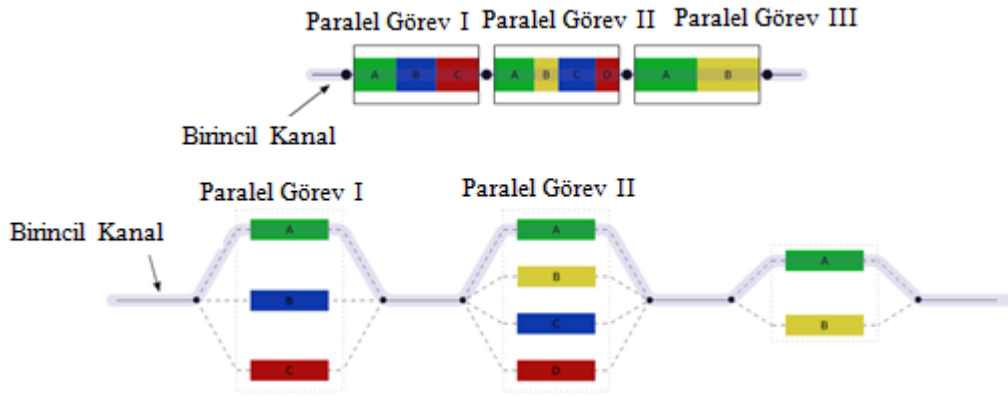
Amdahl kanunu paralel bir hesaplamada elde edilebilecek en iyi hızlanma oranının bulunmasını sağlar ancak gerçek uygulamada bu oran daha düşüğe çıkabilir. Bunun en iyi gözlemlendiği yer dağıtık bellekli sistemlerde yapılan paralel uygulamalardır. Bu uygulamalar göz önüne alındığında Amdahl kanununda, paralelleştirmeye gelen işlemciler arası iletişim süreleri, işlemcilerin boşa bekleme süreleri gibi ilave maliyetlerin ihmal edildiği görülmektedir.

3.2.1.5. OpenMP

OpenMP bilişim sektörünün önde gelen donanım ve yazılım üreticilerinin bir araya gelerek oluşturdukları bir uygulama ara yüzüdür. OpenMP ortak paylaşımlı bellekli paralel uygulama geliştiren kullanıcılar için taşınabilir ve ölçeklenebilir bir modeldir. OpenMP uygulama ara yüzü C/C++ ve Fortran dilleriyle Unix ve Windows işletim sistemleri üzerinde paralel uygulama geliştirme imkânı sağlar. OpenMP ile

geliştirilen paralel uygulamalar, paralel programlama desteği bulunmayan derleyicilerde de derlenebilir ancak OpenMP komutları yorum olarak algılanır. Bu özellik OpenMP' nin taşınabilir olmasını sağlamaktadır.

OpenMP, uygulamaların çok kanallı olarak yürütülmesini sağlar ve bu kanallar ortak bellek alanını kullanırlar. Bu çoklu kanal uygulamasıyla sağlanan paralel modelde çatallaş-birleş(fork-join) yöntemi kullanılır yani bir adet birincil kanal(master thread) vardır ve bu birincil kanal paralel bölgelerde çatallaşır ve alt(köle) kanallara ayrılır. Yapılacak işler bu kanallara paylaştırılarak eş zamanlı olarak çalıştırılır. Paralel bölgede işlevini tamamlayan kanallar paralel bölgenin sonunda ölürler ve birincil kanal programı yürütmeye devam eder. Ayrıca paralel bölgelerde her kanala ait bir tanımlayıcı rakam atanmaktadır. Şekil 3.14.' te bu durum görselleştirilmiştir.



Şekil 3.14. OpenMP çatallaş-birleş metodu

3.2.2. Dalgacık dönüşümü

Pratikte kullanılan birçok sinyal zaman ve genlik ekseninde tanımlanır. Bir sinyal için en önemli parametrelerden birisi frekanstır. Frekans ise bir sinyalin birim zamanda gösterdiği değişimdir. Bundan dolayı bir sinyale ait önemli veriler frekans düzleminde saklıdır. Fourier dönüşümü ve dalgacık dönüşümü sinyal işleme uygulamalarında yaygın olarak kullanılan sinyal analiz yöntemleridir. Bir sinyale Fourier dönüşümü uygulandığı takdirde o sinyale ait frekans-genlik bilgileri elde edilir.



Şekil 3.15. Fourier dönüşümü (Misiti et al, 1996)

Bir sinyale ait frekans bilgisini elde etmemizi sağlayan Fourier dönüşümünün en önemli eksiği frekans alanında iken, zaman bilgisini kaybetmesidir (Misiti et al, 1996). Yani bir işaret ile birim zamanda tek bir dönüşüm yapılabilmektedir. Bu dönüşüm, zaman alanından frekans alanına veya frekans alanından zaman alanına yapılabilir. Bu durum Fourier dönüşümünün belirli zaman aralıklarında frekans değeri değişmeyen sinyaller için kullanılabileceğini göstermektedir.

Bir başka sinyal analiz yöntemi olan dalgacık dönüşümü ile Fourier dönüşümündeki bu eksiklik giderilmiştir. Yani dalgacık dönüşümü ile bir sinyalin birim zamanda frekans bilgileri elde edilirken zaman bilgileri kaybolmamaktadır. Bunun yanında, birim zamanda zaman alanından frekans alanına geçilebildiği gibi, frekans alanından zaman alanına ters dönüşümde yapılabilmektedir.

İlk dalgacık dönüşümü, matematikçi Alfred Haar tarafından 1909 yılında bulunmuştur. En basit dalgacık dönüşümü olan Haar, Daubechies D-2 dönüşümü olarak da bilinmektedir. Haar dönüşümünde verilerin ikiye ikiye ortalaması alınır ve ayrıca ortalamadan oluşan farkta kaybedilmez. Bu işlem sürekli olarak devam eder ve işlem sonucunda bir tane toplam ortalama değer ile $2^n - 1$ adet fark oluşur (Graphs, 1995).

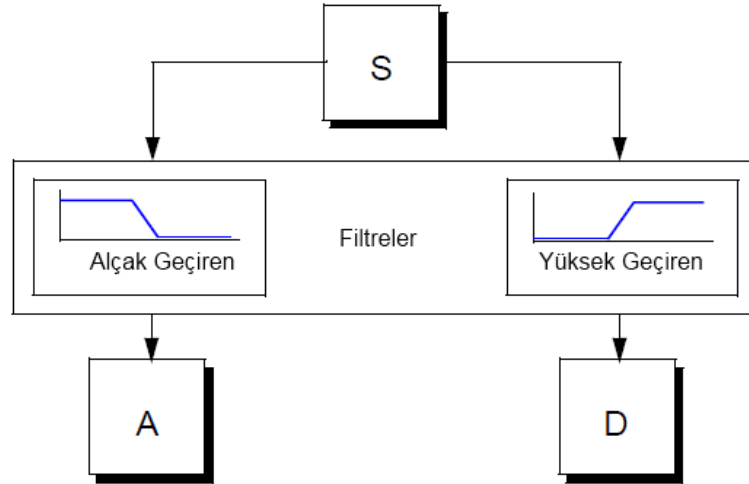
Fourier dönüşümünün zaman bölgesindeki dezavantajının aksine dalgacık dönüşümü ile zaman ve frekans bölgelerinde analiz yapılabilmesi, sinyalin yerel ve genel özelliklerinin çıkarılmasını sağlamaktadır.

3.2.2.2. Ayrık dalgacık dönüşümü

Sürekli dalgacık dönüşümü, ana dalgacığın ölçeklenerek ve ötelenerek tüm zaman boyunca sinyal ile çarpılıp sonuçların toplanmasıdır. Bu süreç sonunda sinyalin ölçek ve konumuna bağlı olarak dalgacık katsayıları elde edilir. Bu işlem işlenecek işaretin boyutuna bağlı olarak çok uzun sürebilmekle beraber sistem kaynaklarının aşırı kullanımında söz konusu olabilmektedir.

Ayrık dalgacık dönüşümünde ise ölçekleme ve öteleme işlemleri ikinin üsleri şeklinde gerçekleştirilerek dalgacık katsayıları çok daha kısa sürelerde ve daha etkili olarak elde edilebilmektedir. Ayrık dalgacık dönüşümünün temelleri 1976 yıllarına dayanmaktadır.

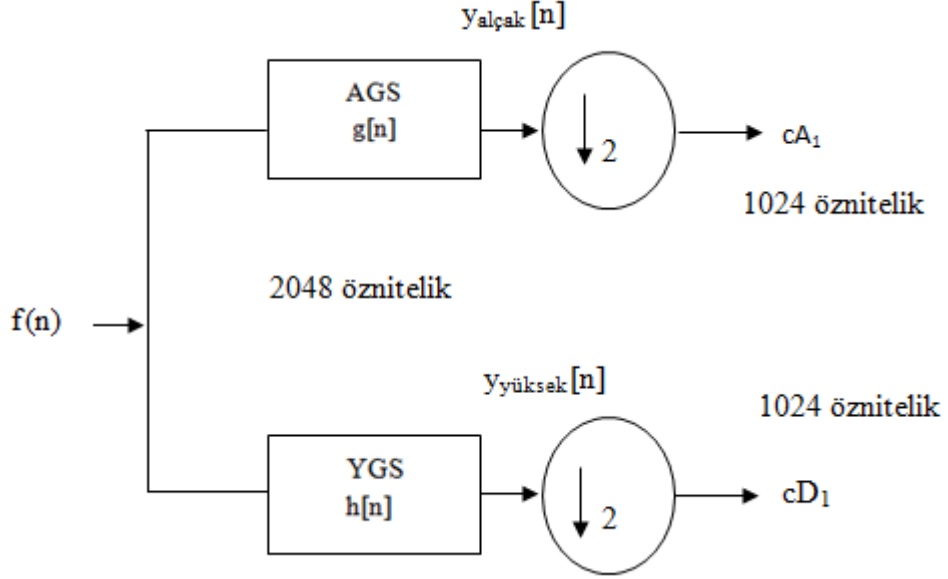
Birçok sinyal için alçak frekans bilgileri sinyalin en önemli parçasıdır. Yüksek frekans bilgileri ise daha çok sinyale ait detay bilgilerini içermektedir. Sinyal işleme uygulamalarında alçak frekans verileri yaklaşım, yüksek frekans verileri ise detay katsayıları olarak adlandırılmaktadır. Şekil 3.16 'da bir sinyalin alçak ve yüksek geçiren filtrelerden geçirilmesiyle elde edilen yaklaşım ve detay katsayıları elde edilmiştir.



Şekil 3.16. Bir sinyalin alçak ve yüksek geçiren filtrelerden geçirilmesi (Misiti et al, 1996)

Tek boyutta dalgacık dönüşümü

Bir $f[n]$ sinyalinin ayrık dalgacık dönüşümü, bu sinyalin alçak geçirenlikli $g[n]$ ve yüksek geçirenlikli $h[n]$ filtresinden geçirilmesiyle hesaplanır. Buradaki $g[n]$ ve $h[n]$ filtreleri sinyalin oluşturan verilerin dağılımına göre seçilmektedir. Şekil 3.17 'de bu durum görselleştirilmiştir.



Şekil 3.17. Bir sinyalin bir seviyeli çözünürlükte bileşenlerine ayrıştırılması

Denklem 3.4 ve 3.5'de $f[n]$ sinyalinin alçak geçiren filtre ve yüksek geçiren filtreden geçirilmesiyle elde edilen cA_1 ve cD_1 katsayıları ifade edilmiştir.

$$cA_1 = \sum_{k=-\infty}^{\infty} f[k]g[2n - k] \quad (3.4)$$

$$cD_1 = \sum_{k=-\infty}^{\infty} f[k]h[2n - k] \quad (3.5)$$

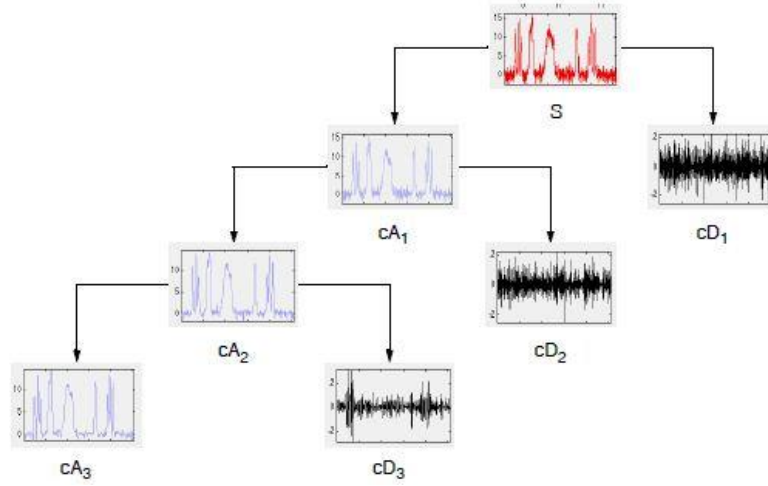
Denklem 3.4 'de elde edilen cA_1 katsayıları sinyalin alçak geçiren filtreden geçirilmesiyle elde edilen yaklaşım katsayılarıdır. Sinyale ait genel özellikleri taşır

ve orijinal sinyalden çok daha az gürültü içermektedir. Bir sinyalin alçak frekans bileşenleri sinyali temsil eden en önemli öznelik grubudur.

Denklem 3.5’de elde edilen cD_1 katsayıları ise sinyalin yüksek geçiren filtreden geçirilmesiyle elde edilir ve sinyale ait detayları içermektedir. cD_1 katsayıları çok küçük değerlerden oluşur ve daha çok sinyaldeki gürültülerdir.

Şekil 3. 17’ de 2048 adet öznelikten oluşan bir sinyal yer almaktadır. Bu sinyal alçak geçiren ve yüksek geçiren filtrelerden geçirildikten sonra ilk etapta 2048 adet yaklaşım katsayısı ve 2048 adet ayrıntı katsayısı elde edilmiş olur. Bunun sonucunda toplamda 4096 katsayı elde edilir.

Bir sinyal tek seviyede bileşenlerine ayrılabilirdiği gibi, çok seviyeli ayrıştırma yapmakta mümkündür. İlk ayrıştırmadan sonraki ayrıştırmalar, genelde sinyal ile ilgili daha önemli bilgileri taşıyan yaklaşım katsayıları üzerinden gerçekleştirilir. Çok seviyeli ayrıştırmada ayrıştırma seviyesi 2^n ifadesindeki n sayısı ile belirlenir. Şekil 3.18.’de tek boyutlu bir sinyalin çok seviyeli ayrıştırılması görselleştirilmiştir.

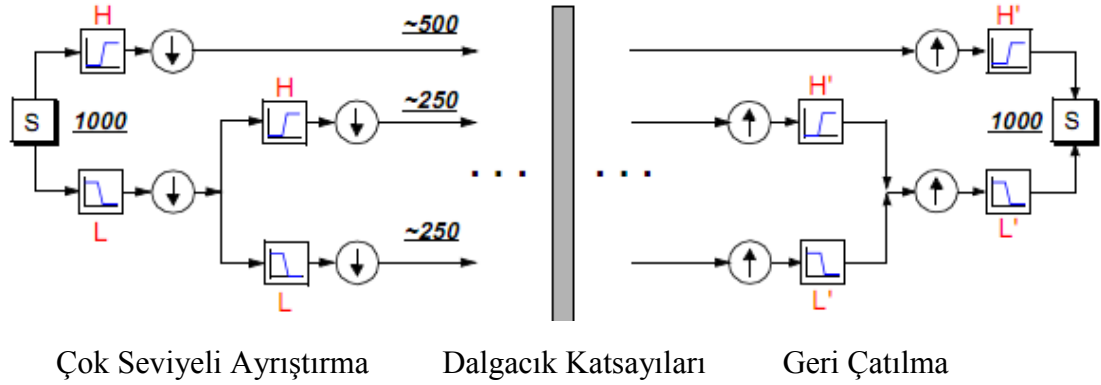


Şekil 3.18. Bir sinyalin çok seviyeli ayrıştırılması(Misiti et al, 1996)

Bileşenlerine ayrılan bir sinyal, yaklaşım ve detay katsayıları kullanılarak tekrar elde edilebilmektedir. Bu işleme “Sentez” veya “Geri Çatılma” denilmektedir. Denklem 3.6 ’da yer alan eşitlikle, j ayrıştırma seviyesinde alçak ve yüksek geçiren filtreler ile yaklaşım ve detay katsayıları elde edilen bir $f(x)$ fonksiyonu tekrar elde edilebilir.

$$c_{j,k} = \sum_n h(n-2k)c_{j-1,n} + g(n-2k)w_{j-1,n} \quad (3.6)$$

Örneğin j seviye ayrıştırma için elde edilen cA_j ve $cD_1 - cD_j$ katsayıları Şekil 3.19'da verilen yöntem kullanılarak tekrar elde edilebilir. Sinyalin geri çatılması işleminde, yukarı doğru örnekleme işlemi gerçekleştirilir. Bu işlem gerçekleştirilirken, dalgacık katsayıları dizilerinde her iki örnek arasına bir "0" genlikli örnek eklenir. Böylece aşağıya doğru örnekleme işlemlerinde göz ardı edilen detay verilerinin, sinyalin tekrar elde edilmesini engelleyecek bilgi kaybına yol açmadığı görülür (Toprak, 2007).



Şekil 3.19. Bir sinyalin geri çatılması (Misiti et al, 1996)

İki boyutlu dalgacık dönüşümü

İki boyutta dalgacık dönüşümünde de tek boyutlu dönüşümde olduğu gibi veri alçak ve yüksek geçiren filtrelerden geçirilerek katsayılar elde edilir. Bir resme uygulanan 2 boyutlu dalgacık dönüşüm işlemi sonucunda dört bileşen elde edilir. Bunlar;

- Alçak Geçiren Satırlar ve Sütunlar (LL)
- Yüksek Geçiren Satırlar ve Alçak Geçiren Sütunlar (HL)
- Alçak Geçiren Satırlar ve Yüksek Geçiren Sütunlar (LH)
- Yüksek Geçiren Satırlar ve Sütunlar (HH)

Matematiksel olarak bu işlem sırasıyla Denklem 3.7, 3.8, 3.9 ve 3.10 'de ifade edilmiştir. Bu ifadeler tekrarlı olarak her satır ve sütun için tekrar edilir.

$$A_{LL,k_1,k_2}^{(n-1)} = \sum_{l_1,l_2} h_{l_1-2k_1} h_{l_2-2k_2} A_{LL,l_1,l_2}^{(n)} \quad (3.7)$$

$$D_{LH,k_1,k_2}^{(n-1)} = \sum_{l_1,l_2} h_{l_1-2k_1} g_{l_2-2k_2} A_{LL,l_1,l_2}^{(n)} \quad (3.8)$$

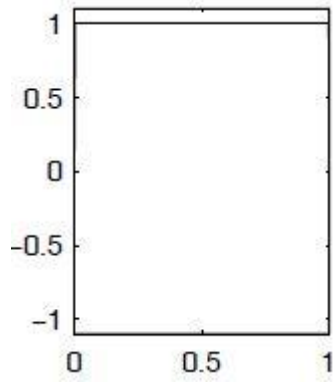
$$D_{HL,k_1,k_2}^{(n-1)} = \sum_{l_1,l_2} g_{l_1-2k_1} h_{l_2-2k_2} A_{LL,l_1,l_2}^{(n)} \quad (3.9)$$

$$D_{HH,k_1,k_2}^{(n-1)} = \sum_{l_1,l_2} g_{l_1-2k_1} g_{l_2-2k_2} A_{LL,l_1,l_2}^{(n)} \quad (3.10)$$

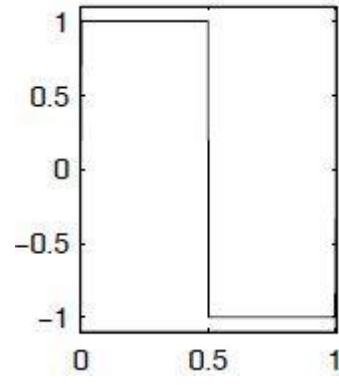
Ana dalgacığın seçilmesi

Dalgacık dönüşümünde, dalgacık filtre seçimi, temsil özelliği yüksek özniteliklerin elde edilmesinde önemli bir rol oynar. Aslında hangi dalgacığın seçileceği konusunda standart bir yöntem yoktur. Dalgacık seçilirken üzerinde çalışılacak sinyale zaman düzleminde görsel olarak ne kadar çok benzerse dalgacık dönüşüm yöntemi o kadar verimli olur. Fakat bu şekilde bir seçim yapmak neredeyse olanaksız gibidir. Bundan dolayı diğer tüm dalgacıklar denenmelidir.

Haar, Daubechies, Symlets, Coiflets ve Biorthogonal gibi uygulamada kullanılan birçok dalgacık ailesi vardır. Haar en basit dalgacık türüdür. Şekil 3.20'de Haar dalgacığına ait ölçekleme ve dalgacık fonksiyonu, Şekil 3.21'de ise yine Haar dalgacığına ait alçak geçiren ve yüksek geçiren filtre katsayıları yer almaktadır.

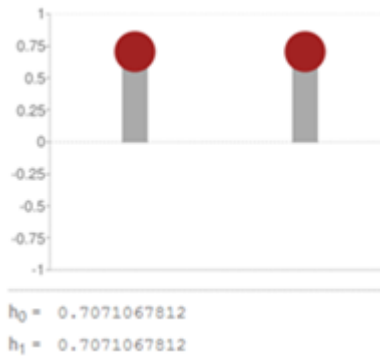


a) Ölçekleme Fonksiyonu

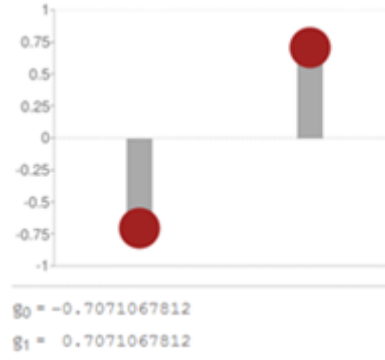


b) Dalgacık Fonksiyonu

Şekil 3.20.Haar ölçekleme ve dalgacık fonksiyonu



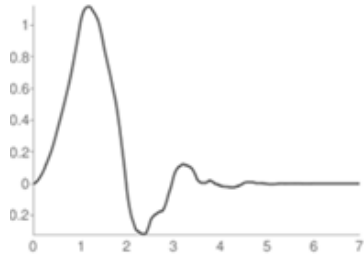
a) Alçak Geçiren Filtre Katsayıları



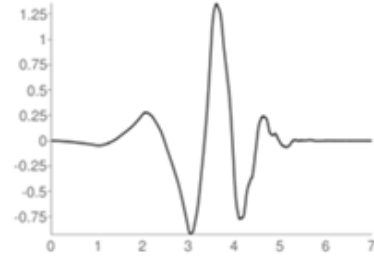
b)Yüksek Geçiren Filtre Katsayıları

Şekil 3.21.Haar alçak ve yüksek geçiren filtre katsayıları

Şekil 3.22' de ise uygulamalarda çok sık kullanılan "Daubchies 4" dalgacığın ait ölçekleme ve dalgacık fonksiyonu, Şekil 3.23'de ise yine "Daubchies 4" dalgacığın ait alçak geçiren ve yüksek geçiren filtre katsayıları yer almaktadır.

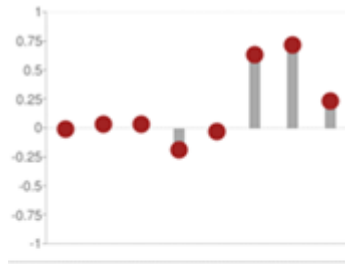


a) Ölçkleme Fonksiyonu

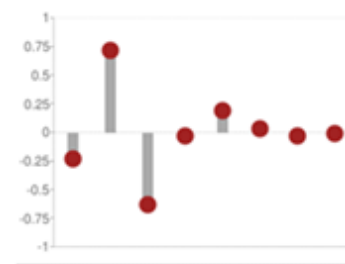


b) Dalgacık Fonksiyonu

Şekil 3.22. Daubchies 4 ölçkleme ve dalgacık fonksiyonu



$h_0 = -0.0105974018$
 $h_1 = 0.0328830117$
 $h_2 = 0.0308413818$
 $h_3 = -0.1870348117$
 $h_4 = -0.0279837694$
 $h_5 = 0.6308807679$
 $h_6 = 0.7148465706$
 $h_7 = 0.2303778133$



$g_0 = -0.2303778133$
 $g_1 = 0.7148465706$
 $g_2 = -0.6308807679$
 $g_3 = -0.0279837694$
 $g_4 = 0.1870348117$
 $g_5 = 0.0308413818$
 $g_6 = -0.0328830117$
 $g_7 = -0.0105974018$

a) Alçak Geçiren Filtre Katsayıları

b)Yüksek Geçiren Filtre Katsayıları

Şekil 3.23. Daubchies 4 alçak ve yüksek geçiren filtre katsayıları

3.2.3. Kenar çıkarma ve eğim

Sobel kenar çıkarma operatörü iki boyutlu uzaysal eğim ölçümü yapar ve böylece kenarlara benzeyen yüksek uzaysal frekanslı alanların üzerinde durur. Genellikle girdi olarak verilen gri görüntünün her bir noktasındaki yaklaşık eğim büyüklüğünü bulmak için kullanılır. Şekil 3.24 'te Sobel operatörüne ait yatay ve düşey matrisler gösterilmiştir.

1	2	1
0	0	0
-1	-2	-1

Düsey bileşen

-1	0	1
-2	0	2
-1	0	1

Yatay bileşen

Şekil 3.24. Sobel kenar çıkarma operatörü yatay ve düşey matrisleri

Bu 3x3 boyutundaki matrisler imgeye ayrı ayrı uygulanarak G_x ve G_y olarak adlandırılan iki ayrı eğim bileşeni elde edilir. Bu eğim bileşenleri kullanılarak her bir piksele ait mutlak eğim büyüklüğü ve yönü bulunabilir.

$$\text{Yön büyüklüğü için:} \quad |G| = \sqrt{G_x^2 + G_y^2} \quad (3.11)$$

$$\text{Kenar yön açısı:} \quad \theta = \arctan(G_y/G_x) \quad (3.12)$$

ifadeleri ile bulunmaktadır (Yavuz, 2008).

Sobel kenar çıkarma filtresini oluşturan matrisler Şekil 3.24 'te görüldüğü gibi bir resmin düşey ve yatay kenar piksellerinin elde edilmesini sağlamaktadır.

Yapısal öznitelik çıkarma yöntemlerinden zincir kodu yöntemi ile yapılan çalışmalarda karakterler için genele dörtlü ve sekizli kodlama yapılmaktadır. Bu çalışmada geleneksel dörtlü ve sekizli kodlama yerine on ikili yön kodlaması yapılmıştır. Bu yöntemde kenar yön açıları 0 ile 2π aralığında otuzar derecelik açı aralıklarında bir ile on iki rakamları arasında kodlanmaktadır (Singh et al, 2010).

3.2.4. Destek vektör makineleri

Vapnik tarafından geliştirilen SVM, makine öğrenmesi algoritmaları arasında yer almasına rağmen günümüzde veri madenciliği uygulamalarında sınıflandırma amacıyla kullanılmaktadır. SVM makinelerinde sınıflandırma işlemi doğrusal ya da doğrusal olmayan bir fonksiyon yardımıyla gerçekleştirilmektedir. SVM lineer olmayan örnek uzayını, örneklerin lineer olarak ayrılabilirliği bir yüksek boyuta

aktararak, farklı örnekler arasındaki maksimum sınırın bulunması esasına dayanır. SVM 'de sınıflandırılacak veriler lineer olarak ayrılabilirler gibi, lineer olarak ayrılamayan durumlarda mevcuttur. Günlük yaşam problemlerinin büyük bir çoğunluğu lineer olarak ayrılamazlar. Lineer olarak ayrılmış olan veriler arasında maksimum sınırın bulunması oldukça kolaydır, ancak lineer olarak ayrılamayan veriler öncelikle lineer olarak ayrılacakları farklı bir uzaya aktarılmalıdırlar (Demirci, 2007).

3.2.4.1. Lineer destek vektör makineleri

Lineer SVM sınıflandırıcıları lineer olarak ayrılabilen ve lineer olarak ayrılamayan veriler ile işlem yapabilen SVM sınıflandırıcıları olarak iki ayrı grupta incelenecektir.

Lineer olarak ayrılabilme durumu

Eğitim amacıyla kullanılacak N elemandan oluşan verinin $x = \{x_i \text{ ve } y_i\}$, $i=1,2,\dots,N$ olduğunu varsayalım. Burada $y_i \in \{-1,1\}$ etiket değerleri ve $x_i \in \mathcal{R}^d$ özellikler vektörüdür. Lineer durumda veriler doğrudan bir aşırı düzlem ile ayrılabilir. SVM 'in amacı bu aşırı düzlemin iki örnek grubuna eşit mesafede olmasını sağlamaktır. SVM lineer olarak ayrılamayan verileri öncelikle yüksek boyutlu bir biçime dönüştürür. Bu yüksek boyut verinin özelliklerinin boyutundan daha büyük derecede olacaktır. Burada amaç, verilerin bir aşırı düzlem ile ayrılmasını sağlamaktır. Aşırı düzlem üzerindeki herhangi bir x noktası, w aşırı düzlemin normali ve $|b|/\|w\|$ hiper-uzayın orijine dik uzaklığı olmak üzere;

$$w^T x + b = 0 \quad (3.13)$$

koşulunu sağlar. SVM yönteminde, örnekler lineer olarak ayrılabilirse, $y_i=+1$ ve $y_i=-1$ etiketli örneklere eşit uzaklıkta olan optimum ayırıcı aşırı düzlemin bulunmasıdır. Bunun için eğitim setinin aşağıdaki yapıları sağladığını varsayalım:

$$y_i = +1 \text{ için, } w^T x_i + b \geq +1 \quad (3.14)$$

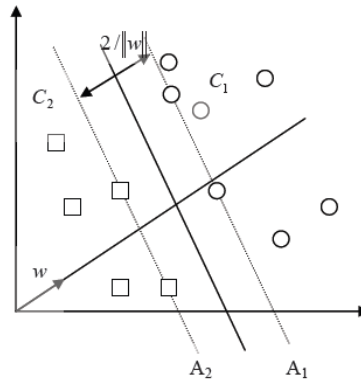
$$y_i = -1 \text{ için, } w^T x_i + b \leq -1 \quad (3.15)$$

Bu eşitsizlikler bir arada ifade edilecek olursa, $i=1,2,\dots,N$ için,

$$y_i(w^T x_i + b) \geq +1 \quad (3.16)$$

elde edilir. Denklem 3.16 daki koşulu sağlayan aşırı düzlemin iki tarafındaki en yakın örneklere olan dik uzaklıkları toplamı sınır olarak adlandırılır.

Şekil 3. 25' de optimum ayırıcı aşırı düzlem, yani sınırı maksimum yapan aşırı düzlem gösterilmiştir.



Şekil 3.25. Optimum ayırıcı düzlem

Denklem 3.14 'de yer alan eşitsizliğin Şekil 3.25'deki C_1 sınıfını ayıran A_1 aşırı düzlemini oluşturan eşitsizlik, Denklem 3.15 'deki eşitsizliğin ise aynı şekildeki C_2 sınıfını ayıran A_2 aşırı düzlemini oluşturan eşitsizlik olduğunu varsayalım. Bu durumda A_1 aşırı düzleminin orijine dik uzaklığı $|1-b|/||w||$ ve A_2 aşırı düzleminin orijine uzaklığı $|-1-b|/||w||$ olacaktır. Bu iki aşırı düzlemin optimal aşırı düzleme uzaklıkları ise $1/||w||$ kadardır, yani iki örnek kümesi arasındaki uzaklık A_1 ve A_2 aşırı düzlemlerinin birbirlerine paralel olmalarından dolayı $2/||w||$ kadardır. Burada dikkat edilirse A_1 ve A_2 aşırı düzlemleri arasında eğitim verilerine ait hiçbir örnek bulunmamaktadır. Bu iki aşırı düzlem arasındaki maksimum uzaklık ise $||w||$ değerinin minimize edilmesiyle bulunabilir. SVM yöntemiyle yapılmaya çalışılan bu iki aşırı düzlemin arasındaki uzaklığın maksimum olmasını sağlamaktır. Aralarındaki sınır

maksimum olan aşırı düzlemlerin üzerinde yer alan noktalara ise Destek Vektörler adı verilmektedir (Demirci, 2007). Maksimum sınırın bulunması işlemi;

$$\min \frac{1}{2} \|w\|^2 \quad (3.17)$$

$$y_i(w^T x_i + b) \geq +1, \forall_i \quad (3.18)$$

ile ifade edilir.

Burada Denklem 3.17 çözülecek problem ve Denklem 3.18 ise problemin çözümü sırasında kullanılan koşuldur ve bu ifade ikinci dereceden optimizasyon problemidir. Bu optimizasyon problemi için Lagrange formülasyonu uygulanır. Lagrange formülasyonu ise,

$$L_P = \frac{1}{2} \|w\|^2 - \sum_{i=1}^N a_i y_i (w^T x_i + b) + \sum_{i=1}^N a_i \quad (3.19)$$

şeklindedir. Bu formülasyonda $a_i \geq 0$ değerleri pozitif Lagrange çarpanları olarak adlandırılır. Ancak Denklem 3.19'da ifade edilen formülasyonun çözülmesi oldukça karmaşıktır. Çözümün bulunması için Denklem 3.19'daki ifade Karush-Kuhn-Tucker (KKT) koşulları kullanılarak dual problemine dönüştürülür. Bu problem için KKT koşulları:

$$\frac{\partial L_P}{\partial w} = 0 \rightarrow w = \sum_i a_i y_i x_i \quad (3.20)$$

$$\frac{\partial L_P}{\partial b} = 0 \rightarrow \sum_i a_i y_i = 0 \quad (3.21)$$

şeklindedir. Bu koşullar Denklem 3.19'da yerine yazılacak olursa:

$$\begin{aligned} L_d &= \frac{1}{2} (w^T w) - w^T \sum_i a_i y_i x_i - b \sum_i a_i y_i + \sum_i a_i \\ &= -\frac{1}{2} (w^T w) + \sum_i a_i \end{aligned} \quad (3.22)$$

$$= \sum_i a_i - \frac{1}{2} \sum_{i,j} a_i a_j y_i y_j x_i^T x_j \quad (3.23)$$

$$a_i \geq 0, \forall i$$

ifadesi elde edilmiş olur (Demirci, 2007).

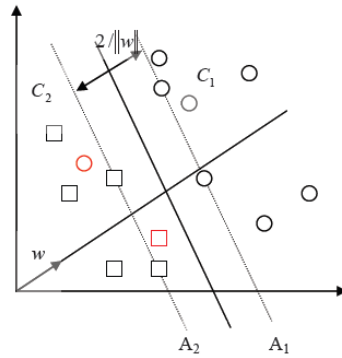
Lineer olarak ayrılamama durumu

Eğer örnekler lineer olarak ayrılabilir durumda değilse problemin çözümü için pozitif zayıflık değişkenleri, ξ , $i = 1, 2, \dots, l$ kullanılır. 3.14 ve 3.15 eşitsizlikleri bu zayıflık değişkenleri ile yeniden tanımlanacak olursa, yeni ifadeler 3.24 ve 3.25 eşitsizlikleri olacaktır.

$$y_i = +1 \text{ için, } w^T x_i + b \geq +1 - \xi_i \quad \xi_i \geq 0 \quad (3.24)$$

$$y_i = -1 \text{ için, } w^T x_i + b \leq -1 + \xi_i \quad \xi_i \geq 0 \quad (3.25)$$

$\xi_i = 0$ olması durumunda x_i örneği doğru sınıflandırılmış, $0 < \xi_i < 1$ olması durumunda x örneği doğru sınıflandırılmış, ancak A_1 ve A_2 aşırı düzlemleri arasında yer alıyor, $\xi_i \geq 1$ ise yanlış sınıflandırılmış demektir (Demirci, 2007).



Şekil 3.26. Lineer olmama durumunda optimal ayırıcı düzlem

Lineer olarak ayrılama durumunda eğitim verisi içinde olası her durum için bir çözüm üretmemesi için sisteme bir C üst sınırı eklenir. Bu üst sınır Lagrange çarpanlarının alabilecekleri maksimum değeri göstermektedir. Bu şekilde Lagrange çarpanlarının $0 \leq \alpha_i \leq C$ aralığında kalması sağlanmaktadır. Bu bilgilere göre Lagrange formülasyonu yeniden şu şekilde ifade edilecektir.

$$L_p = \frac{1}{2} \|w\|^2 + C \sum_i \xi_i - \sum_i \alpha_i \{y_i(w^T x_i + b) - 1 + \xi_i\} - \sum_i \mu_i \xi_i \quad (3.26)$$

Yukarıdaki formülasyonda μ_i , ξ_i 'nin pozitif olmasını garanti etmek için kullanılmış olan Lagrange parametreleridir. Bu Lagrange formülasyonunda çözülmesi zor olduğundan dolayı lineer ayrılabilir örneklerde olduğu gibi dual probleme dönüştürülmektedir. Bu probleme de KKT şartı uygulanırsa:

$$\frac{\partial L_p}{\partial w} = w - \sum_i \alpha_i y_i x_i = 0 \quad (3.27)$$

$$\frac{\partial L_p}{\partial b} = - \sum_i \alpha_i y_i = 0 \quad (3.28)$$

$$\frac{\partial L_p}{\partial \xi_i} = C - \alpha_i - \mu_i = 0 \quad (3.29)$$

ifadeleri elde edilir. Bu ifadeler Denklem 3.26 'da yerlerine yazılırsa:

$$L_d = \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j x_i^T x_j \quad (3.30)$$

$$0 \leq \alpha_i \leq C, \forall i$$

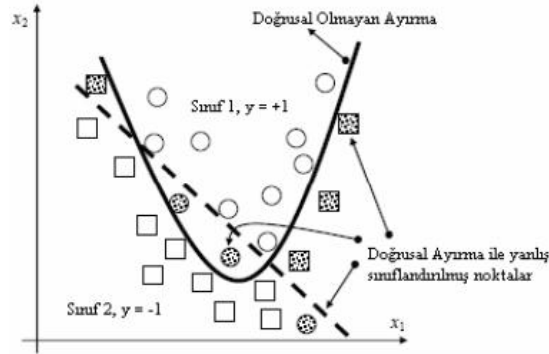
elde edilir. Bu problemin çözümünde $0 \leq \alpha_i \leq C$ aralığında yer alan Lagrange çarpanlarına karşılık gelen x_i değerleri destek vektörlerdir (Demirci, 2007).

3.2.4.2. Lineer olmayan destek vektör makinesi sınıflandırıcıları

Günlük hayatta karşılaşılan problemlerin büyük çoğunluğu farklı bileşenden oluşan problemlerdir. Lineer olarak ayrılamayan veriler, çekirdek fonksiyonları ile örneklerin öncelikle daha yüksek boyutlu ve doğrusal olarak ayrılacakları bir uzaya taşınıp, ardından çözümün bu yeni uzayda aranması ile mümkün olmaktadır.

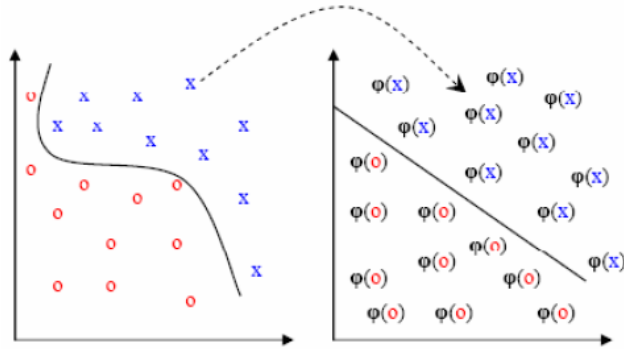
Giriş uzayındaki eğitim örneklerini bir H Euclid uzayına taşıyan Φ fonksiyonu göz önüne alınırsa,

$$\Phi: \mathcal{R}^d \mapsto H \text{ olur.}$$



Şekil 3.27. Doğrusal olmayan bir destek vektör makinesi

Şekil 3.27 'de bir doğrusal olmayan SVM; Şekil 3.28 'de ise bir giriş uzayı verilerinin doğrusal olarak ayrılacağı daha yüksek boyutlu bir özellik uzayına taşınması görülmektedir.



Şekil 3.28. Giriş uzayını özellik uzayına eşleme

Artık destek vektör makinelerinin eğitim aşaması, H uzayındaki verilerin $\Phi(x_i) \cdot \Phi(x_j)$ iç çarpımlarına bağlı olacaktır. Bu iç çarpım K ile gösterilir ise,

$$K(x_i, x_j) = \Phi(x_i) \cdot \Phi(x_j) \quad (3.31)$$

olarak ifade edilir. Burada K fonksiyonu çekirdek fonksiyonu olarak adlandırılmaktadır. Sonuç olarak karar fonksiyonu,

$$f(x) = \sum_{i=1}^{l_s} \alpha_i y_i \Phi(x_i) \cdot \Phi(x) + b \quad (3.32)$$

fonksiyonunun işareti ile belirlenir. Yani karar fonksiyonu yeniden yazılır ise,

$$= \text{sign}(f(x)) = \text{sign} \left(\sum_{i=1}^{l_s} \alpha_i y_i K(x_i, x) + b \right) \quad (3.33)$$

olur. Bu fonksiyonda l_s , destek vektörlerinin sayısı, x_i de destek vektörleridir (Eray, 2008). Çekirdek fonksiyonu olarak çeşitli fonksiyonlar kullanılabilir. Bunlardan bazıları şunlardır:

Radyal tabanlı çekirdek fonksiyonu(RBF)

Radyal tabanlı çekirdek fonksiyonu (RBF) Denklem 3.34'de görülmektedir.

$$K(x_i, x_j) = \exp \left(\frac{-\|x_i - x_j\|^2}{2\sigma^2} \right) \quad (3.34)$$

Polinom çekirdek fonksiyonu

Polinom çekirdek fonksiyonu Denklem 3.35 'de görülmektedir.

$$K(x_i, x_j) = (x_i^T x_j)^d \quad (3.35)$$

Doğrusal çekirdek fonksiyonu

Doğrusal çekirdek fonksiyonu Denklem 3.36 'da görülmektedir.

$$K(x_i, x_j) = (x_i^T x_j) \quad (3.36)$$

4. ARAŞTIRMA BULGULARI VE TARTIŞMA

Bu bölümde, öncelikle gerçekleştirilen çalışmanın uygulama platformu ve ön hazırlıkları anlatılmaktadır. Daha sonra, karakter resimlerinin ön işlemlerden geçirilmesi ve özniteliklerin elde edilmesi anlatılmaktadır. Bir sonraki kısımda elde edilen özniteliklerin üç ayrı sınıflandırıcı ile sınıflandırılması anlatılmakta ve elde edilen tanıma oranları çeşitli parametrelere bağlı olarak tablolar halinde sunulmaktadır.

Bu bölümün son kısmında ise sistemin seri ve paralel olarak çalıştırılmasıyla elde edilen sonuçlar tablolar halinde sunulmuştur. Gerçekleştirilen çalışmada başarı kriteri, sistemin paralel çalıştırılmasıyla elde edilen performans kazancı ve tanıma oranının istenilen seviyelere çıkması olarak belirlenmiştir.

4.1 Uygulama Platformu ve Ön hazırlıklar

Gerçekleştirilen çalışma MS Windows 7 işletim sistemi üzerinde, Intel Core 2 Quad CPU 2.66 GHZ işlemci ve 3 GB bellek donanımına sahip bir bilgisayarda çalıştırılmıştır. Hazırlanan programın seri ve paralel kodları tamamen C++ programlama dili ve OpenCV bilgisayarlı görme kütüphanesi kullanılarak yazılmıştır. Sistemin paylaşımlı bellek mimarisine göre paralel çalışabilmesi için, OpenMP paralel programlama kütüphanesi kullanılmıştır. Ayrıca çalışmada MS Visual Studio 2010 editörü ve Intel Parallel Studio 2011 aracı kullanılmıştır.

Sistemin uygulama safhasından önce ön hazırlık olarak MNIST veri tabanı dört parça olarak temin edilmiştir. Bunlar; 60.000 adet eğitim karakterlerinin yer aldığı 44,8 MB büyüklüğündeki görüntü dosyası, eğitim verilerine ait etiket bilgilerinin tutulduğu 60 KB büyüklüğündeki etiket dosyası, 10.000 adet test karakterlerinin yer aldığı 7.47 MB büyüklüğündeki görüntü dosyası ve test karakterlerine ait etiket bilgilerinin tutulduğu 12 KB büyüklüğündeki etiket dosyasıdır.

MNIST veri tabanı içerisindeki eğitim ve test karakterlerine ait görüntü dosyalarında her bir karakter 784 piksellik dizi elamanları ile temsil edilmektedir. Etiket dosyalarında ise her 784 piksellik elamanın 0-9 arasındaki rakam değeri tutulmaktadır.

4.2. Karakter İmgelerinin Ön İşlemden Geçirilmesi

MNIST veri tabanından okunan gri seviye karakter imgeleri ikili formata çevrilmiştir. Veri tabanı içerisinde yer alan karakterler 20x20 piksel ölçülerinde olup 28x28 piksel ölçülerindeki bir düzleme oturtulmuştur. Çalışmada iki farklı öznelik çıkarma yöntemi kullanıldığından karakter imgeleri iki farklı boyutta ölçeklenmiştir.

İlk öznelik çıkarma yönteminde, 2 boyutlu dalgacık dönüşümünde kullanılan aşağı örnekleme işlemi 2^n kuralına göre yapıldığı için her bir karakter 32x32 piksel olarak ölçeklendirilmiştir.

İkinci öznelik grubu için ise karakterlerin orijinal piksel ölçüleri olan 28x28 ölçeği kullanılmıştır.

4.3. Öznelik Çıkarma Aşamasında Yapılan Çalışmalar

4.3.1. İki boyutlu ayrık dalgacık dönüşümü ile özneliklerin elde edilmesi

32x32 piksel olarak ölçeklenen karakter imgelerinden 2 boyutlu ayrık dalgacık dönüşümü ile ilk öznelik grubu elde edilmiştir. Bu dönüşümde karakterler alçak geçiren ve yüksek geçiren süzgeçlerden geçirilmiştir. Alçak geçiren süzgeç imgenin yaklaşım katsayılarını verirken, yüksek geçiren süzgeç ise detay katsayılarını vermektedir. İmgeler alçak ve yüksek geçiren süzgeçlerden geçirildikten sonra bir seviye aşağı örnekleme yapılmış ve her bir karakter için 16x16 boyutunda yaklaşım ve 16x16 boyutunda detay (yatay, düşey, köşegen) katsayıları elde edilmiştir. Elde edilen ayrıntı katsayıları yüksek frekans bileşenli gürültü katsayılarını göstermektedir. Bu nedenle orijinal imgeden daha az gürültü içeren yaklaşım katsayıları öznelik olarak tercih edilmiştir.

Dalgacık dönüşümünde ana dalgacığın seçim işleminde belli bir yöntem veya kural yoktur. Seçilen dalgacık ile analiz edilen sinyalin zaman düzleminde benzerlik göstermesi etkili bir dönüşüm olarak düşünülebilir. Ancak bu durumu tek tek irdelemek olanaksızdır. Bundan dolayı bu çalışmada ana dalgacık olarak Biorthogonal, Coiflet, Haar ve Daubechies dalgacık aileleri tek tek denenmiş ve her bir dalgacık ailesinin tanıma oranına katkısı gözlemlenmiştir.

Çalışmanın ilk öznelik grubunu oluşturan dalgacık katsayıları her bir karakter için 256 adettir. Toplamda 60.000 eğitim karakterinden elde edilen öznelikler 60.000 x 256 boyutunda bir matriste saklanmıştır. Aynı şekilde 10.000 test karakterinden elde edilen öznelikler 10.000 x 256 boyutundaki bir matriste saklanmıştır.

4.3.2. Kenar çıkarma ve eğim öznelikleri

Bu tez çalışmasının ikinci öznelik grubunu karakter imgelerine ait piksellerin eğimlerinden elde edilen yön öznelikleri oluşturmaktadır. İlk olarak girdi imgesine Sobel kenar çıkarma operatörü uygulanarak düşey(g_y) ve yatay(g_x) bileşenler elde edilmiştir.

1	2	1
0	0	0
-1	-2	-1

Düşey bileşen

-1	0	1
-2	0	2
-1	0	1

Yatay bileşen

Şekil 4.1. Sobel kenar çıkarma operatörü yatay ve düşey matrisleri

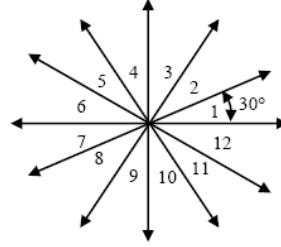
Daha sonra Denklem 4.1 ile imgenin kenar yön büyüklükleri ve Denklem 4.2 ile de imgenin kenar yön açıları elde edilmiştir.

$$|G| = \sqrt{G_x^2 + G_y^2} \quad (4.1)$$

$$\theta = \arctan(G_y/G_x) \quad (4.2)$$

İmge içerisinde yer alan siyah piksellere ait yön büyüklükleri 0 değerini alırken diğer piksellere ait yön büyüklük değerleri 0 ile $6.28(2\pi)$ arasında değerler almıştır. Kenar yön açıları ise 0 ile 2π arasında değerler almıştır. İmgeye ait her pikselin yön büyüklüğü hesaplandıktan sonra üst ve alt sınır değerleri göz önünde bulundurularak yön büyüklük değerleri 0- 2π aralığında 30° ar derecelik açılarla sınıflandırılmış ve 0 ile 12 arasındaki rakamlarla yön kodlaması yapılmıştır. Çizelge 3.2 'de kullanılan

yön kodları ve açı aralıkları sunulmuştur. Şekil 4.2 'de ise yön kodları ve açı aralıkları görselleştirilmiştir.



Şekil 4.2. Kenar yön açılarına göre yön kodlaması

Çizelge 4.1 Kenar yön açı aralıkları ve yön kodlaması

Piksele Ait Yön Büyüklük Değeri(g)	Atanan Yön Kodu
$g=0$	0
$0 \leq g \leq \pi/6$	1
$\pi/6 < g \leq \pi/3$	2
$\pi/3 < g \leq \pi/2$	3
$\pi/2 < g \leq 2\pi/3$	4
$2\pi/3 < g \leq 5\pi/6$	5
$5\pi/6 < g \leq \pi$	6
$\pi < g \leq 7\pi/6$	7
$7\pi/6 < g \leq 4\pi/3$	8
$4\pi/3 < g \leq 3\pi/2$	9
$3\pi/2 < g \leq 5\pi/3$	10
$5\pi/3 < g \leq 11\pi/6$	11
$11\pi/6 < g \leq 2\pi$	12

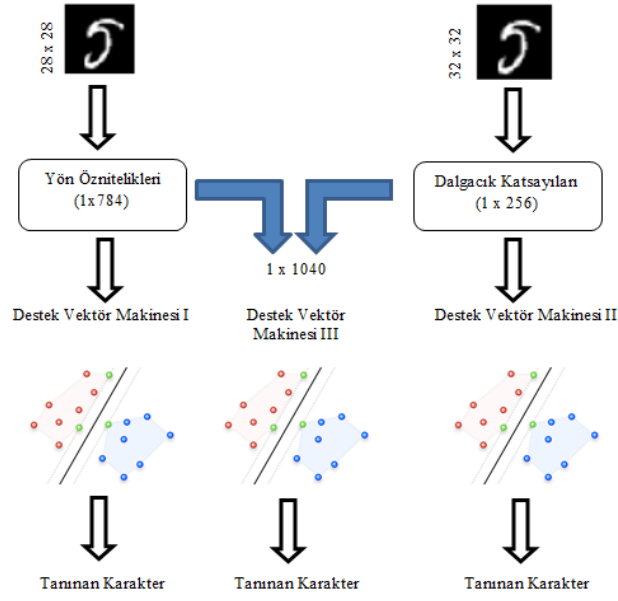
Her bir karakter için 28x28 piksel boyutunda kodlanmış yön öznitelikleri elde edilmiştir. Karakter imgelerinin siyah arka plan piksellerine ait yön kodları 0 olarak alınmıştır. Sınıflandırma aşamasında kullanılan bu karakter öznitelikleri eğitim sınıfı için 60.000 x 784, test sınıfı için ise 10.000 x 784 boyutlarındaki matrislerde tutulmuştur.

4.4. Sınıflandırıcılar ve Tanıma Oranları

Bu tez çalışmasının sınıflandırma aşamasında istatistiksel bir öznitelik çıkarım yöntemi olan Gradient yöntemi ile son zamanlarda oldukça popüler olan ve başarılı

sonular veren Dalgacık Dönüřüm yönteminin tanıma oranına etkileri karşılaştırılmıştır. Bu doğrultuda karakterlerden elde edilen farklı iki öznitelik grubu 3 ayrı destek vektör makinesi kullanılarak sınıflandırılmıştır. Buna göre karakter piksellerine ait yön özniteliklerini içeren ilk öznitelik seti ilk SVM, dalgacık dönüşümü ile elde edilen ikinci öznitelik seti ikinci SVM ve her iki öznitelik setinin birleştirilmesiyle oluşan üçüncü öznitelik seti ise üçüncü SVM tarafından sınıflandırma işleminde kullanılmıştır. Her bir SVM, Lineer, Polinom ve RBF çekirdek fonksiyonları ile ayrı ayrı test edilmiştir.

Gerçekleştirilen bu çalışmada SVM 'ler 0-9 arası rakamların sınıflandırılmasında kullanılmıştır. Yani SVM 'ler çoklu sınıflandırıcı olarak kullanılmıştır. SVM 'lerin çok sınıfı ayrıştırmasında iki farklı yaklaşım vardır. İlkinde Lagrange fonksiyonu kullanılarak sınıflandırma yapılmakta ancak bu yaklaşımda sınıf sayısı arttıkça hata oranı da artmaktadır. İkinci yaklaşımda ise bire karşı bir, bire karşı hepsi yöntemleri en sık kullanılan yöntemlerdir. Bu çalışma da bire karşı bir yöntemi kullanılmıştır. Bu yöntemde göre her bir rakam diğerleri ile ayrı ayrı eğitilmiştir. Bu işlemde 0-9 arası toplam 10 rakam olduğu için toplamda 45 eğitim ile tamamlanmıştır. SVM 'lerde yer alan C ve γ parametre değerlerinin seçiminde ise grid-search yöntemi kullanılmıştır. Buna göre C parametresi için $2^{-5}, 2^{-3}, \dots, 2^{15}$ arasında, γ parametresi için ise $2^{-15}, 2^{-13}, \dots, 2^3$ arasında üssel olarak değişen değerler test edilmiştir. Test edilen bu değerler arasında en iyi sonuçlar C: 2^{11} ve $\gamma :2^{-5}$ olduğu durumda elde edilmiştir. Şekil 4.3 'te öznitelik setlerinin oluşturulması ve sınıflandırılması görselleştirilmiştir.



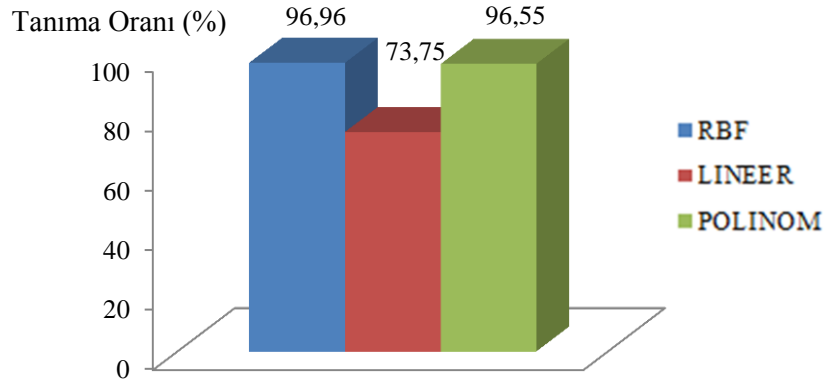
Şekil 4.3. Öznitelik setleri ve sınıflayıcılar

İlk öznitelik seti, eğitim sınıfı için 60.000 x 784 ve test sınıfı için 10.000 x 784 boyutundaki matrislerden oluşmaktadır. Bu öznitelik seti için ilgili SVM, RBF, Polinom ve Lineer olmak üzere üç farklı çekirdek fonksiyonu ile test edilmiştir. Her bir çekirdek fonksiyonuna ait tanıma oranları ve seri koda ait performans değerleri Çizelge 4.2’ de sunulmuştur.

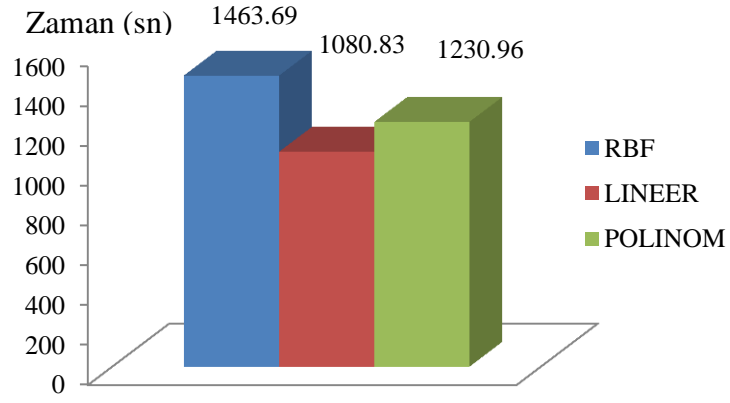
Çizelge 4.2. Piksel yönlerini içeren öznitelik seti ile elde edilen sonuçlar

Çekirdek Tipi	Eğitim Süresi (sn)		Test Süresi (sn)		Tanıma Oranı(%) (Test)	Tanıma Oranı(%) (Eğitim)
	CPU	Toplam	CPU	Toplam		
Lineer	1078.02	1080.83	741.06	742.56	73.75	74.86
Polinom	1070.64	1230.96	543.69	545,12	96.55	100
RBF	1462.02	1463.69	735.51	736.61	96,96	100

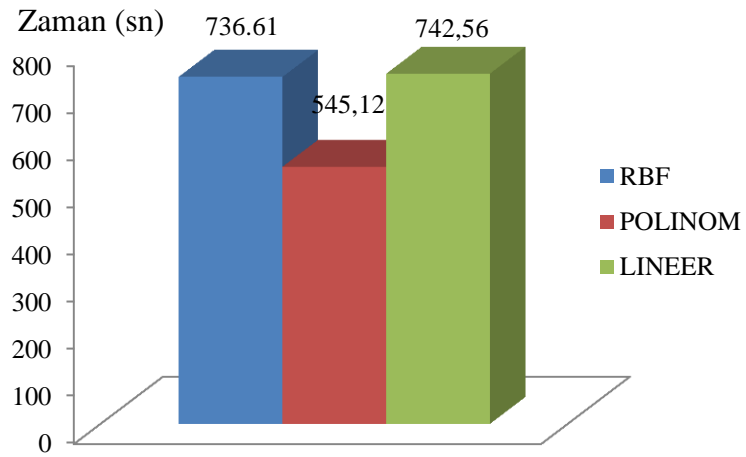
Yapılan uygulamaya ait tanıma oranları, eğitim ve test süreleri olmak üzere sırasıyla Şekil 4.4, 4.5 ve 4.6’ da görselleştirilmiştir.



Şekil 4.4. Çekirdek fonksiyonlarına göre doğru tanıma oranları.

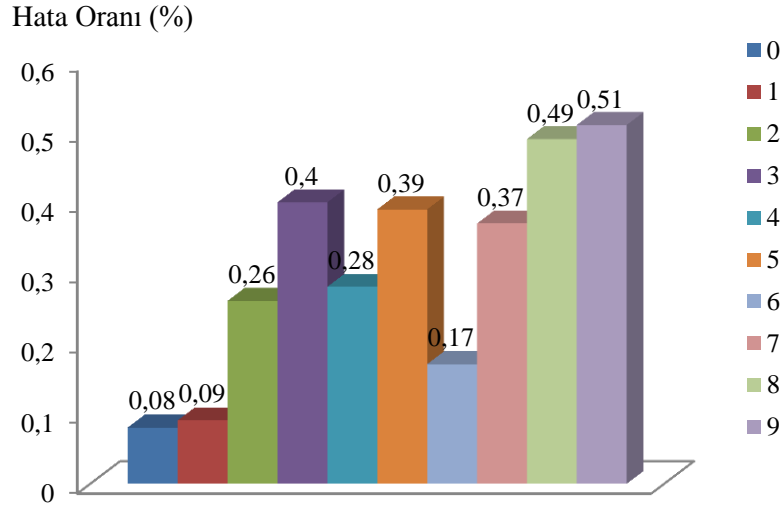


Şekil 4.5. Çekirdek fonksiyonlarına göre eğitim süreleri

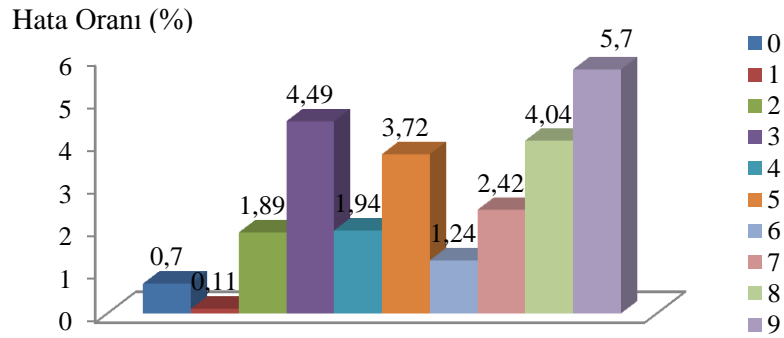


Şekil 4.6. Çekirdek fonksiyonlarına göre test süreleri

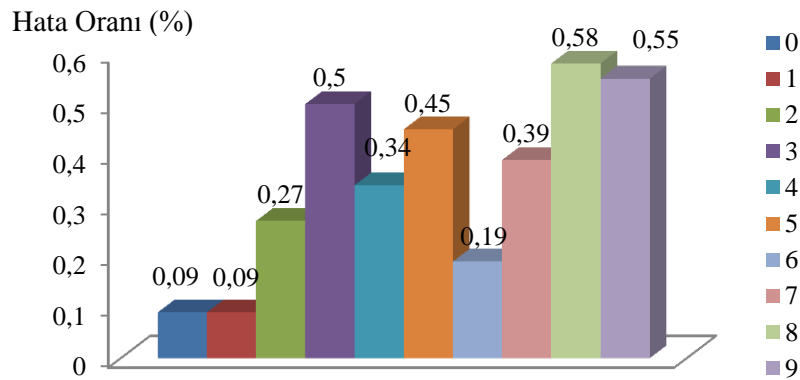
Şekil 4.7, 4.8 ve 4.9 'da ise sınıflandırma aşamasında kullanılan farklı çekirdek fonksiyonlarına ait hata dağılımları görselleştirilmiştir.



Şekil 4.7. RBF çekirdek fonksiyonu hata dağılımı



Şekil 4.8. Lineer çekirdek fonksiyonu hata dağılımı



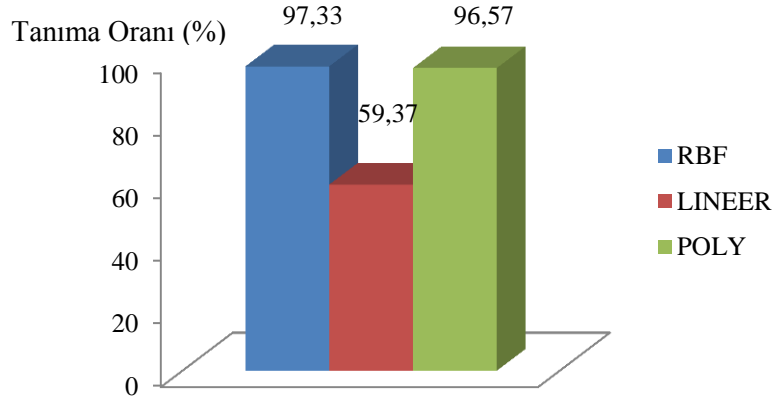
Şekil 4.9. Polinom çekirdek fonksiyonu hata dağılımı

İkinci öznitelik seti, eğitim sınıfı için 60.000 x 256 ve test sınıfı için 10.000 x 256 boyutundaki matrislerden oluşmaktadır. İkinci öznitelik setini oluşturan dalgacık katsayılarını elde ederken ana dalgacık olarak Biorthogonal, Coiflet ve Daubechies aileleri kullanılmış ve en yüksek tanıma oranı Coiflet2 filtresiyle elde edilmiştir. Bu öznitelik seti için SVM üç farklı çekirdek fonksiyonu ile test edilmiştir. Elde edilen başarı oranları ve eğitim-test süreleri Çizelge 4.3 ' te sunulmuştur.

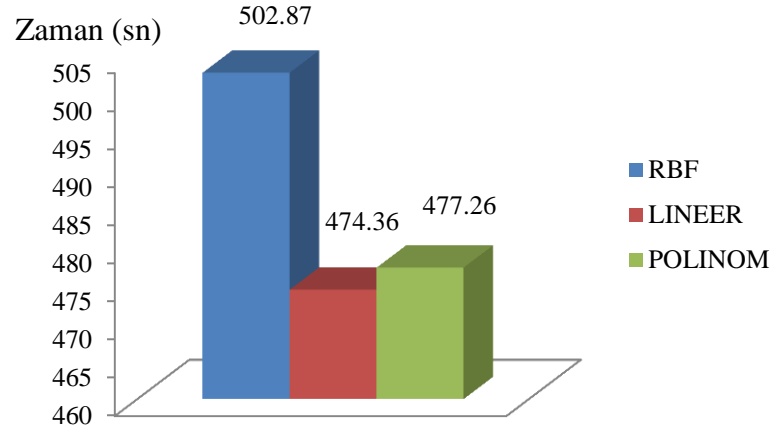
Çizelge 4.3. Coiflet2 dalgacık seti ile elde edilen sonuçlar

Çekirdek Tipi	Eğitim Süresi(sn)		Test Süresi(sn)		Tanıma Oranı(%) (Test)	Tanıma Oranı(%) (Eğitim)
	CPU	Toplam	CPU	Toplam		
Linear	472,35	474,36	262,90	264,35	59,37	59,87
Polinom	475,20	477,26	243,51	244,51	96,57	99,82
RBF	500,84	502,87	258,91	258,56	97,33	100

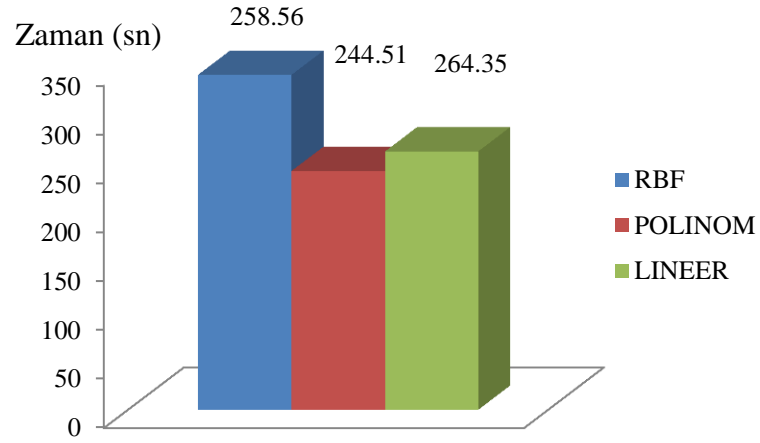
Yapılan uygulamaya ait başarı oranları, eğitim ve test süreleri sırasıyla Şekil 4.10, 4.11 ve 4.12' de görselleştirilmiştir.



Şekil 4.10. Çekirdek fonksiyonlarına göre başarı oranları.

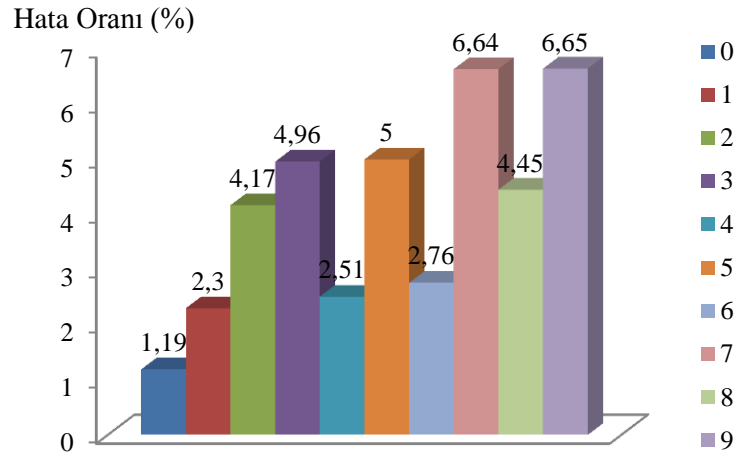


Şekil 4.11. Çekirdek fonksiyonlarına göre eğitim süreleri

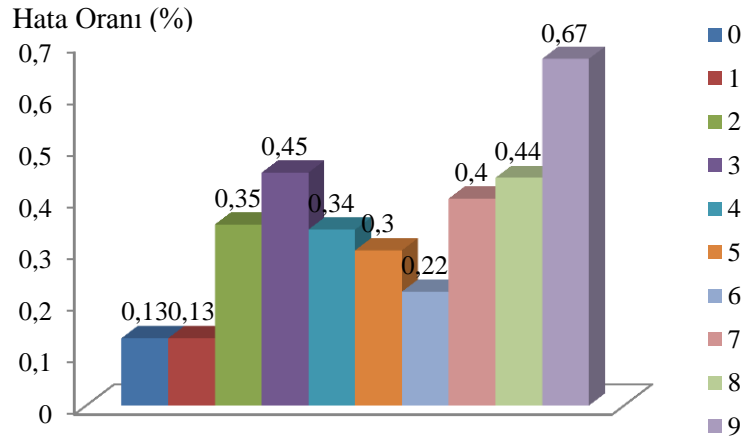


Şekil 4.12. Çekirdek fonksiyonlarına göre test süreleri

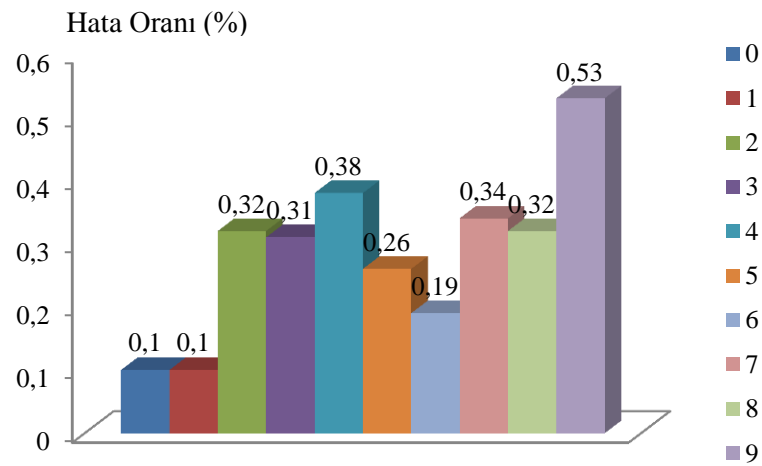
Aşağıda yer alan çizelgelerde ise sınıflandırma aşamasında hatalı tanınan rakam oranları kullanılan çekirdek fonksiyonuna göre ayrı ayrı gösterilmektedir.



Şekil 4.13. Lineer çekirdek hata dağılımı



Şekil 4.14. Polinom çekirdek hata dağılımı



Şekil 4.15. Radyal tabanlı fonksiyon çekirdek hata dağılımı

İkinci öznitelik setini oluşturan dalgacık ailelerine ait filtrelerin tanıma oranına etkileri Çizelge 4.4 'te sunulmuştur.

Çizelge 4.4. Dalgacık filtrelerinin tanıma oranına etkileri

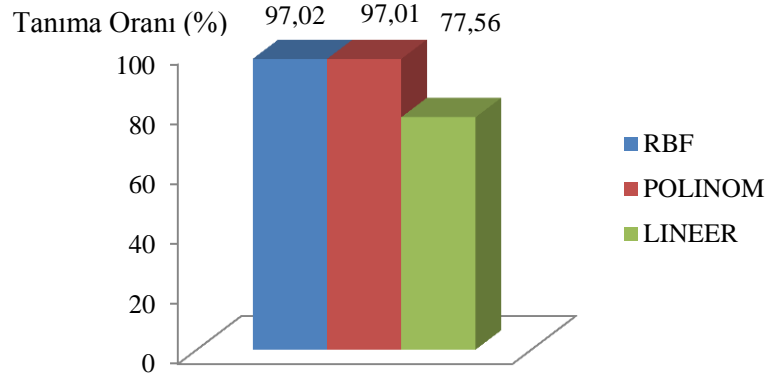
DALGACIK	RBF	LİNEER	POLİNOM
BIOR13	94.58	55.80	94.13
BIOR15	95.20	53.76	94.50
BIOR22	96.87	60.44	96.26
BIOR24	96.98	64.22	96.47
BIOR26	96.93	64.94	96.25
BIOR28	96.92	63.45	96.34
BIOR46	71.07	15.88	23.77
BIOR48	68.49	15.96	23.40
BIOR53	96.87	60.44	92.26
BIOR97	96.98	64.22	96.47
COIF1	96.94	62.60	96.44
COIF2	97.33	59.37	96.57
COIF3	96.97	62.32	96.57
COIF4	96.96	58.54	96.39
COIF5	96.81	60.27	96.23
DAUB1	90.12	56.00	90.59
DAUB2	96.83	55.27	96.46
DAUB3	96.95	59.39	96.75
DAUB4	96.83	59.93	96.24
DAUB5	96.85	64.17	96.38
DAUB6	96.91	60.14	96.38
DAUB7	96.80	57.94	96.13
DAUB8	95.80	50.93	94.87
DAUB9	95.52	47.43	94.90
DAUB10	95.44	46.42	94.55

İlk ve ikinci öznitelik setinin birleşmesiyle oluşan üçüncü öznitelik seti ise, eğitim sınıfı için 60.000 x 1040 ve test sınıfı için 10.000 x 1040 boyutundaki matrislerden oluşmaktadır. Bu öznitelik seti için destek vektör makinesi üç farklı çekirdek fonksiyonu ile test edilmiştir. Elde edilen başarı oranı ve eğitim-test süreleri Çizelge 4.5'de sunulmuştur.

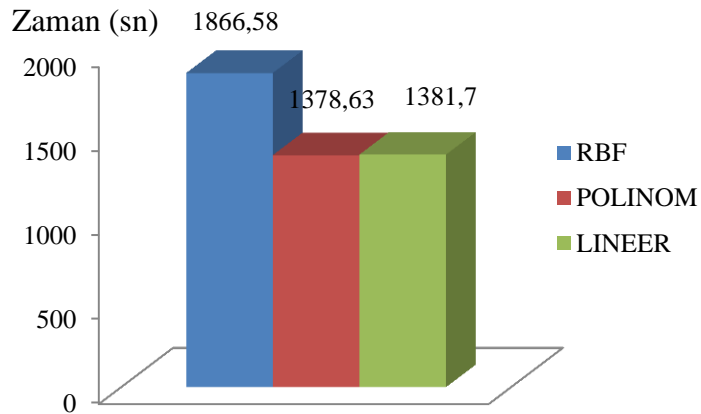
Çizelge 4.5. Dalgacık katsayıları ve piksellere ait yön kodları ile elde edilen sonuçlar

Çekirdek Tipi	Eğitim Süresi(sn)		Test Süresi(sn)		Tanıma Oranı(%) (Test)	Tanıma Oranı(%) (Eğitim)
	CPU	Toplam	CPU	Toplam		
Lineer	1379,17	1381,70	624,15	626,49	77,56	79,17
Polinom	1375,75	1378,63	677,67	678,78	97,01	100
RBF	1865,13	1866,58	998,72	999,45	97,02	100

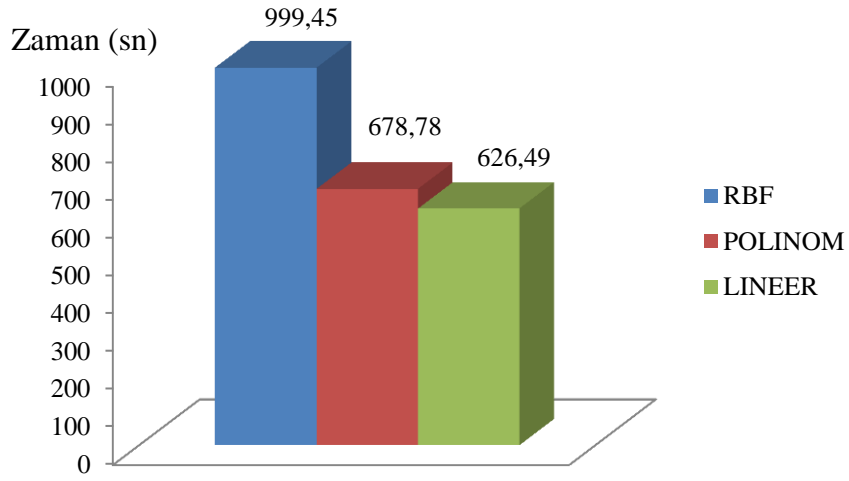
Yapılan uygulamaya ait başarı oranları, eğitim ve test süreleri sırasıyla Şekil 4.16, 4.17 ve 4.18' de görselleştirilmiştir.



Şekil 4.16. Çekirdek fonksiyonlarına göre başarı oranları

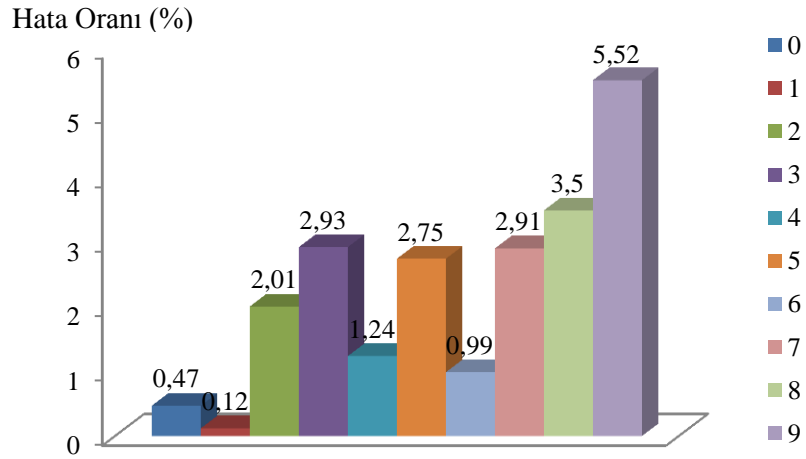


Şekil 4.17. Çekirdek fonksiyonlarına göre eğitim süreleri

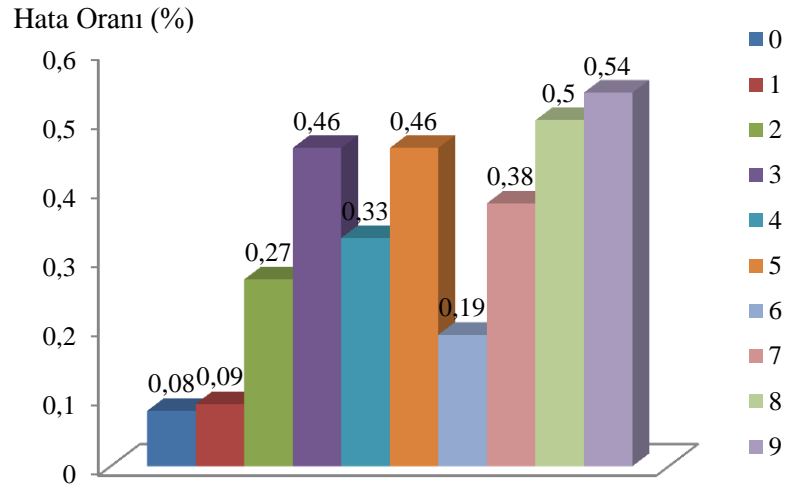


Şekil 4.18. Çekirdek fonksiyonlarına göre test süreleri

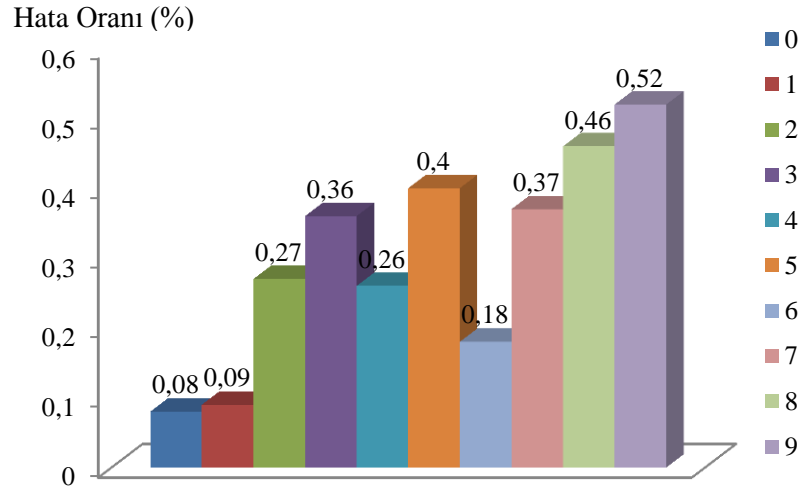
Şekil 4.19, 4.20 ve 4.21’de ise sınıflandırma aşamasında hatalı tanınan rakam oranları, kullanılan çekirdek fonksiyonuna göre ayrı ayrı gösterilmektedir.



Şekil 4.19. Lineer çekirdek hata dağılımı



Şekil 4.20. Polinom çekirdek hata dağılımı



Şekil 4.21. Radyal tabanlı fonksiyon hata dağılımı

Aşağıda yer alan Çizelge 4.6 ise öznitelik setleri ve çekirdek fonksiyonları açısından elde edilen en iyi sonuçları göstermektedir.

Çizelge 4.6. Elde edilen en iyi tanıma oranları

Öznitelik Türü	SVM Çekirdek Fonksiyonu	Başarı oranı(%)	Toplam Süre
Piksel yön kodları	RBF	96,96	742,56
Dalgacık Katsayıları	RBF	97,33	231,77
Piksel yön kodları + Dalgacık katsayıları	RBF	97,02	999,45

4.5 Seri ve Paralel Kodlara Ait Performans Analizleri

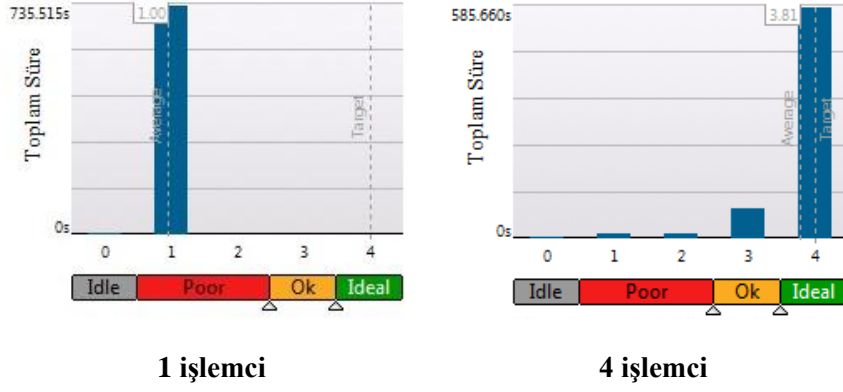
Çalışmanın bu bölümünde projeye ait seri kodlar üç ayrı öznitelik seti için en yüksek tanıma oranlarının elde edildiği parametreler göz önünde bulundurularak Intel Parallel Studio XE 2011 aracı ile analiz edilmiştir. Bu analizler sonucunda çalışmaya ait seri kod blokları içerisinde en fazla zaman kaybedilen bölgeler tespit edilmiştir. Bu çalışmada en fazla süre kaybı, üç öznitelik seti içinde SVM 'ler içerisinde yer alan çekirdek fonksiyonlarında olmuştur. Çalışmada en fazla süre kaybının yaşandığı seri fonksiyonlar ve bu fonksiyonlarda harcanan süreler Çizelge 4.7 'de sunulmuştur.

Çizelge 4.7. Seri kod bloklarının performans sonuçları

Öznitelik Türü	RBF Çekirdek Fonksiyonu(sn)	Toplam Zaman(sn)
I. Öznitelik Seti	724,67	742,56
II. Öznitelik Seti	187,96	231,77
III. Öznitelik Seti	930,69	999,45

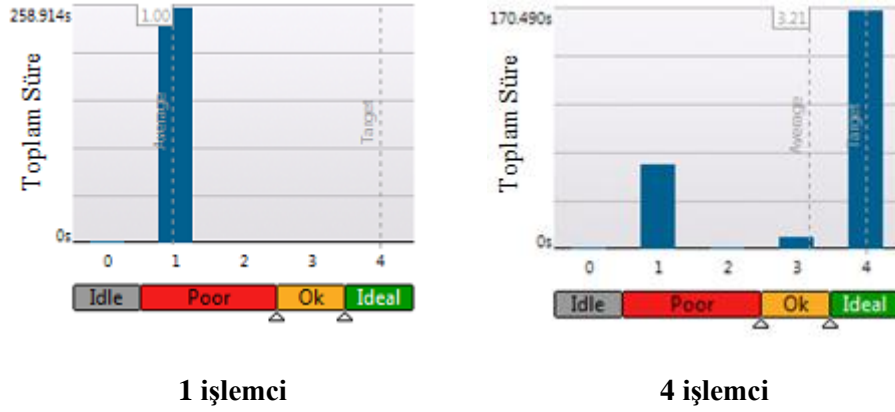
Başta döngüler olmak üzere paralel çalışmaya uygun olan kod blokları OpenMP kütüphanesi kullanılarak tekrar yazılmıştır. Paralel kodlar yazılırken, yapılan işler aynı fakat veri blokları farklı olduğu için veri ayrışımı ("domain decomposition") yöntemi izlenmiştir.

Sobel tabanlı gradient yöntemi ile elde edilen öznitelik setinin RBF çekirdek fonksiyonuna sahip bir SVM ile sınıflandırılmasında kullanılan kod bloklarına ait seri (1 işlemci) ve paralel (4 işlemci) performans analizleri Şekil 4.22 'de gösterilmiştir.



Şekil 4.22 İlk öznelik seti için seri ve paralel kod analizleri

2 boyutlu dalgacık dönüşüm yöntemi ile elde edilen öznelik setinin RBF çekirdek fonksiyonuna sahip bir SVM ile sınıflandırılmasında kullanılan kod bloklarına ait seri ve paralel performans analizleri sırasıyla Şekil 4.23'te gösterilmiştir.

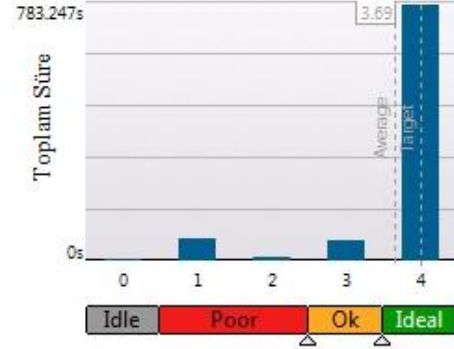


Şekil 4.23. İkinci öznelik seti için seri ve paralel kod analizleri

2 boyutlu dalgacık dönüşüm yöntemi ve sobel tabanlı gradient yöntemi ile elde edilen özneliklerin birleştirilmesiyle oluşan üçüncü öznelik seti de RBF çekirdek fonksiyonuna sahip bir SVM ile sınıflandırılmıştır. İlgili kod bloklarına ait seri ve paralel performans analizleri sırasıyla Şekil 4.24 'te gösterilmiştir.



1 işlemci



4 işlemci

Şekil 4.24. Üçüncü öznitelik seti için seri ve paralel kod analizleri

5. SONUÇ

Bu tez çalışmasında, el yazısı rakamlardan oluşan MNIST veri tabanı kullanılarak hızlı ve tanıma oranı yüksek bir rakam tanıma sisteminin geliştirilmesi amaçlanmıştır. Gerçekleştirilen çalışma üç aşamadan oluşmaktadır. İlk aşamada veri tabanı içerisinde yer alan eğitim ve test karakterleri ön işlemlerden geçirilmiştir. İkinci aşamada her bir karakter için dalgacık yöntemi ve gradient yöntemi kullanılarak karakterlere ait öznitelikler elde edilmiştir. Son aşamada ise elde edilen öznitelikler lineer, polinom ve radyal tabanlı çekirdek fonksiyonlarına sahip SVM'ler ile sistemin eğitimi ve sınıflandırılması amacıyla kullanılmıştır.

Öznitelik çıkarma aşamasında iki farklı yöntem kullanılmıştır. İlkinde son zamanlarda oldukça popüler olan iki boyutlu dalgacık dönüşüm yöntemi tercih edilmiştir. Dalgacık dönüşüm yönteminin kullanılma nedeni, bir sinyale ait bölgesel analizlerin yapılabilmesi ve bir sinyalin alçak frekans ve yüksek frekans bilgilerinin elde edilebilmesidir. Yani bir görüntü üzerindeki genel ve detay bilgilerin elde edilebilmesidir. Bu özellikte görüntüyü temsil eden özniteliklerin kalitesini artırmaktadır. Yapılan çalışmada 2 boyutlu dalgacık katsayılarının elde edilmesi için Biorthogonal, Coiflet, Haar ve Daubechies filtre aileleri kullanılmış ve bu katsayıların sınıflandırma performanslarına etkileri karşılaştırılmıştır. Bu öznitelik setinde en yüksek tanıma oranında % 97.33 oranında Coiflet2 dalgacığı ile elde edilmiştir.

İkinci öznitelik çıkarma yönteminde karakterlerin yapısal olarak incelendiği Sobel tabanlı Gradient yöntemi tercih edilmiştir. Bu yöntemde her bir karakterin kenar piksellerinin eğimleri bilinen 4'lü ve 8'li kodlamaların dışında 12 farklı yönde kodlanmıştır. Bu yöntemin 4'lü kenar kodlamaya göre üstünlüğü yatay ve dikey pikseller dışında çapraz piksellerinde incelenmesi iken 8'li kodlamaya göre üstünlüğü kenar piksellerinin daha detaylı olarak ele alınmasıdır. Bu öznitelik seti ile yapılan testlerde % 96.96 oranında bir tanıma elde edilmiştir.

Birinci ve ikinci öznitelik setlerinden elde edilen tanıma oranları karşılaştırılacak olursa, dalgacık dönüşümü ile elde edilen özniteliklerin tanıma oranının daha yüksek olduğu saptanmıştır. Bu sonuç son yıllarda oldukça popüler olan dalgacık dönüşüm

yönteminin geleneksel olarak kullanılan birçok öznitelik çıkarma yönteminden üstün sonuçlar verdiğinin bir göstergesidir.

Öznitelik çıkarma aşamasında üçüncü öznitelik seti, dalgacık ve gradient yöntemleri ile elde edilen özniteliklerin birleşmesinden oluşmaktadır. Bu öznitelik setinde ise % 97.02 tanıma oranı elde edilmiştir.

Bu çalışmanın seri kodları üzerinde yapılan analizler sonucunda ise en fazla sürenin SVM 'ler içerisinde yer alan çekirdek fonksiyonlarda harcandığı gözlemlenmiştir. Belirlenen fonksiyonlarda döngüler başta olmak üzere paralellik analizi yapılmıştır. Çalışmanın paralelleştirilebilen kısımları belirlenerek seri kodlar mümkün olduğunca küçük parçalara bölünmüştür. Daha sonra veri ayrışımı ("domain decomposition") yöntemi referans alınarak seri kodlar OpenMP kütüphanesi kullanılarak paralelleştirilmiştir. Yapılan testler sonucunda en fazla 1.5 katlık bir hızlanma elde edilmiştir. Paralel programlamada performans artışını sağlayan temel etken paralel olarak yürütülen kod parçalarının seri olarak yürütülenlere oranla daha fazla olmasıdır. Çizelge 4.7'de görüldüğü üzere en fazla zaman kaybı çekirdek fonksiyonlarında yaşanmıştır. Bu çalışmada OpenCV 2.0 içerisinde yer alan SVM sınıfı kullanılmıştır. SVM içerisinde seri olarak çalışan kısımlar paralel kodlara nazaran daha fazladır ve bu durumda işlemci sayısının artırılması ile hızlanma azda olsa belirli bir oranda artabilir ancak asıl artış SVM içerisindeki paralel çalışan kod bloklarının artmasıyla görünecektir.

Ayrıca bu çalışmanın ilerleyen safhalarında, öznitelik setleri içerisinde en iyi özniteliklerin seçilerek sınıflandırılması ve paralel programlama ile hızlanma için GPU kartlarının kullanılması durumunda çok daha iyi sonuçların elde edilebileceği düşünülmektedir.

6. KAYNAKLAR

- Argonne National Laboratory, 2007. İnternet Sitesi. <http://www.mcs.anl.gov/~itf/dbpp/text/book.html>. Erişim Tarihi: 04.04.2011.
- Awcock, G. J., Thomas, R., 1996. Applied Image Processing. McGraw-Hill Inc, 300p. Singapore.
- Bhattacharya, U., Vajda, S., Mallick, A., Chaudhuri, B.B., Belaid A., 2004. On the Choice of Training Set, Architecture and Combination Rule of Multiple(MLP) Classifiers for Multi resolution Recognition of Handwritten Characters, International Workshop on Frontiers in Handwriting Recognition. Frontiers in Handwriting Recognition International Workshop, pp 419-424.
- Bishop, C. M., 1995. Neural Networks For Pattern Recognition. Clarendon Press, 504p. USA.
- Brown, R.M., Fay, T.H., Walker, C.L., 1988. Handprinted Symbol Recognition System. Pattern Recognition, Vol.21 No.2, pp. 91-118.
- Burges, C.J.C., Schölkopf, B., 1997. Improving the Accuracy and Speed of Support Vector Learning Machines. Advances in Neural Information Processing Systems 9. MIT Press, pp.375-381.
- Cai, J.-H., Liu, Z.-Q., 1999. Integration of Structural and Statistical Information for Unconstrained Handwritten Numeral Recognition,. IEEE Transaction Pattern Analysis and Machine Intelligence, Vol. 21 No.3, pp.263-270.
- Cetişli, B.,2005. El Yazısı Karakter Tanıma: Dalgacık Moment Özniteliklerinin Yenilenen ANFIS ile Sınıflandırılması. Osmangazi Üniversitesi Fen Bilimleri Enstitüsü, Doktora Tezi, 156s, Eskişehir.
- Demirci, D.A., 2007. Destek Vektör Makineleri İle Karakter Tanıma. Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, 43s, İstanbul.
- Dong, J. -X., Krzyzak, A.C., Suen, Y., 2005. Fast SVM Training Algorithm with Decomposition on Very Large Data Sets. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 27 No.4, pp. 603-618.
- Elmas, Çetin., 2003. Yapay Sinir Ağları (Kuram, Mimari, Eğitim, Uygulama), Seçkin Yayıncılık, 190s. Ankara.
- Eray, Osman., 2008. Destek Vektör Makineleri ile Ses Tanıma Uygulaması. Pamukkale Üniversitesi Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, 90s, Denizli.

- Filatov, A., Nikitin, A., Volgunin, A., Zelinsky, P., 1999. The AddressScript™ Recognition System for Handwritten Envelops. Document Analysis Systems: Theory and Practice, pp.157-171.
- Franke, J., Lam, L., Legault, R., Nadal, C., Suen, C.Y., 1993. Experiments with the CENPARMI Database Combining Different Classification Approaches. Proceedings of the Third International Workshop on Frontiers of Handwriting Recognition, pp. 305-311.
- Franke, J., 1997. Isolated Handprinted Digit Recognition. Handbook of Character Recognition and Document Image Analysis, pp. 103-121.
- Gader, P.D., Khabou, M.A., 1996. Automatic Feature Generation for Handwritten Digit Recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 18 No.12, pp. 1256-1261.
- Gao, J., Ding, X.Q., 2000. On Improvement of Feature Extraction Algorithms for Discriminative Pattern Classification. 15th International Conference on Pattern Recognition, Vol.2 No.1, pp.101-104, Spain.
- Gose, E., Johnsonbaugh, R., Jost, S., 1996. Pattern Recognition and Image Analysis. Prentice Hall, 483p. New Jersey.
- Güler, İ., Übeyli, E.D., 2004. Dalgacık Dönüşümünün Kullanımı ile Teşhis Sistemleri için Öznitelik Çıkarma: İç Karotid Atardamar Doppler İşaretlerinin Durum Analizi. Akıllı Sistemlerde Yenilikler ve Uygulamaları Sempozyumu, s 23-26.
- Graps, A., 1995. An Introduction to Wavelets. IEEE Computational Science & Engineering, Vol. 2 No. 2, pp.50-61.
- Ha, T., Bunke, H., 1997. Off-line Handwritten Numeral Recognition by Perturbation Method. IEEE Transaction Pattern Analysis and Machine Intelligence, Vol. 19 No. 5, pp. 535-539.
- Hazar, Ş., 2007. Yapay Zeka Uygulamaları: Yüz Tanıma. Beykent Üniversitesi Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, 92s, İstanbul.
- Huang, Y. S and Suen, C. Y., 1995. A Method of Combining Multiple Experts for the Recognition of Unconstrained Handwritten Numerals. On Combining Classifiers. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.17 No.1, pp. 90-94.
- Hwang, Y.-S., Bang, S.-Y., 1997. Recognition of Unconstrained Handwritten Numerals by a Radial Basis Function Neural Network Classifier. Pattern Recognition Letters, Vol. 18 No. 7, pp.657-664.

- Kan, C., Srinath, M.D., 2001. Combined Features of Cubic B-Spline Wavelet Moments and Zernike Moments for Invariant Character Recognition. International Conference on Information Technology: Coding and Computing, pp 511.
- Kan C., Srinath, M.D., 2002. Invariant character recognition with Zernike and orthogonal Fourier-Mellinmoments. Pattern Recognition, Vol.35 No.1, pp:143-154.
- Karasulu, Bahadır., 2006. Paralel Hesaplama Tekniğinin Monte Carlo Simülasyonunda Uygulanması. Maltepe Üniversitesi Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, 106s, İstanbul.
- Keerthi, S. S., Shevade, S. K., Bhattachayya, C., Murth, K.R.K., 2001. Improvements to Platt's Smo Algorithm for Svm Classifier Design. Neural Computation, Vol.13 No.3, pp. 637-649.
- Khotanzad, A., Hong, Y.H., 1990. Invariant Image Recognition by Zernike Moments, IEEE PAMI, Vol.12 No.5, pp:489-497.
- Kittler, J., Hatef M., Duin, R. P. W., Matas, J., 1998. On Combining Classifiers. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.20, No. 3, pp.226-239.
- Lawrence Livermore National Laboratory, 2007. İnternet Sitesi. https://computing.llnl.gov/tutorials/parallel_comp/. Erişim Tarihi: 30.03.2011.
- LeCun, Y., Jackel, L., Bottou, L., Brunot, A., Cortes, C., Denker, J., Drucker, H., Guyon, I., Muller, U., Sackinger, E., Simard, P., Vapnik, V., 1995. Comparison of Learning Algorithms for Handwritten Digit Recognition. Proceedings of the International Conference on Artificial Neural Networks, pp. 53-60.
- LeCun, Y., Bottou, L., Bengio, Y., Haffner, P., 1998. Gradient-Based Learning Applied to Document Recognition. Proceedings of the IEEE, Vol. 86 No. 11, pp 2278-2324.
- LeCun, Y., MNIST OCR Data, 2004. <http://yann.lecun.com/exdb/mnist/>. Erişim Tarihi: 30.03.2011
- Lee, D.-S., Srihari, S.N., 1993. Handprinted Digit Recognition: A Comparison of Algorithms. Proceedings of the Third International Workshop on Frontiers of Handwriting Recognition, pp. 153-164.
- Lee, S. W., 1996. Off-Line Recognition of Totally Unconstrained Handwritten Numerals Using Multilayer Cluster Neural Network. IEEE Transactions on PAMI, Vol.18 No.6, pp. 648-652.

- Lee, S. W., 1995. Multilayer Cluster Neural Network for Totally Unconstrained Handwritten Numeral Recognition. *Neural Networks*, Vol.8 No.5, pp. 783-792.
- Lim, J.S., 1990. *Two dimensional signal and image processing*, Prentice Hall, 694p. USA.
- Liu, C. L., Nakashima, K., Sako, H., Fujisawa, H., 2002. Handwritten Digit Recognition Using State-of-the-art Techniques. *Pattern Recognition*, Vol.36 No.10, pp.2271-2285.
- Liu, C.-L., Nakagawa, M., 1999. Handwritten Numeral Recognition Using Neural Networks: Improving the Accuracy by Discriminative Training. *Proceedings of the Fifth International Conference on Document Analysis and Recognition*, pp. 257-260.
- C.-L. Liu, Y.-J. Liu, R.-W. Dai, 1997. Preprocessing and Statistical/Structural Feature Extraction for Handwritten Numeral Recognition. *Progress of Handwriting Recognition*, pp. 161-168.
- Mowlaei, A., Faez, K., Haghghat, A.T., 2002. Feature Extraction with Wavelet Transform for Recognition of Isolated Handwritten Farsi/Arabic Characters and Numerals. *International Conference on Digital Signal Processing*, pp. 923 – 926.
- Misiti, M., Misiti, Y., Oppenheim, G., Poggi, J.M., 1996. *Wavelet Toolbox for Use with Matlab*.
- NIST(National Institute of Standarts and Technology), 2002. İnternet Sitesi. <http://www.nist.gov/srd>. Erişim Tarihi: 30.03.2011
- Oh, I.-S., Lee, J.-S., Suen, C.Y., 1999. Analysis of Class Separation and Combination of Class-Dependent Features for Handwriting Recognition. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, Vol.21 No. 10, pp. 1089-1094.
- Oliveira, L. S., Sabourin, R., Bortolozzi, F., Suen, C.Y., 2002. Impacts of Verification on A Numeral String Recognition System. *Pattern Recognition Letters*, Vol. 24 No.7, pp. 1023-1031.
- Özkan, Yalçın., 2008. *Veri Madenciliği Yöntemleri*, Papatya Yayıncılık Eğitim, 215s, İstanbul.
- Pitas, I., 1999. *Digital Image Processing Algorithms and Applications*. John Wiley & Sons Inc, 360p. U.S.A.
- Saraç, T., 2004. *Yapay Sinir Ağları*. Gazi Üniversitesi Endüstri Mühendisliği Anabilim Dalı, Seminer Projesi, 71s. Ankara.

- Schalkoff, R. J., 1989. Digital Image Processing and Computer Vision, John Wiley & Sons Inc, 489p. U.S.A.
- Schalkoff, R. J., 1992. Pattern Recognition: Statistical, Structural And Neural Approaches, John Wiley & Sons Inc, 384p. U.S.A.
- Shen, D., Ip, H.H.S., 1999. Discriminative wavelet shape descriptors for recognition of 2-D patterns. Pattern Recognition, Vol.32 No.2, pp:151-165.
- Shi, M., Fujisawa, Y., Wakabayashi, T., Kimura, F., 2002. Handwritten Numeral Recognition Using Gradient and Curvature of Gray Scale Image. Pattern Recognition, Vol.35 No.10, pp. 2051-2059.
- Suen, C.Y., Nadal, C., Legault, R., Mai, T.A., Lam, L., 1992. Computer Recognition of Unconstrained Handwritten Numerals. Proceedings of the IEEE, Vol. 80 No. 7, pp. 1162-1180.
- Suen, C.Y., Liu, K., Strathy, N.W., 1999. Sorting and Recognizing Cheques and Financial Documents. Document Analysis Systems: Theory and Practice, pp. 173-187.
- Singh, D., Singh, S.K., Dutta, M., 2010. Hand Written Character Recognition Using Twelve Directional Feature Input and Neural Network. International Journal of Computer Applications, Vol.1 No.3, pp. 78-103.
- Teow, L.N., Loe, K.F., 2002. Robust Vision-Based Feature and Classification Schemes for Off-Line Handwritten Digit Recognition. Pattern Recognition, Vol. 35 No.1, pp. 2355-2364.
- Tepe, C., Sezgin, H., 2009. EKG Sinyalinde Gürültü Gidermede Ayrık Dalgacık ve Paket Dalgacık Yöntemlerinin Karşılaştırılması. Elektrik – Elektronik – Bilgisayar ve Biyomedikal Müh. 13. Ulusal Kongresi, Ankara.
- Tou, J.T., Gonzalez, R.C., 1977. Pattern Recognition Principles. Massachusetts, Addison-Wesley P.C, 377p. U.S.A.
- Trier, Ø. D., Jain, A.K., Taxt, T., 1996. Feature Extraction Methods for Character Recognition-A Survey. Pattern Recognition, Vol.29 No. 4, pp. 641-662.
- Theodoridis, S., Koutroumbas, K., 2006. Pattern Recognition. Academic Press, 689p. U.S.A.
- Woods, K., Kegelmeyer, W. P., Bowyer, K., 1997. Combination of Multiple Classifiers Using Local Accuracy Estimates. IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol.19, No.4, pp.405-410.
- Xu, L., Krzyzak, A., Suen, C. Y., 1992. Methods of Combining Multiple Classifiers and Their Applications to Handwriting Recognition. IEEE Transactions on Systems, Man, and Cybernetics, Vol.22, No.3, pp.418-435.

Yang, L. H., Suen, C. Y., Bui, T. D., Zhang, P., 2005. Discrimination of Similar Handwritten Numerals Based on Invariant Curvature Features. Pattern Recognition, Vol.38 No.7, pp. 947-963.

Yavuz, Gözde., 2008. Plaka Tanıma Sistemi. Sakarya Üniversitesi Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, 75s, İstanbul.

Zhang, P., Bui, T.D., Suen, C.Y., 2005. Wavelet Feature Extraction for the Recognition and Verification of Handwritten Numerals. Wavelet Analysis and Active Media Technology, Vol 1, pp.3-9.

ÖZGEÇMİŞ



Adı Soyadı : Ali GÜNEŞ

Doğum Yeri ve Yılı: Mersin- 1986

Medeni Hali : Bekar

Yabancı Dili : İngilizce

Eğitim Durumu (Kurum ve Yıl)

Lise : Atatürk Anadolu Teknik Lisesi (2005)

Lisans : S.D.Ü Teknik Eğitim Fakültesi Bilgisayar Sistemleri Öğretmenliği(2009)

Yüksek Lisans : S.D.Ü Mühendislik Mimarlık Fakültesi Bilgisayar Mühendisliği
A.B.D (2009-)

Çalıştığı Kurum/Kurumlar ve Yıl:

-Süleyman Demirel Üniversitesi M.M.F. Bilgisayar Mühendisliği A.B.D,
Araş.Gör, (2009-Devam Ediyor)

Yayınları (SCI ve diğer makaleler)

1- KÜÇÜKSİLLE, E.U.,YİĞİT,T.,GÜNEŞ,A.,SARIMAN,G., "A system for the detection and reporting of wireless modem signals", Scientific Research and Essays, 5/11,2010,pp. 1346-1350

2- Yigit T.,Günes A.,Okumus Ş.,Orhan M.,"Generating Web Template with Suitable Colors Based on Genetic Algorithm",International Conference on Evolutionary Computation Theory and Applications,24-26 October 2011,Paris-France