T.C.
YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

DESTEK VEKTÖR MAKİNELERİ KULLANARAK GÖMÜLÜ SİSTEM ÜZERİNDE YÜZ TANIMA UYGULAMASI

HİLAL GÜNEREN

YÜKSEK LİSANS TEZİ
ELEKTRONİK VE HABERLEŞME MÜHENDİSLİĞİ ANABİLİM DALI
ELEKTRONİK PROGRAMI

DANIŞMAN
DOÇ. DR. BURCU ERKMEN

İSTANBUL, 2015

T.C.

YILDIZ TEKNİK ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

DESTEK VEKTÖR MAKİNELERİ KULLANARAK GÖMÜLÜ SİSTEM ÜZERİNDE YÜZ TANIMA UYGULAMASI

Hilal GÜNEREN tarafından hazırlanan tez çalışması 20.11.2015 tarihinde aşağıdaki jüri tarafından Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü Elektronik ve Haberleşme Mühendisliği Anabilim Dalı'nda **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Tez Danışmanı

Doç. Dr. Burcu ERKMEN

Yıldız Teknik Üniversitesi

Jüri Üyeleri

Doç. Dr. Burcu ERKMEN

Yıldız Teknik Üniversitesi _____

Doç. Dr. Serkan TOPALOĞLU

Yeditepe Üniversitesi _____

Yrd. Doç. Dr. Revna ACAR VURAL

Yıldız Teknik Üniversitesi _____

ÖNSÖZ

Tez çalışmam boyunca değerli fikirlerinden istifade ettiğim, tez danışmanım Doç. Dr. Burcu Erkmen Hocama teşekkürlerimi bir borç bilirim.

Tez çalışmam süresince benden manevi desteklerini esirgemeyen sevgili arkadaşım Canan GİRGİN ve canım aileme en derin şükranlarımı sunarım.

Kasım, 2015

Hilal GÜNEREN

İÇİNDEKİLER

Sayfa

SİMGE LİSTESİ

A	Normalize edilmiş yüz uzay matrisi
b	DVM hiper düzlemin yönelimi
c	Sınıf sayısı
C	Saçılım (kovaryans) matrisi
K	Kernel fonksiyonu
L	Saçılım (kovaryans) matrisi
L_p	Lagrange fonksiyonu
m	$W_{\text{özyüz}}$ öznitelik matrisinin ortalaması
m_j	$W_{\text{özyüz}}$ öznitelik matrisindeki her sınıfa ait sınıf içi öznitelik matrislerinin
M	Yüz görüntüsü satır uzunluğu
N	x DVM girdi vektörünün uzunluğu
N	Yüz görüntüsü sütun uzunluğu
P	Eğitim setinde yer alan yüz görüntüsü sayısı
P_j	Her bir sınıfa ait eğitim verisi adedi
P'	Seçilen özyüz sayısı ortalaması
$Qm.n$	Sabit noktalı sayı formatı gösterimi
S_w	Sınıf içi saçılım matrisi
S_b	Sınıflar arası saçılım matrisi
T	Yüz uzay matrisi
T_j	Her sınıfa ait yüz uzay matrisi
u	Özvektörler
U	Özyüzler matrisi
$U_{\text{özyüz}}$	Azaltılmış özyüzler matrisi
U_{DAA}	DAA sonu elde edilen özvektörler
U_{fisher}	Fisher yüzler matrisi
v	Özvektörler
w	DVM hiper düzlemin normal vektörü
$W_{\text{özyüz}}$	Özyüz öznitelik vektörlerinden oluşan öznitelik matrisi
W_{fisher}	Fisher yüz öznitelik vektörlerinden oluşan öznitelik matrisi
x	Destek Vektör Makinesi girdi vektörü
x_i	Destek vektörleri
y_i	Sınıf etiketi

α_i	Lagrange çarpanları
λ	Özdeğerler
μ_i	ξ_i değerinin pozitif olmasını sağlayan Lagrange parametresi
ξ_i	DVM eğitim hatası sapması
Γ_i	Matrisden vektöre dönüştürülen yüz görüntüsü vektörü
Γ_{yeni}	Matrisden vektöre dönüştürülen yeni yüz görüntüsü vektörü
$\emptyset(x)$	Yüksek boyut uzayına taşınan x DVM girdi vektörü
Φ_i	Normalize edilmiş yüz görüntüsü vektörü
Φ_{yeni}	Normalize edilmiş yeni yüz görüntüsü vektörü
Ψ	Yüz uzay matrisinin ortalaması
Ψ_j	Her sınıfa ait yüz uzay matrisinin ortalaması
Ω_i	Yüz görüntüsüne ait öznitelik vektörü
Ω_{yeni}	Yeni yüz görüntüsüne ait öznitelik vektörü

KISALTMA LİSTESİ

BBA	Bağımsız Bileşen Analizi
CUDA	Hesaplamalı Tümleşik Aygıt Mimarisi (Compute Unified Device Architecture)
DAA	Doğrusal Ayırtaç Analizi
DVM	Destek Vektör Makineleri
FPGA	Sahada Programlanabilir Kapı Dizileri (Field Programmable Gate Array)
GKM	Gauss Karışım Modeli (Gaussian Mixture Models)
GPU	Grafik İşlem Birimi (Graphics Processing Unit)
k-NN	k En Yakın Komşuluk Yöntemi
ORL	Olivetti Araştırma Laboratuvarı (Olivetti Research Laboratory)
TBA	Temel Bileşen Analizi
VHDL	Çok Yüksek Hızlı Tümleşik Devre Donanım Tanımlama Dili (VHSIC Hardware Description Language)
YSA	Yapay Sinir Ağları

ŞEKİL LİSTESİ

Sayfa

ÇİZELGE LİSTESİ

Sayfa

ÖZET

DESTEK VEKTÖR MAKİNELERİ KULLANARAK GÖMÜLÜ SİSTEM ÜZERİNDE YÜZ TANIMA UYGULAMASI

Hilal GÜNEREN

Elektronik ve Haberleşme Mühendisliği Anabilim Dalı

Yüksek Lisans Tezi

Tez Danışmanı: Doç. Dr. Burcu ERKMEN

Yüz tanıma insanların günlük yaşamlarında zorlanmadan ve sıklıkla gerçekleştirdikleri görevlerden biridir. İnsan yüzü oldukça ayırt edici özelliklere sahiptir ve insan beyni yüze ait görsel bilgileri, biyometrik tanımlayıcılar olarak kullanır.

Bilgisayarlı görü, insanları tanımlamak için yüz görüntülerini kullanarak beynin bu karmaşık fonksiyonunu taklit etmeyi amaçlar. Yüz tanıma problemi, bir yüz görüntüsü veya yüz görüntüleri içeren bir video kaydı girdi olarak verildiğinde, bilinen kişilerin yüz görüntülerini içeren bir veritabanı kullanılarak girdideki bir ya da daha fazla yüz görüntüsünün tanımlanması veya doğrulanması olarak ifade edilebilir.

Sahada Programlanabilir Kapı Dizileri (FPGA); sayısal işaret işleme, biyometrik tanıma, medikal görüntü işleme, uzay ve savunma sistemleri, bilgisayar görüntüsü gibi birçok alanlarında kullanılmaktadır. FPGA programlanabilir mantık elemanlarıdır ve her bir mantık bloğunun işlevi kullanıcı tarafından düzenlenebilmektedir.

Bu tez çalışmasında Destek Vektör Makineleri (DVM) kullanılarak FPGA ve GPU üzerinde yüz tanıma uygulaması gerçekleştirilmiştir.

Öncelikle MATLAB ortamında, yüz görüntülerinden özyüz (eigenface) ve Fisher yüz (Fisherface) yöntemleri ile çeşitli boyutlarda öznitelikler elde edilmiştir. Çıkarılan öznitelikler; DVM, Yapay Sinir Ağları (YSA) ve k En Yakın Komşuluk Yöntemi (k-NN)

kullanılarak eğitilmiş ve sınıflandırma performansları karşılaştırılmıştır. Daha sonra MATLAB ortamında eğitilen DVM sınıflandırıcısı, FPGA ve Grafik İşlem Birimi (GPU) üzerinde gerçekleştirilmiş ve performansları karşılaştırılmıştır.

Gerçeklenen sistem, 40 kişi ve her kişiye ait 10 farklı yüz görüntüsünden oluşan ORL yüz veritabanı ile eğitilmiş ve test edilmiştir. Destek Vektör Makinelerinin eğitimi sırasında her bir kişi için o kişiye ait 4 yüz görüntüsü kullanılmış, kalan 6 yüz görüntüsü ile sınıflandırma testi yapılmıştır. DVM sınıflandırıcısının test başarısı özyüzler kullanıldığında %90 seviyesine ulaşırken, Fisher yüzler kullanıldığında %91 seviyesine ulaşmıştır.

Anahtar Kelimeler: Yüz Tanıma, Özyüzler, Fisher Yüzler, Destek Vektör Makineleri, Yapay Sinir Ağları, k En Yakın Komşu Yöntemi

ABSTRACT

FACE RECOGNITION APPLICATION ON EMBEDDED SYSTEM USING SUPPORT VECTOR MACHINES

Hilal GÜNEREN

Department of Electronics and Communications Engineering

MSc. Thesis

Advisor: Assoc. Prof. Dr. Burcu ERKMEN

Face recognition is one of the ordinary tasks of people; they perform it in their daily lives without difficulty. Human face has distinctive characteristic and human mind uses the visual information, which belongs to face, as biometric descriptor.

Computer vision aims to identify people by using the face images; it imitates the complex function of the brain. Face recognition problem, uses a database, which contains face images of known people. When an input (face image or video that contains face images) is given to the problem; via this database the input can be defined and verified.

FPGA (Field Programmable Gate Array) is frequently used in the fields that are digital signal processing, biometric identification, medical vision processing, space and defense systems, computer vision etc. FPGA is programmable logic units and users can be configured each logic block.

In this thesis; FPGA and GPU based face recognition application is developed by using Support Vector Machines (SVM).

First of all; Eigenface and Fisherface methods are used to extract features via MATLAB. This extracted features are educated by SVM, Artificial Neural Networks and k-Nearest Neighborhood methods and then their classification performances are compared.

The system is educated and tested with ORL database that contains 40 people and each person have 10 different face images. During the education of the SVM 4 face images are used for each person and remaining 6 face images are used for testing. As a result, the success of SVM classifier has been reached 90% when we used eigenfaces and 91% when we used Fisherfaces.

Keywords: Face Recognition, Eigenfaces, Fisherfaces, Support Vector Machines, Artificial Neural Network, k Nearest Neighborhood Algorithm

BÖLÜM 1

GİRİŞ

1.1 Literatür Özeti

Yüz tanıma, girdi olarak verilen bir yüz görüntüsünden, daha önce sisteme kayıtlı olan kişilerin yüz görüntülerini içeren bir veritabanı kullanılarak, bireyin kimliğini belirleme süreci olarak tanımlanmaktadır. Bir yüz tanıma sisteminde, yüzü en iyi temsil eden özniteliklerin elde edilmesi, o yüz tanıma sisteminin performansını önemli ölçüde arttırmaktadır. Yüzü en iyi temsil eden özniteliklerin elde edilmesi ve yüz tanıma probleminin çözümünde; Bütüncül yöntemler, Öznitelik Tabanlı yöntemler ve bu yöntemlerin birlikte kullanıldığı melez yöntemler bulunmaktadır.

Bütüncül yöntemlerde, piksellerinin aldığı değerler ya da bu piksellere uygulanan Gabor filtre çıktıları [1] ile ifade edilen yüz görüntüsünün tamamı girdi olarak kullanılmaktadır. Bu yöntemlerde amaç, yüksek boyutlu olan yüz görüntülerini düşük boyutlarda etkin bir gösterim uzayına dönüştürmektir. Bütüncül yöntemlere ilişkin en temel örnekler Özyüz [2], Fisher yüz [3], Temel Bileşen Analizi (TBA), Doğrusal Ayırtaç Analizi (DAA) [4] ve Bağımsız Bileşen Analizidir (BBA).

Bütüncül yaklaşımlardan Özyüzler yaklaşımı ve Temel Bileşen Analizi ilk olarak 1987 yılında Sirovich ve Kirby [5] tarafından yüzü etkin bir şekilde göstermek için kullanılmıştır. 1991 yılında Turk ve Pentland [2], özyüzler yöntemini ilk kez tam otomatik bir siteme dönüştürerek özyüzlerin hesaplanması ve bir yüz resminin özyüzler kullanılarak sınıflandırılması hakkında detaylı bir çalışma sunmuşlardır. Çalışmalarında yüzden kimlik tanıma problemini iki boyutlu tanıma problemi olarak ele almışlar ve Temel Bileşen Analizi kullanmışlardır.

Bu yaklaşımın bir diğer önemli örneği Fisher yüz tanıma yöntemidir. Fisher yüzler, görüntü üzerinde; iki boyutlu veriyi, daha düşük boyutlu öznitelik vektörüne iz düşürürken sınıflar arası değişimleri arttıran ve sınıf içi değişimleri azaltan Doğrusal Ayraç Analizi (LDA) yöntemini kullanır. Bu çalışma Belhumeur tarafından [3] 1997 yılında gerçekleştirilmiştir. Bu yaklaşımda kişilere ait birden fazla görüntü kullanılmakta, yüz uzayında farklı kişilere ait görüntüler arası ayrım arttırılırken aynı kişiye ait görüntülerin bir arada bulunması sağlanmaktadır.

Öznitelik tabanlı yüz tanıma sistemleri; Geometrik Öznitelik tabanlı [6], Şablon Eşleşmeli yöntem [7], Elastik Demet Grafı Eşleme [8], Yapısal Eşleme [9], Gizli Markov Modeli [10], Dalgacık Dönüşüm tabanlı [11] yöntemler olarak sıralanabilir.

Geometrik Öznitelik tabanlı yüz tanıma yönteminde, yüz üzerinde belirleyici özelliğe sahip olan kaş, ağız burun gibi alanların boyutları, aralarındaki mesafe ve geometrisi veya bu alanların birleştirilmesi sonucu oluşan vektörler incelenmektedir. Yüz tanıma, hesaplamalardan kişiye özel sonuçların elde edilmesi ile sağlanır. Belirleyici öznitelik taşıyan noktaların tespit edilmesi ve bu noktaların birleştirilmesi ile oluşan şekillerin geometrisi ilk olarak 1971 yılında Goldstein, Hormon ve Lesk [12] tarafından hesaplanmıştır. Sonrasında bu fikir 1973 yılında Kanade [13] tarafından geliştirilmiş ve bilgisayar tabanlı çalışan bir yüz tanıma sistemi yapılmıştır.

Geometrik tanımadan farklı olarak Şablon Eşleşmeli yöntem, daha basit bir mantığa sahiptir. Yüz görüntüsü üzerine yerleştirilen standart şablondaki belirli noktalar referans alınarak, yüzdeki belirleyici noktaların konumu bulunur. Eşleme algoritması, eğitim ve test verileri arasındaki benzerlikleri ve farkları hesaplar. Yöntem, 1992 yılında Brunelli ve Poggio [6] tarafından geliştirilmiştir.

Yapısal Eşleme yöntemi yüzdeki organların konumlarının tespit edilmesi şeklindedir [14, 15]. Organların birbirlerine göre konumları, şekilleri, özellikleri, aralarındaki açılar ve mesafeler tanıma işleminde kullanılan belirleyici özniteliklerdir. Bu yönteme ait bir diğer yaklaşım şablon ve geometrik eşleşmeli yüz tanıma yöntemlerinden faydalanılarak yüzdeki organların çizgisel kenar haritalarının (Line Edge Map) çıkarılmasıdır [9]. Yüz görüntüleri arasındaki benzerlikler yüz öznitelik şemaları kullanılarak bulunur.

Elastik Demet Grafı yöntemi ile eşleme yaparken Gabor dalgacık dönüşümü kullanılmaktadır. Yüz görüntülerine, üzerinde düğümler bulunan bir dikdörtgenler ızgarası örtülerek yüz üzerindeki kritik noktalar işaretlenir. Izgara üzerinde yer alan her bir düğüm, jet adı verilen farklı frekansa sahip Gabor dalga katsayıları ile ifade edilir. Bir jet, görüntünün Gabor kernelleri ile konvolüsyonu olarak tanımlanır [16]. İki boyutlu Gabor dalgacıklarının konvolüsyon kümesi bu ızgara üzerindeki her noktada hesaplanır ve işaretlenen noktaları konvolüsyon kümesi ile ifade eden öznitelik vektörü çıkarılır.

Elde edilen öznitelik vektörlerinden yüz görüntüsünü sınıflandırmada yani yüz tanıma işleminde, test verileri öznitelik vektörleri ve eğitim verileri öznitelik vektörleri ile olan benzerlik oranı değerlendirilir. Tanıma için en yüksek benzerliğe bakılır. Bu yöntemlerin yanında, Yapay Sinir Ağları [17] ve sınıflar arası marjini en büyütmeyi amaçlayan Destek Vektör Makineleri [18] gibi yöntemler de tanıma için kullanılmaktadır.

Son birkaç yılda sivil ve askeri uygulamalarda, yüz tanımaya olan ilgi artmakta ve yapılan yüz tanıma uygulamalarında hesaplama sürelerini düşürme arayışına gidilmektedir [19]. Bu problem üzerine literatürde FPGA ve GPU kullanılarak yüz tanıma çalışmaları yer almaktadır.

Endluri, Kathait ve Ray çalışmalarında [20], Temel Bileşen Analizi kullanarak Xilinx Spartan XC3S1400AN üzerinde FPGA tabanlı bir yüz tanıma sistemi geliştirmişlerdir. FPGA, 320 X 240 boyutunda bir yüz görüntüsünü gerçek zamanlı olarak tanıma işlemini sağlamaktadır. Neggazi, Bengherabi, Boulkenafet ve Amira ise çalışmalarında [21] Gauss Karışım Modeli (GKM) kullanarak FPGA tabanlı bir yüz tanıma sistemi geliştirmiş ve Virtex5 LX15 FPGA kartında mikroblaze kullanmadan yapılan tasarımda yüz tanıma süresini 1.5 sn. olarak hesaplamışlardır.

Ouerhani, Jridi ve Alfalou'nun yaptığı çalışmada [19], Nvidia Geforce 8400 GS GPU üzerinde geliştirilen hızlı bir yüz tanıma sistemi tanıtılmış ve 64 X 64 boyutunda bir yüz görüntüsünü tanıma 0.25 ms'de tamamlanmıştır. Yine Alexiadis, Papastergiou ve Hatzigaidas çalışmalarında [22] 2 boyutlu DCT tabanlı bir yüz tanıma sistemi NVIDIA GeForce GTS 250 GPU üzerinde geliştirilmiştirler ve 2.8 ms'de sınıflandırma tamamlanmıştır.

Bu tez kapsamında ise, yüz görüntülerinden elde edilen özyüz ve Fisher yüz öznitelik vektörlerinin sınıflandırılmasında Destek Vektör Makinesi kullanılmıştır. Sınıflandırma problemi, bir kişiye ait birden çok eğitim yüz görüntülerinden elde edilen öznitelik vektörleri birinci grup ve kalan diğer yüz görüntülerine ait öznitelik vektörleri ikinci grup olmak üzere iki sınıflı bir sınıflandırma problemi olarak ele alınmış ve Destek Vektör Makineleri eğitilmiştir. Daha sonra test yüz görüntülerine ait öznitelik vektörlerinin Destek Vektör Makineleri ile elde edilen sınıflandırma başarımı, bir Çok Katmalı Yapay Sinir Ağı sınıflandırıcısı başarımı ve k En Yakın Komşuluk sınıflandırıcısı başarımı ile karşılaştırılmıştır. Tezde son olarak literatürdeki çalışmalardan farklı olarak, yüz tanıma için eğitilen DVM sınıflandırıcısı FPGA ve GPU üzerinde gerçeklenmiş ve performansları karşılaştırılmıştır.

1.2 Tezin Amacı

Bu tez çalışmasında, DVM kullanarak FPGA tabanlı bir yüz tanıma sistemi ve DVM kullanarak GPU tabanlı bir yüz tanıma sistemi gerçeklenip, FPGA ve GPU performanslarının karşılaştırılması amaçlamaktadır.

1.3 Orijinal Katkı

Bu tez çalışmasında literatürdeki çalışmalardan farklı olarak, Destek Vektör Makineleri kullanılarak FPGA ve GPU üzerinde bir yüz tanıma sistemi gerçeklenmiştir. Bu sistemde ORL veritabanına ait yüz görüntülerinden özyüz ve Fisher yüz öznitelik vektörlerinin sınıflandırılması için eğitilen Destek Vektör Makinelerinin sınıflandırma performansı FPGA üzerinde gerçekleştirilen sınıflandırıcı ile test edilmiştir. FPGA üzerinde gerçekleştirilen bu sistem ayrıca GPU ile gerçekleştirilerek performansları karşılaştırılmıştır.

BÖLÜM 2

DESTEK VEKTÖR MAKİNELERİ

Destek Vektör Makineleri, istatistiksel öğrenme teorisine dayalı bir eğitimli sınıflandırma tekniği olup temelleri Cortes ve Vapnik (1995) [23] tarafından geliştirilmiştir. DVM, dağılımı hakkında herhangi bir bilgi varsayımı olmayan eğitim veri setleri üzerinde öğrenme yaparak, yeni veriler üzerinde genelleştirme ve tahmin yapmaya çalışan bir makine öğrenmesi algoritmasıdır. Eğitim setlerindeki girdi değişkenleri ve çıktılar eşlenir. Eşler aracılığıyla, farklı sınıflara ait verileri birbirinden ayıracak karar fonksiyonları elde edilir [24]. Karar fonksiyonu ile yeni veri girdi değişkenleri sınıflandırılır.

Genel olarak DVM'nin çalışma prensibi, iki sınıfa ait verileri birbirinden en uygun şekilde ayıran karar sınırlarının (hiper düzlemlerin) belirlenmesidir [25]. DVM için en temel sınıflandırma problemi, "doğrusal" olarak ayrılabilen "iki sınıflı" bir veri setinin sınıflandırılmasıdır. DVM bu problemin çözümü için, iki sınıf arasındaki ayırımı en iyi yapan ve sınıflar arasındaki sınırın maksimum olduğu en iyi ayırıcı düzlemi belirlemeye çalışır. En iyi ayırıcı düzlem, her bir sınıfa ait verilerin, ayırıcı düzleme olan uzaklıklarını maksimum hale getirir. Hiper düzlemlere en yakın öğrenme verileri iki sınıf arasındaki sınırı belirleyen destek vektörlerini oluşturur [24].

Destek Vektör Makineleri, veri setinin doğrusal olarak ayrılabilme ve doğrusal olarak ayrılamama durumuna göre temel olarak ikiye ayrılmaktadır.

2.1 Doğrusal Destek Vektör Makineleri

Doğrusal Destek Vektör Makineleri kullanılan verilerin doğrusal ayrılabilme ya da ayrılamama durumuna göre incelenecektir.

2.1.1 Doğrusal Ayrılabilme Durumu

Destek Vektör Makineleri iki sınıflı bir sınıflandırma tekniği olup, DVM kullanılarak yapılan sınıflandırma işlemlerinde iki sınıfa ait test verilerinin, eğitim verisi ile elde edilen karar fonksiyonu ($g(x) = sign(f(x))$) ile sınıflandırılması amaçlanır. Bu iki sınıf genellikle $(-1, +1)$ sınıf etiketleri ile gösterilmektedir. Girdi verileri, doğrusal olarak ayrılabildiğinde; verileri ayırabilecek sonsuz sayıdaki ayırıcı düzlem içerisinden karar sınırını en büyük yapacak olan ayırıcı düzlemi seçmeyi hedefler. Eğitim sonucu elde edilen en iyi ayırıcı düzlem kullanılarak test verilerini sınıflandıracak olan karar fonksiyonu belirlenir.

$x\epsilon R^N$, eğitim kümesindeki veriler için öznitelikler vektörü, $y_i\epsilon\{-1, +1\}$ ise sınıf etiketlerini göstermek üzere; pozitif ve negatif etiketli verileri birbirinden ayırabilen çok sayıda ayırıcı düzlem çizilebilir. Burada DVM'nin amacı kendisine en yakın noktalar arasındaki uzaklığı maksimum yapan ayırıcı düzlemi bulmaktır. Bulunan bu ayırıcı düzleme en iyi ayırıcı düzlem ve bu düzleme komşu olan, sınır genişliğini sınırlandıran noktalara ise destek vektörleri adı verilir. Bu ayırıcı düzlem üzerindeki herhangi bir x_i noktası (2.1) eşitliğini sağlamaktadır. Burada w ayırıcı düzlemin normal vektörü, b yönelim değerini göstermektedir.

$$(w.x_i) + b = 0 \tag{2.1}$$

DVM eğitimi için kullanılacak N elemanlı bir veri olarak kabul edilsin;

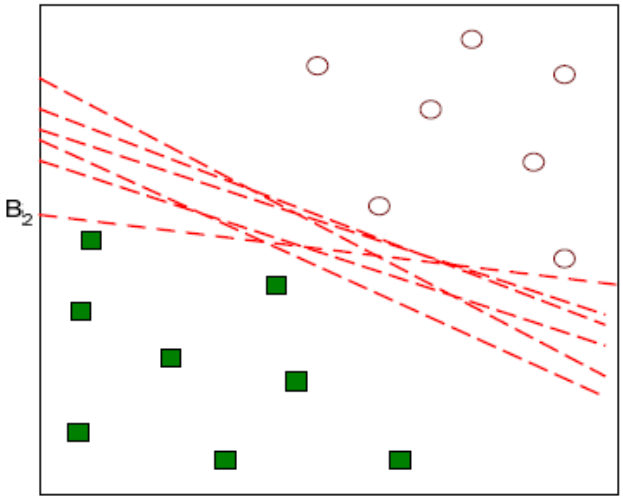
$$E = \{x_i, y_i\}, i = 1,2, \dots, N$$

En iyi karar sınırına ait eşitsizlikler (2.2) ve (2.3) olarak yazılırlar.

$$(w.x_i) + b \geq +1, y_i = +1\ için \tag{2.2}$$

$$(w.x_i) + b \leq -1, y_i = -1\ için \tag{2.3}$$

Şekil 2. 1'de gösterildiği gibi iki sınıflı bir veri setini ikiye ayırabileceğiniz sonsuz sayıda doğru çizilebilir.



Şekil 2. 1 İki sınıflı bir veri setini ikiye ayırabilecek çeşitli doğrular [24]

Amaç, bilinmeyen bir veri seti ile karşılaşıldığında sınıflama hatasını minimum yapacak, yani; farklı sınıftan örnekler arasındaki uzaklığı en büyüten doğruyu seçmektir. Çünkü her iki sınıf verilerinden de mümkün olduğunca uzakta olan karar sınırı, en iyi ayırıcıdır. Büyük sınır, kestirimin eğitim setinde güvenilir olmasını ve yeni örnekler üzerinde kestirim başarımının iyi olmasını sağlar.

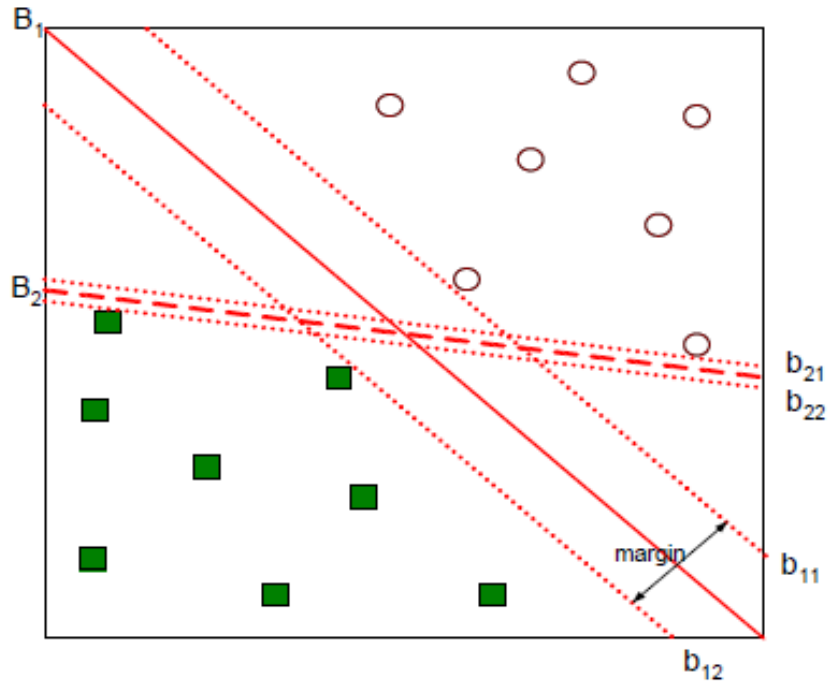
Şekil 2. 2'de kesikli çizgilerle gösterilen ve ayırıcı düzleme eşit uzaklıkta paralel olarak çizilmiş düzlemlere hiper düzlemler denir. Hiper düzlemlere en yakın eğitim verileri destek vektörleri olarak adlandırılır. Hiper düzlemler, eşitlik (2.4) ve (2.5) ile ifade edildiği gibidir. Sınıflandırıcı karar fonksiyonu ise eşitlik (2.6) ile ifade edilir.

$$(w.x_i) + b = \pm 1 \tag{2.4}$$

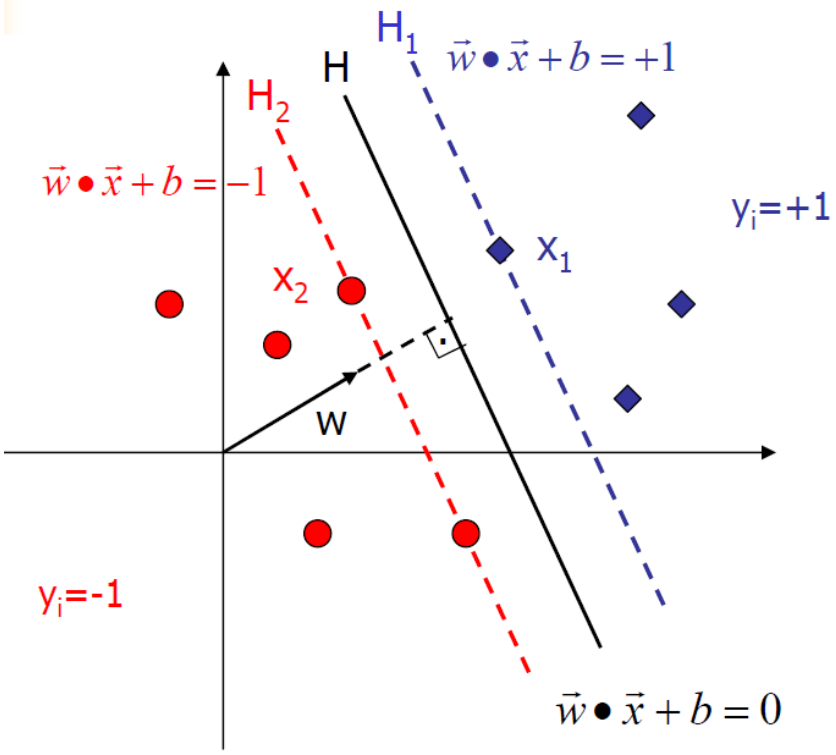
$$y_i\big((w.x_i) + b\big) - 1 \geq 0, y_i \epsilon \{-1, +1\} \tag{2.5}$$

$$f(x) = sign((w.x_i) + b) \tag{2.6}$$

Bu iki hiper düzlem arasındaki uzaklığa ise marjin adı verilmektedir. Marjini en büyük yapan ayırıcı düzlem en iyi ayırıcı düzlemi verir.



Şekil 2. 2 Farklı sınıftan örnekler arasındaki uzaklığı en büyüten ayırıcı düzlem: B1 [24]



Şekil 2. 3 Doğrusal olarak ayrılabilen iki sınıflı veri setleri ve en iyi ayırıcı ve hiper düzlemler [24]

En iyi ayırıcı düzlemin bulunması için uygun w ve b değerleri hesaplanmalıdır. Eşitlik (2.4) ile ifade edilen sınır hiper düzlemlerinin orijine olan uzaklıkları sırasıyla $\frac{|-1-b|}{\|w\|}$ ve $\frac{|1-b|}{\|w\|}$ olarak hesaplanır. Bu iki hiper düzlem arası uzaklık ise $\frac{2}{\|w\|}$ kadardır. En iyi ayırıcı düzlem bulunurken, bu düzlemin sınıra olan uzaklığı maksimuma çıkarılmaya çalışılır. Bunun için ise $\|w\|$ ifadesinin minimum hale getirilmesi gerekmektedir. Bu durumda maksimum sınırın bulunması için $min\left[\frac{1}{2}\ \|w\|^2\right]$ ifadesi (2.5) koşuluna bağlı olarak hesaplanmalıdır [25]. En iyileme problemi eşitlik (2.7)'deki gibidir.

$$min\left[\frac{1}{2}\ \|w\|^2\right]\ koşulu\ ile, y_i\big((w.x_i)+b\big)-1\geq 0, y_i\epsilon\{-1,+1\} \tag{2.7}$$

Söz konusu problem, koşul ve eşitlik ile ifade edilen doğrusal olmayan bir kısıtlı en iyileme (constraint optimzation) problemidir [24] ve bu optimizasyon problemi, Lagrange fonksiyonu ve Lagrange çarpanları $(\alpha_i, i=1,\dots,N)$ kullanılarak çözülebilmektedir. α_i değerleri Lagrange çarpanlarıdır [26], [27].

$$L_p=\frac{1}{2}\|w\|^2-\sum_{i=1}^{N}\alpha_i[y_i(w.x_i+b)-1] \tag{2.8}$$

Eşitlik (2.8)'deki verilen Lagrange denklemi, w ve b değişkenlerine göre en küçüklenir, α_i çarpanlarına göre en büyüklenir. Problem;

$$\underset{\alpha}{\operatorname{argmax}}\ \min_{w,b}\{L_p\} \tag{2.9}$$

olarak yazılır ve en iyileme problemleri ikincil biçimlere dönüştürülebilirler. Bu, Lagrange denkleminin doğru değişkenlerine göre kısmi türevleri alınıp çözülmesi ve sonuçların Lagrange denkleminde yerlerine konarak elenmesi şeklinde yapılır. Sonuç, sadece Lagrange çarpanlarında en büyüklenecek bir bağıntıdır [25].

Birincil Lagrange denkleminin w ve b'ye göre kısmi türevlerinden, (2.10) ve (2.11) eşitliği elde edilir ve (2.12) elde edilir [27].

$$\frac{dL_p}{dw} = 0 => w = \sum_{i=1}^{N} \alpha_i y_i x_i \tag{2.10}$$

$$\frac{dL_p}{db} = 0 => \sum_{i=1}^{N} \alpha_i y_i = 0 \tag{2.11}$$

$$L_d = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j x_i^T x_j \, , \alpha_i \geq 0, \forall i \tag{2.12}$$

(2.12) bir karesel programlama (QP) problemidir ve bu karmaşık problem, Karush-Kuhn-Tucker (KKT) tamamlayıcı koşulu kullanılarak eşitlik (2.13) elde edilir [27].

$$\alpha_i (y_i (w.x_i + b) - 1) = 0, i = 1, \dots, N \tag{2.13}$$

(2.13) eşitliğinden b çekilir. b, eşitlik (2.14)'te verildiği gibidir.

$$b = \ y_i - w.x_i \tag{2.14}$$

Eğitim setinde bulunan her örnek için bir tane Lagrange çarpanı bulunmaktadır. Denklemin çözümü sırasında elde edilen pozitif değerli Lagrange çarpanlarının α_i değerli x_i vektörleri destek vektörlerini oluşturur ve bu destek vektörleri (2.13) eşitliğini sağlayan hiper düzlemler üzerinde yer alır.

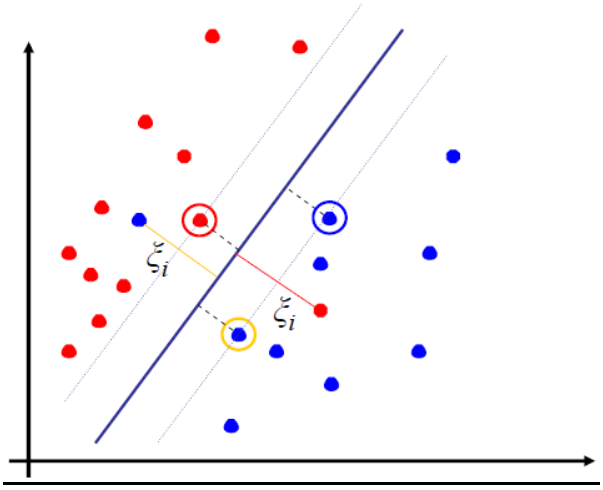
Eşitlik (2.6) ile verilen karar fonksiyonu, hesaplanan x_i destek vektörleri ve α_i ağırlıklarıyla eşitlik (2.10) kullanılarak tekrar yazıldığında;

$$f(x) = sign\big((w.x_i) + b\big) = \ sign\left(\sum_{i=1}^{N} \alpha_i y_i (x.x_i) + b \right) \tag{2.15}$$

karar fonksiyonu (2.15) olarak elde edilir. Yeni bir x verisini test etmek için (2.15) hesaplanır ve x, bu toplam pozitif ise birinci sınıfa, diğer hallerde ikinci sınıfa aittir denir.

2.1.2 Doğrusal Ayrılamama Durumu

Birçok sınıflandırma uygulamasında veri setlerinin doğrusal olarak ayrılması mümkün değildir. Eğitim verilerinin bir kısmının en iyi ayırıcı düzlemin diğer tarafında kalmasından kaynaklanan bu problem bir hata değişkeninin tanımlanması ile çözülmektedir.



Şekil 2. 4 Doğrusal olarak ayrılamaması durumunda iki sınıflı veri setleri ve en iyi ayırıcı hiper düzlemler [24]

En iyi ayırıcı düzlemin hesaplanabilmesi için (2.2) ve (2.3) eşitsizlikleri eğitim hatası sapması değeri ξ_i ile birlikte tekrar yazılarak (2.16), (2.17) ve (2.18) eşitsizlikleri elde edilir.

$$(w.x_i) + b \geq 1 - \xi_i, y_i = +1 \; için \tag{2.16}$$

$$(w.x_i) + b \leq -1 + \xi_i, y_i = -1 \; için \tag{2.17}$$

$$\xi_i \geq 0, \forall i \tag{2.18}$$

Buna göre eşitlik (2.19) gibi yazılabilir.

$$y_i\big((w.x_i) + b\big) - 1 + \xi_i \geq 0, y_i \epsilon \{-1, +1\} \; ve \; \xi_i \geq 0 \tag{2.19}$$

Bu durumda bir x_i örneğinin yanlış sınıflandırılmış olması için $\xi_i < 0$ olmalıdır. Doğru sınıflandırılmış bir x_i veri setinin $0 < \xi_i < 1$ arasında olması bu verinin aslında iki sınır hiper düzlemi arasında yer aldığı anlamına gelmektedir.

Genelleştirilmiş en iyi ayırma düzlemi, $\frac{1}{2}\ \|w\|^2 + C\sum_{i=1}^{N}\xi_i$ fonksiyonunu en küçükleyen w vektörü ile belirlenir. Bu durumda maksimum sınırın bulunması için $min\left[\frac{1}{2}\ \|w\|^2 + C\sum_{i=1}^{N}\xi_i\right]$ ifadesi (2.16) koşuluna bağlı olarak hesaplanmalıdır [25]. Burada C, yanılgı ile sınır arasındaki üst sınır (tradeoff) parametresidir. En iyileme problemi eşitlik (2.20)'deki gibidir.

$$min\left[\frac{1}{2}\ \|w\|^2 + C\sum_{i=1}^{N}\xi_i\right] koşulu\ ile, y_i\big((w.x_i)+b\big) - 1 + \xi_i \geq 0, \quad y_i \epsilon \{-1, +1\} \tag{2.20}$$

C üst sınırı Lagrange çarpanlarının $0 < \xi_i < C$ arasında kalmasını sağlar [25]. Söz konusu koşul ve eşitlik ile ifade edilen doğrusal olmayan bir kısıtlı en iyileme problemi, Lagrange fonksiyonu ve Lagrange çarpanları $(\alpha_i, i = 1, \dots, N)$ kullanılarak çözülebilmektedir. α_i değerleri Lagrange çarpanlarıdır [26], [27].

$$L_p = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N}\xi_i - \sum_{i=1}^{N}\alpha_i[y_i(w.x_i + b) - 1 + \xi_i] - \sum_{i=1}^{N}\mu_i\xi_i \tag{2.21}$$

Bu ifadedeki μ_i değeri, ξ_i değerinin pozitif olmasını sağlayan Lagrange parametresidir.

Eşitlik (2.21)'deki verilen Lagrange denklemi, w, b ve ξ_i değişkenlerine göre en küçüklenir, α_i çarpanlarına göre en büyüklenir. Problem;

$$\underset{\alpha}{\operatorname{argmax}}\ \min_{w,b}\{L_p\} \tag{2.22}$$

olarak yazılır ve en iyileme problemleri ikincil biçimlerine dönüştürülebilirler. Bu, Lagrange denkleminin doğru değişkenlerine göre kısmi türevleri alınıp çözülmesi ve sonuçların Lagrange denkleminde yerlerine konarak elenmesi şeklinde yapılır. Sonuç, sadece Lagrange çarpanlarında en büyüklenecek bir bağıntıdır [25].

Birincil Lagrange denkleminin w, b ve ξ_i'ye göre kısmi türevlerinden, (2.23), (2.24) ve (2.25) eşitliği elde edilir ve (2.26) elde edilir [27].

$$\frac{dL_p}{dw} = 0 => w = \sum_i \alpha_i y_i x_i \tag{2.23}$$

$$\frac{dL_p}{db} = 0 => \sum_i \alpha_i y_i = 0 \tag{2.24}$$

$$\frac{dL_p}{d\xi_i} = 0 => C - \alpha_i - \mu_i = 0 \tag{2.25}$$

$$L_d = \sum_i \alpha_i - \frac{1}{2}\sum_i \sum_j \alpha_i \alpha_j y_i y_j x_i^T x_j, C > \alpha_i \geq 0, \forall i \tag{2.26}$$

Karush-Kuhn-Tucker (KKT) tamamlayıcı koşulu kullanılarak eşitlik (2.27) elde edilir [27].

$$\alpha_i(y_i(w.x_i + b) - 1 + \xi_i) = 0, i = 1, \dots, N \tag{2.27}$$

(2.27) eşitliğinden b çekilir. b, eşitlik (2.28)'de verildiği gibidir.

$$b = y_i(1 - \xi_i) \ - w.x_i \tag{2.28}$$

Eğitim setinde bulunan her örnek için bir tane Lagrange çarpanı bulunmaktadır. Denklemin çözümü sırasında elde edilen pozitif değerli Lagrange çarpanlarının $C > \alpha_i \geq 0, \forall i$ aralığında yer alan α_i değerli x_i vektörleri ise destek vektörlerini oluşturacaktır ve bu destek vektörleri (2.27) eşitliğini sağlayan hiper düzlemler üzerinde yer alır.

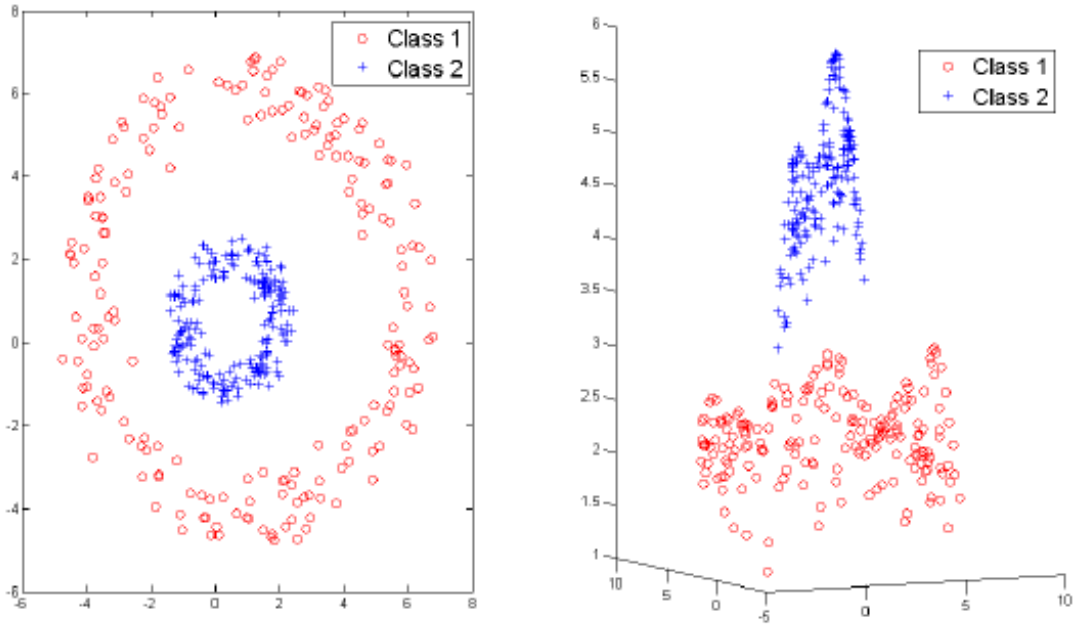
Eşitlik (2.6) ile verilen karar fonksiyonu, hesaplanan x_i destek vektörleri ve α_i ağırlıklarıyla eşitlik (2.23) kullanılarak tekrar yazıldığında;

$$f(x) = sign\big((w.x_i) + b\big) = \ sign\left(\sum_{i=1}^{N} \alpha_i y_i (x.x_i) + b\right) \tag{2.29}$$

karar fonksiyonu (2.29) olarak elde edilir. Yeni bir x verisini test etmek için (2.29) hesaplanır ve x, bu toplam pozitif ise birinci sınıfa, diğer hallerde ikinci sınıfa aittir denir.

2.2 Doğrusal Olmayan Destek Vektör Makineleri

Veriler doğrusal olarak ayrılamadığında, Doğrusal olmayan Destek Vektör Makineleri veriyi, orijinal girdi uzayından daha yüksek boyuttaki bir özellik uzayına aktarır. Bu yeni boyutta veriyi en iyi ayıracak karar sınırını araştırır [24].



Şekil 2. 5 Doğrusal olmayan iki sınıflı veri seti [27]

$$f(x) = sign\left(\sum_{i=1}^{N} \alpha_i y_i \emptyset(x)\emptyset(x_i) + b\right) \tag{2.30}$$

Doğrusal olmayan Destek Vektör Makineleri karar fonksiyonu (2.30)'da verilmiştir. Doğrusal olarak verilerin ayırt edilemediği durumlarda verileri daha yüksek boyutlu bir uzaya taşıyarak çözümlemek için doğrusal olmayan fonksiyonlar kullanılmaktadır. (2.30)'da verilen $\emptyset(x)\emptyset(x_i)$ skaler çarpımının yerine $K(x_i, x_j) = \emptyset(x)\emptyset(x_i)$ şeklinde ifade edilen kernel fonksiyonları kullanılarak söz konusu dönüşümler yapılabilir [23]. Kernel fonksiyonları doğrusal olmayan dönüşümler yapılabilmekte ve verilerin yüksek boyutta doğrusal olarak ayrılabilmesine imkân sağlamaktadır [26].

En çok kullanılan kernel fonksiyonları:

1. Doğrusal fonksiyon: $K(x_i, x_j) = (x_i^T x_j)$
2. Polinomiyal fonksiyon: $K(x_i, x_j) = (x_i x_j)^d$

3. Sigmoid fonksiyon: $K(x_i, x_j) = tanh(kx_i x_j - \delta)$
4. Radyal tabanlı fonksiyon: $K(x_i, x_j) = exp\left(-\gamma \|x_i - x_j\|^2\right),\ \gamma > 0$

2.3 Çok Sınıflı Destek Vektör Makineleri

Destek Vektör Makinelerinde çok sınıf probleminin çözümü için en bilinen 2 yöntem kullanılır.

1. Lagrange fonksiyonun çok sınıfla işlem yapabilecek hale getirilmesidir. Yalnız bu yöntem, sınıf sayısının artması ile hata miktarının artması nedeniyle tercih edilen bir yöntem değildir.
2. Destek Vektör Makinelerinin ikili sınıflar halinde sınıflandırma yapacak biçimde çalışmasını sağlamaktır. Bu yaklaşımda bire karşı bir, bire karşı hepsi ve döngüsel olmayan yaklaşımlar kullanılmaktadır.
    a. Bire karşı bir yaklaşımda her örnek veri seti, diğer örnek veri setleri ile ayrı ayrı eğitilir. Bu yöntemde n adet sınıf için $\frac{\mathrm{n}(\mathrm{n}-1)}{2}$ tane sınıflandırıcı oluşturulur.
    b. Bire karşı hepsi yönteminde, her bir örnek veri seti bir sınıf, geriye kalan sınıfların tamamı bir sınıf olarak kabul edilir ve eğitim bu şekilde yapılır. Bu yöntemde n adet sınıf için n tane sınıflandırıcı oluşturulur.
    c. Döngüsel olmayan graf yöntemi ise bire karşı bir yöntemiyle aynıdır. Eğitim bire karşı bir yöntemiyle yapılır, fakat test aşamasında eğitim örnekleri kontrol edilmek yerine, sınıflandırılacak elemanın ait olmadığı düşünülen sınıflar elenerek işlem yapılır.

Bu tez çalışmasında bire karşı hepsi yöntemiyle, doğrusal kernel fonksiyonlu destek vektör yapısı kullanılmıştır ve kernel nokta çarpımdır (2.31).

$$\mathrm{c} = \sum_i \alpha_i (x_i * x) + b \tag{2.31}$$

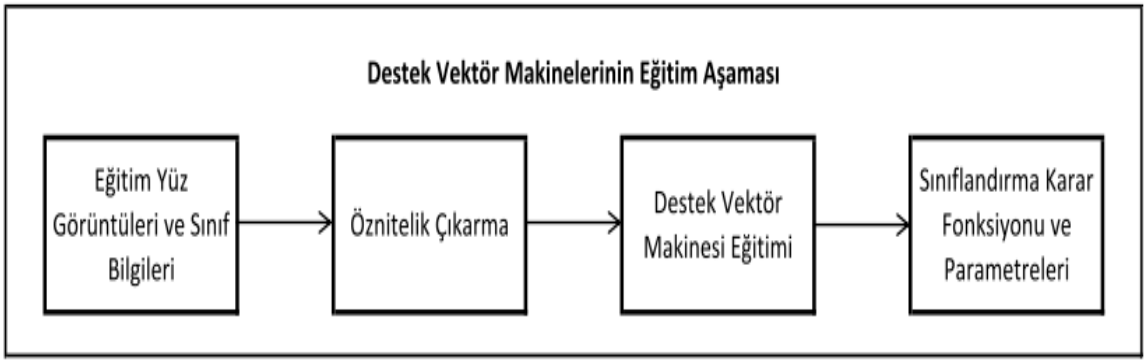
$\mathrm{c} \geq 0$ ise, x giriş vektörü birinci gruba değilse diğer gruplardan birine aittir.

BÖLÜM 3

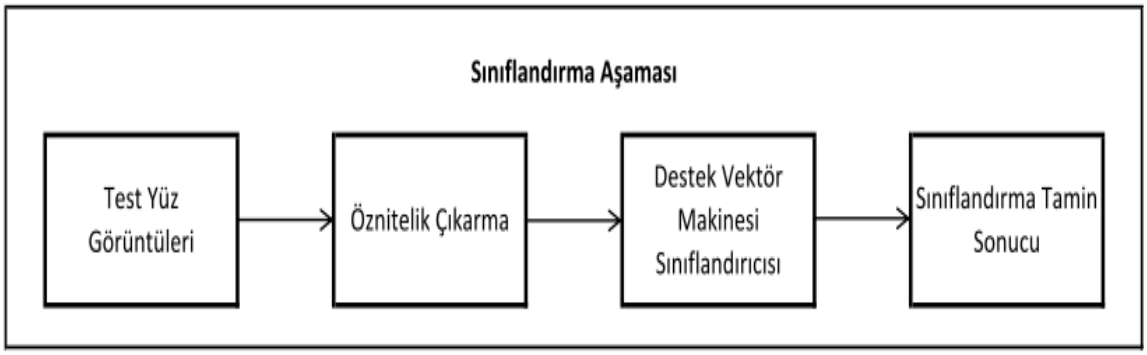
DVM İLE YÜZ TANIMA

Destek Vektör Makineleri, eğitimli sınıflandırma tekniği olup temelleri Cortes ve Vapnik (1995) [23] tarafından geliştirilmiştir. DVM, dağılımı hakkında herhangi bir bilgi varsayımı olmayan eğitim veri setleri üzerinde öğrenme yaparak, yeni veriler üzerinde genelleştirme ve tahmin yapmaya çalışan bir makine öğrenmesi algoritmasıdır. Eğitim setlerindeki girdi değişkenleri ve çıktılar eşlenir. Eşler aracılığıyla, farklı sınıflara ait verileri birbirinden ayıracak karar fonksiyonları elde edilir [24].

Bu tez çalışmasında Destek Vektör Makineleri, bir yüz tanıma sistemi geliştirilmek üzere eğitilmiştir. Genel sınıflandırma yapılarında olduğu gibi, öncelikle yüz görüntülerinin öznitelikleri çıkarılmış ve çıkarılan öznitelikler ait oldukları sınıflarla eşlenerek, sınıflandırma karar fonksiyonlarını elde etmek üzere Destek Vektör Makineleri öğrenimine verilmiştir. Eğitim aşamasında elde edilen karar fonksiyonları ile yeni veri üzerinde sınıflandırma işlemi yapılır. Sistemin eğitim ve sınıflandırma blok şeması sırasıyla Şekil 3. 1 ve Şekil 3. 2 gösterilmektedir.



Şekil 3. 1 Destek Vektör Makinelerinin yüz tanıma sistemi için eğitim aşaması



Şekil 3. 2 DVM parametreleri ve DVM kernel fonksiyonu doğrultusunda elde edilen sınıflandırıcı ile sınıflandırma aşaması

3.1 Öznitelik Çıkarma

Bu tez çalışmasında özellik çıkarma için bütüncül yöntemlerden, özyüz [2], Fisher yüz [3] yaklaşımları kullanılmıştır. Bütüncül yöntemlerde, yüz görüntüsünün tamamı girdi olarak kullanılmaktadır. Bu yöntemlerde amaç, yüksek boyutlu olan girdi yüz görüntülerini düşük boyutlarda etkin bir gösterim uzayına dönüştürmektir.

Bu bölümde özyüz [2], Fisher yüz [3] yaklaşımları anlatılmıştır.

3.1.1 Özyüzler Yöntemi

Temel Bileşen Analizi (TBA), Karhunen-Loeve dönüşümü veya asal bileşen analizi olarak da isimlendirilen örüntü tanıma çalışmalarında özellikle boyut küçültmek amacıyla kullanılan bir alt uzay iz düşüm yöntemidir [2]. TBA birbiriyle korelasyona sahip çok değişkenli veri setlerinin analizi için kullanılır. TBA'nın amacı eldeki veriyi daha az sayıda değişken ile ifade edebilecek en iyi dönüşümü belirlemektir. TBA korelasyonlu veri setlerini temsil eden daha az sayıda korelasyonsuz bileşenler tanımlar ve veri setleri arasındaki temel farklılıkları çıkarır. Dönüşüm sonrasında elde edilen değişkenler, girdi verilerinin temel bileşenleri olarak isimlendirilir. İlk temel bileşen değişinti değeri en büyük olandır ve diğer temel bileşenler değişinti değerleri azalacak şekilde sıralanır. Gürültüye karşı hassasiyet, gereksiz bilgilerin giderilmesi, verinin yorumlanmasını kolaylaştırması ve hafıza ihtiyaçlarını azaltması TBA'nın temel avantajları arasında söylenebilir [28].

TBA, çok sayıda birbiriyle ilişkili değişkenler içeren veri setinin boyutlarını, N- boyutlu uzay yerine, veri içerisindeki farklılıkların mümkün olduğunca korunarak daha az boyuta indirgenmesini sağlayan bir dönüşüm tekniğidir [2]. Aralarında ilişki bulunan N-boyutlu verinin açıkladığı yapıyı, aralarında bağımlılık bulunmayan, orijinal veri boyutundan daha düşük boyutta, orijinal verinin doğrusal bileşenleri ile ifade eder. TBA bu sayede sınıflar arası dağılımı maksimum yapar [28]. Elde edilen temel bileşenler yüz uygulamalarında özyüzler olarak adlandırılırlar [2]. Özyüzler yöntemi ile elde edilen özvektörler, öznitelik vektörlerinin oluşturulması için kullanılır. Bu öznitelik vektörleri daha sonra tanıma işleminde kullanılarak gelen yeni yüz görüntüsünün en çok benzediği eğitim seti elemanı bulunur.

Bir yüz resmi sırasıyla satır ve sütun olmak üzere $(M \times N)$ matris gösterimi ile ifade edilmektedir. Eğitim setinde P tane yüz resmi olduğu varsayılsın. Bir yüz resmini ifade eden $(M \times N)$ matrisi $(M * N) \times 1$'lik vektör biçimine dönüştürülür. Aynı işlemler diğer yüz resimlerine uygulanır ve oluşan tüm vektörler yan yana getirilerek $T = (M * N) \times P$ yüz uzay matrisi oluşturulur.

Yüz uzay matrisi;

$$T = (M * N) \times P = [\Gamma_1 \, \Gamma_2 \, \Gamma_3 \dots \, \Gamma_P] \tag{3.1}$$

olarak gösterilir. (3.1)'de Γ_i $(i = 1,2, \dots, P)$ her biri bir yüz vektörünü ve T, P adet yüz ile oluşturulan yüz uzay matrisinini ifade etmektedir. P adet yüz ile oluşturulan yüz uzay matrisinin ortalaması hesaplanır.

$$\Psi = \frac{1}{P}\sum_{i=1}^{P} \Gamma_i \tag{3.2}$$

Oluşturulan her bir yüz vektöründen Ψ ortalama yüz vektörü çıkartılır.

$$\Phi_i = \Gamma_i - \Psi \tag{3.3}$$

$$A = [\Phi_1 \, \Phi_2 \dots \, \Phi_P] \tag{3.4}$$

Böylece Φ_i $(i = 1,2, \dots, P)$ 'lerden A ortalaması çıkartılmış yüz uzay matrisi oluşturulur. A matrisinin transpozesi alınıp kendisi ile çarpılarak C saçılım matrisi hesaplanır.

$$C = AA^T \tag{3.5}$$

C saçılım matrisinin özdeğer ve özvektörleri bulunur.

$$Cu = \lambda u \tag{3.6}$$

(3.6)'da u özvektör ve λ özdeğerleri ifade etmektedir. Ancak C matrisinin boyutu $(M * N) \times (M * N)$'lik bir kare matris büyüklüğündedir. C matrisinin boyutunun büyük olması özdeğer ve özvektör hesaplanırken özellikle gömülü sistemlerde çok fazla zaman kaybına ve hafıza problemlerine neden olmaktadır. Bu durumun üstesinden gelmek için,

$$L = A^T A \tag{3.7}$$

$P \times P$'lik kare matris büyüklüğündeki L matrisinin özdeğer ve özvektörleri hesaplanır. Genellikle yüz tanıma sistemlerinde $(M * N) \gg P$ olması nedeniyle L matrisinin boyutu oldukça küçüktür.

$$Lv = \lambda v \tag{3.8}$$

L matrisinin özdeğerleri ve özvektörleri hesaplanır. L matrisinin özdeğerleri C matrisinin de özdeğerleridir. Ancak C matrisinin özvektörlerini bulmak için (3.8) eşitliğinin, her iki tarafı soldan A matrisi ile çarpılır.

$$ALv = A\lambda v \tag{3.9}$$

$$AA^T Av = \lambda Av \tag{3.10}$$

$$C(Av) = \lambda(Av) \tag{3.11}$$

$$u = Av \tag{3.12}$$

$$Cu = \lambda u \tag{3.13}$$

$$U = [u_1\ u_2 \dots u_P] = [\Phi_1\ \Phi_2 \dots \Phi_P][v_1\ v_2 \dots v_P] \tag{3.14}$$

Sırasıyla işlemler yapıldığında C matrisinin özvektörleri, L matrisinin özvektörlerinin A matrisi ile sağdan çarpılması ile bulunur. Her bir özdeğere ait, bir özvektör bulunmaktadır. Özdeğerler büyükten küçüğe doğru sıralanır. L matrisinin rankı $P-1$'e eşit olduğundan, başka bir deyişle L matrisinin sıfır olmayan uzayı $P-1$'e eşit olduğundan sıralanan özdeğerlerin en küçüğü sıfıra eşittir. Sıfıra eşit olan özdeğer ve buna ait özvektör sıralamadan çıkartılır.

$$\lambda_1 > \lambda_2 > \lambda_3 > \cdots > \lambda_{P-1} \tag{3.15}$$

En büyük özdeğerin özvektörü ilk sütuna gelecek şekilde özvektörler yan yana yazılarak "özyüz" matrisini oluşturur.

$$U = [u_1\ u_2\ \ldots\ u_{P-1}] \tag{3.16}$$

U, özyüz matrisini ifade etmektedir. Özyüz matrisinin boyutunu azaltmak için matrisi oluşturan her özvektörün yüz uzayı içindeki ağırlıklarına bakılır. Bunu saptamak için her özvektöre ait özdeğerin tüm özdeğerlerin toplamına ne kadar katkıda bulunduğuna bakılır.

$$P' = \min_r \left\{ r \left| \frac{\sum_{i=1}^{r} \lambda_i}{\sum_{i=1}^{r} \lambda_i} \right. \right\} \tag{3.17}$$

P adet özdeğerden ne kadar özdeğer seçileceği (3.17)'den faydalanarak hesaplanır. Burada, eşik değer olarak pratikte 0.95 değeri kullanılabilir. 0.95 eşik değerini sağlayacak P' sayıda özdeğer seçilir. Çıkartılan özdeğerlere ait özvektörler de U özyüz matrisinden atılarak $(u_i\ (i = 1,2, \ldots, P'))$ $U_{\text{özyüz}}$ özyüz matrisi yeniden oluşturulur.

Başlangıçtaki yüz resmini ifade eden her bir yüz vektörü $\Gamma_i\ (i = 1,2, \ldots, P)$ sırasıyla ortalama yüz vektöründen çıkartılır. Daha sonra, özyüz matrisinin transpozesi ile çarpılıp özyüz uzayına yansıtılarak her bir yüzün öznitelik vektörü (3.18) eşitliği ile çıkartılır.

$$\Omega_i = U_{\text{özyüz}}{}^T(\Gamma_i - \Psi) = U_{\text{özyüz}}{}^T(\Phi_i)\ ;\ i = 1,2, \ldots, P \tag{3.18}$$

$$W_{\text{özyüz}} = [\Omega_1\ \Omega_2\ \ldots\ \Omega_P] \tag{3.19}$$

Ω_i $(i = 1,2, \dots, P)$ her biri bir yüzün öznitelik vektörünü, $W_{\text{özyüz}}$ ise özniteliklerin oluşturduğu matrisi göstermektedir.

Herhangi bir yüz resminin hangi yüz sınıfına ait olduğunu bulmak için, yeni yüz resmi ilk önce $(M * N) \times 1$'lik vektör haline getirilerek normalleştirilir. Daha sonra, ortalama yüz vektöründen çıkartılarak özyüz matrisinin transpozesi ile soldan çarpılır. Böylece yeni yüz resmi, özyüz uzayına yansıtılmış olur ve öznitelik vektörü hesaplanır.

$$\Phi_{yeni} = \Gamma_{yeni} - \Psi \tag{3.20}$$

$$\Omega_{yeni} = {U_{\text{özyüz}}}^T \Phi_{yeni} \tag{3.21}$$

Γ_{yeni}, yeni yüz resmini, Φ_{yeni} ortalama çıkartılmış yeni yüz resmini ve Ω_{yeni} ise özyüz uzayına yansıtılmış yeni yüz resmine ait öznitelik vektörünü ifade etmektedir.

Özyüz yönteminde saçılıma yol açan tüm etmenler, yani hem sınıflar arası saçılımlar, hem de sınıf içi saçılımlar da hesaba katılmış olur. Oysa sınıf içi farklar, sınıflandırma işlemi için faydasız hatta zararlıdır [28].

3.1.2 Fisher Yüzler Yöntemi

Fisher yüz algoritması 1997 yılında P. Belhumeur [3] tarafından Doğrusal Ayırtaç Analizi (DAA) temel alınarak geliştirilmiştir. DAA veriler arasındaki ayrımı sağlayan yönün bulunmasını hedefler [3].

Fisher yüzler, aynı kişiye ait birden fazla yüz görüntüsünün yer aldığı çok sınıflı bir veritabanı kullanır. Burada görüntü uzayını daha düşük boyutlu öznitelik uzayına iz düşürürken sınıf içi değişimleri azaltan ve sınıflar arası değişimleri arttıran Doğrusal Ayrıştırma Analizi (LDA) yöntemini kullanır [28]. Fisher yüzler, örüntü saçılımını, sınıflandırma için en uygun şekle sokmaya çalışır. Bu yöntem, veri seti içerisinde bulunan farklı sınıflara ait grupların doğrusal ayrılabilirliğini maksimize ederek boyut azaltması yapar. Bu sayede, Fisher yaklaşımında tanımlama başarım oranında artış gözlenmektedir [28].

Eğitim setinde $(M \times N)$ boyutunda P_j örnek içeren c farklı sınıftan oluşan bir yüz eğitim seti olduğu varsayılsın. Böylece yüz eğitim seti toplam $P = P_j * c$ kadar örnekten

oluşur. Bir yüz resmini ifade eden $(M \times N)$ matrisi $(M * N) \times 1$'lik vektör biçimine dönüştürülür. Aynı işlemler diğer yüz resimlerine uygulanarak oluşan tüm vektörler yan yana getirilip, $T = (M * N) \times P$ global yüz uzay matrisi, P_j adet sınıf içi yüz görüntüsü ile oluşturulan c adet $T_j = (M * N) \times P_j$ sınıf içi yüz uzay matrisi oluşturulur.

Yüz uzay matrisleri;

$$T = (M * N) \times P = [\Gamma_1 \, \Gamma_2 \, \Gamma_3 \, \dots \, \Gamma_P] \tag{3.22}$$

$$T_j = (M * N) \times P_j = \left[\Gamma_1 \, \Gamma_2 \, \dots \, \Gamma_{P_j}\right], j = 1,2, \dots, c \; ve \; \Gamma_i \, \epsilon \, sınıf_j \tag{3.23}$$

olarak gösterilir. (3.22)'de Γ_i $(i = 1,2, \dots, P)$ her biri bir yüz vektörünü ve T, P adet yüz ile oluşturulan yüz uzay matrisinini ve (3.23)'de T_j, P_j adet yüz ile oluşturulan sınıf içi yüz uzay matrisinini ifade etmektedir.

Fisher yaklaşımda, ortalama görüntü vektörü hem tüm veritabanı içerisinde global olarak hem de her bir sınıf için ayrı ayrı hesaplanır. Sınıf içi ortalama görüntü vektörü, Ψ_j olarak simgelenmektedir. Tüm veritabanı içerisindeki P adet yüz ile oluşturulan yüz uzay matrisinin ortalaması (3.24) eşitliği ile hesaplanır.

$$\Psi = \frac{1}{P} \sum_{i=1}^{P} \Gamma_i \tag{3.24}$$

Her sınıf için, P_j adet sınıf içi yüz görüntüsü ile oluşturulan yüz uzay matrisinin ortalaması (3.25) eşitliği ile hesaplanır. Burada Ψ_j ve P_j, j. sınıfa ait ortalama vektörü ve örnek sayısını göstermektedir.

$$\Psi_j = \frac{1}{P_j} \sum_{i=1}^{P_j} \Gamma_i \tag{3.25}$$

DAA sınıflar arası saçılımıençoklarken sınıf içi saçılma oranını en küçükleyen U matrisini bulmaya çalışmaktadır. Bu yüzden öncelikle, oluşturulan vektörlerden sınıf içi ve sınıflar arası saçılma matrisleri elde edilir. Sınıf içi saçılma matrisi,

$$S_w = \sum_{i=1}^{c} \sum_{k=1}^{P_j} (\Gamma_k - \Psi_i)\,(\Gamma_k - \Psi_i)^T \tag{3.26}$$

(3.26) eşitliği ile hesaplanır. Sınıflar arası saçılma matrisi ise,

$$S_b = \sum_{i=1}^{c} P_i(\Psi_i - \Psi)\,(\Psi_i - \Psi)^T \tag{3.27}$$

(3.27) eşitliği ile hesaplanır.

Optimum özvektör matrisi U, sınıflar arası saçılım matrisinin determinantının sınıf içi saçılım matris determinantına oranınıençoklayacak şekilde seçilir;

$$U = argmax_u \frac{|U^T S_b U|}{|U^T S_w U|} = [u_1|u_2| \dots |u_d] => (S_b - \lambda S_w)u = 0 \tag{3.28}$$

(3.28) eşitliği genelleştirilmiş özdeğer problemi olarak karşımıza çıkmaktadır ve U matrisinin çözümü genelleştirilmiş özdeğer çözümü (3.29) ile yapılmaktadır.

$$(S_b - \lambda S_w)u = 0 \tag{3.29}$$

Tanımlama için kullanılacak olan U matrisi, S_b ve S_W matrislerinden elde edilen özvektör matrisidir ve "Fisher yüz" matrisi olarak isimlendirilir.

Özyüz yaklaşımına benzer şekilde, başlangıçtaki yüz resmini ifade eden her bir yüz vektörü Γ_i $(i = 1,2, \dots, P)$ sırasıyla ortalama yüz vektöründen çıkartılır. Daha sonra, Fisher yüz matrisinin transpozesi ile çarpılıp, Fisher yüz uzayına yansıtılarak her bir yüzün öznitelik vektörü (3.30) eşitliği ile elde edilir.

$$\Omega_i = U^T(\Gamma_i - \Psi);\ i = 1,2, \dots, P \tag{3.30}$$

$$W = [\Omega_1\ \Omega_2 \dots\ \Omega_N] \tag{3.31}$$

Ω_i $(i = 1,2, \dots, P)$ her biri bir yüzün öznitelik vektörünü ve W ise özniteliklerin oluşturduğu matrisi göstermektedir.

Herhangi bir yüz resminin hangi yüz sınıfına ait olduğunu bulmak için yeni yüz resmi ilk önce $(M * N) \times 1$'lik vektör haline getirilerek normalleştirilir. Daha sonra, ortalama yüz vektöründen çıkartılarak Fisher yüz matrisinin transpozesi ile soldan çarpılır. Böylece yeni yüz resmi, Fisher yüz uzayına yansıtılmış olur.

$$\Phi_{yeni} = \Gamma_{yeni} - \Psi \tag{3.32}$$

$$\Omega_{yeni} = U^T \Phi_{yeni} \tag{3.33}$$

Γ_{yeni}, yeni yüz resmini, Φ_{yeni} ortalama çıkartılmış yeni yüz resmini ve Ω_{yeni} ise Fisher yüz uzayına yansıtılmış yeni yüz resmine ait öznitelik vektörünü ifade etmektedir.

Fisher yaklaşımına ilişkin algoritmanın yüksek boyutlu matrisler üzerinde yürütülmesi, hesaplama karmaşıklığı ve yükü açısından oldukça güç olmaktadır. Toplam görüntü sayısının, görüntü boyutundan düşük olduğu koşulda $\left((M \times N) \gg P\right)$, Fisher yüz uzayının bulunmasından önce, TBA yöntemi ile her bir görüntüye ait özniteliklerin oluşturulup bu değerlerin DAA yöntemine girilmesi gerekmektedir [3]. Bu sayede, Fisher algoritmasına giriş yapılan matris boyutu $(M \times N)$'den P'ye indirgenmektedir.

Fisher yüz uzayının elde edilmesi için özyüz uzayına ait özniteliklerin kullanılması, boyut indirgenmesinin yanı sıra, tüm veritabanının kullanıldığı durumla kıyasla, tanımlamanın başarım performansını da artırmaktadır [29].

TBA yöntemi ile elde edilen özniteliklerin DAA yönteminde kullanılmasıyla, Fisher yaklaşımındaki yeni veri matrisi, (3.34)'te verilen $W_{\text{özyüz}}$ öznitelik matrisidir.

$$W_{\text{özyüz}} = [\Omega_1 \, \Omega_2 \, \dots \, \Omega_P] \tag{3.34}$$

$$U_{\text{özyüz}} = [u_1 \, u_2 \, \dots \, u_{P'}] \tag{3.35}$$

$$\Psi = \frac{1}{P} \sum_{i=1}^{P} \Gamma_i \tag{3.36}$$

Ω_i $(i = 1,2, \dots, P)$ herbiri bir yüzün TBA ile elde edilmiş öznitelik vektörü, $U_{\text{özyüz}}$, özyüz vektörü ve Ψ, tüm veritabanı içerisindeki P adet yüz ile oluşturulan yüz uzay matrisinin ortalamasıdır.

Tüm veritabanı içerisindeki P adet yüze ait öznitelik vektörü ile oluşturulan öznitelik uzay matrisinin ortalaması (3.37) eşitliği ile hesaplanır.

$$m = \frac{1}{P}\sum_{i=1}^{P} \Omega_i \tag{3.37}$$

Her sınıf için, P_j adet sınıf içi yüz görüntülerine ait öznitelik vektörü ile oluşturulan öznitelik uzay matrisinin ortalaması (3.38) eşitliği ile hesaplanır. Burada m_j ve P_j, j. sınıfa ait ortalama vektörü ve örnek sayısını göstermektedir.

$$m_j = \frac{1}{P_j}\sum_{i=1}^{P_j} \Omega_i \tag{3.38}$$

Sınıflar arası saçılımıençoklarken sınıf içi saçılma oranını en küçükleyen U matrisi, sınıf içi ve sınıflar arası saçılma matrisleri elde edilir.

Sınıf içi saçılma matrisi,

$$\mathrm{S}_w = \sum_{i=1}^{c}\sum_{k=1}^{P_j} (\Omega_k - m_i)\,(\Omega_k - m_i)^T \tag{3.39}$$

(3.39) eşitliği ile hesaplanır. Sınıflar arası saçılma matrisi ise,

$$\mathrm{S}_b = \sum_{i=1}^{c} P_i (m_i - m)\,(m_i - m)^T \tag{3.40}$$

(3.40) eşitliği ile hesaplanır.

Genelleştirilmiş özdeğer çözümü yöntemi ile eşitlik (3.29)'un çözülmesi sonucunda elde edilen özvektörler, U_{DAA} matrisinde saklanmaktadır. Elde edilen U_{DAA} özvektörlerinin transpozesi, $U_{\text{özyüz}}$ özvektörleri ile eşitlik (3.41)'de gösterildiği gibi sağdan çarpılarak U_{fisher} özvektörleri elde edilir.

$$U_{fisher} = U_{\text{özyüz}} {U_{DAA}}^T \tag{3.41}$$

Fisher özniteliklerinin elde edilmesi için yine özyüz yaklaşımına benzer şekilde, başlangıçtaki yüz resmini ifade eden her bir yüz vektörü Γ_i $(i = 1,2, \dots, P)$ sırasıyla global ortalama yüz vektöründen çıkartılır. Daha sonra, U_{fisher} özvektörleri (Fisher yüzler) matrisinin transpozesi ile çarpılıp Fisher yüz uzayına yansıtılarak her bir yüzün "öznitelik" vektörü (3.42) eşitliği ile çıkartılır.

$$\Omega_i = {U_{fisher}}^T(\Gamma_i - \Psi);\ i = 1,2, \dots, P \tag{3.42}$$

$$W_{fisher} = [\Omega_1\ \Omega_2 \dots\ \Omega_N] \tag{3.43}$$

Ω_i $(i = 1,2, \dots, P)$ her biri bir yüzün öznitelik vektörünü başka bir ifade ile yüz sınıfını ve W_{fisher} ise özniteliklerin oluşturduğu matrisi göstermektedir.

Herhangi bir yüz resminin hangi yüz sınıfına ait olduğunu bulmak için yeni yüz resmi ilk önce $(M * N) \times 1$'lik vektör haline getirilerek normalleştirilir. Daha sonra, ortalama yüz vektöründen çıkartılarak özyüz matrisinin transpozesi ile soldan çarpılır. Böylece yeni yüz resmi Fisher yüz uzayına yansıtılmış olur.

$$\Phi_{yeni} = \Gamma_{yeni} - \Psi \tag{3.44}$$

$$\Omega_{yeni} = {U_{fisher}}^T \Phi_{yeni} \tag{3.45}$$

Γ_{yeni}, yeni yüz resmini, Φ_{yeni} ortalama çıkartılmış yeni yüz resmini ve Ω_{yeni} ise Fisher yüz uzayına yansıtılmış yeni yüz resmine ait öznitelik vektörünü ifade etmektedir.

Fisher yaklaşımı kullanarak öznitelik uzayının çıkarılabilmesi için, veri sınıflarının algoritma öncesinde belirlenmesi gerekir. Öncelikle bir kişiye ait birden fazla görüntünün veritabanına kaydedilmesi ile sınıflar oluşturulur. Aynı kişiye ait görüntüler yardımıyla sınıf içi saçılma matrisi, farklı kişilere ait görüntüler yardımıyla da sınıflar arası saçılma matrisi elde edilir.

3.2 Destek Vektör Makineleri Eğitimi

Destek Vektör Makineleri, eğitimli sınıflandırma tekniğidir. DVM, dağılımı hakkında herhangi bir bilgi varsayımı olmayan eğitim veri setleri üzerinde öğrenme yaparak, yeni veriler üzerinde genelleştirme ve tahmin yapmaya çalışan bir makine öğrenmesi

algoritmasıdır. Eğitim setlerindeki girdi değişkenleri ve çıktılar eşlenir. Eşler aracılığıyla, farklı sınıflara ait verileri birbirinden ayıracak karar fonksiyonları elde edilir [24].

Bu tez çalışmasında, 40 gruptan oluşan ORL yüz veritabanına [30] ait yüz görüntüleri sınıflandırma için kullanılmaktadır. DVM sınıflandırıcısı iki sınıflı bir sınıflandırıcı olup, bu çok sınıflı problemin çözümünde Destek Vektör Makinelerinin eğitimi, "bire karşı hepsi" yöntemiyle yapılır ve 40 farklı Destek Vektör Makinesi karar fonksiyonu seti elde edilir.

Bire karşı hepsi yöntemi ile Destek Vektör Makinelerinin eğitimleri için eğitim kümesi iki sınıfa ayrılır.

1. Test edilen kişiye ait yüz görüntülerinden elde edilen öznitelik vektörleri $+1$ sınıf etiketiyle eşlenir ve birinci grup olarak Destek Vektör Makinesi eğitimine verilir.
2. Diğer bütün yüz görüntülerinden elde edilen öznitelik vektörleri -1 sınıf etiketiyle eşlenir ve ikinci grup olarak Destek Vektör Makinesi eğitimine verilir.

Destek Vektör Makinesi bu iki sınıfa ait eğitim verileri üzerinde öğrenme yapar ve DVM sınıflandırma karar fonksiyonu hesaplanır. Bu adım eğitim setindeki kişi sayısı kadar tekrarlanır ve her bir grup için farklı karar fonksiyonları hesaplanır.

3.3 Sınıflandırma

"3.2 Destek Vektör Makineleri Eğitimi" başlığında anlatılan adımlarla eğitilen Destek Vektör Makinelerinden elde edilen karar fonksiyonları ile veri üzerinde sınıflandırma işlemi yapılır. Öznitelikleri elde edilen yüz görüntüleri, elde edilen bu sınıflandırma karar fonksiyonları ile sınıflandırma işlemine tabi tutulur ve sınıflandırma sonucu elde edilir.

$$f(x) = sign\left(\sum_{i=1}^{N} \alpha_i y_i k(x, x_i) + b\right), k; kernel\ fonksiyonu \tag{3.1}$$

(3.1) karar fonksiyonu sonucu $+1$ ise, yüz görüntüsünü nitelendiren öznitelik vektörü x, yoklanan kişiye ait, -1 ise yoklanan kişiden farklı bir sınıfa aittir.

x öznitelik vektörünün ait olduğu sınıfın bulunabilmesi için, her farklı sınıf için eğitilen Destek Vektör Makinesi karar fonksiyonu ile sırasıyla, $+1$ sonucu elde edilene kadar, sınıflandırma işlemine tabi tutulur. Hiçbir karar fonksiyonun çıkışında $+1$ sonucu elde edilememesi durumunda sınıflandırıcı, herhangi bir sınıfa aitlik tahmini yapamamıştır.

3.4 Test

Sisteme, kayıtlı yüzlerle karşılaştırılarak sınıflandırılacak yeni bir yüz görüntüsü verildiğinde, Şekil 3. 2 sınıflandırma blok şemasında gösterildiği gibi;

1. Girdi yüz görüntüsünün öznitelikleri elde edilir.
2. Her öznitelik vektörünün, bulunan 40 farklı Destek Vektör Makinesi karar fonksiyonları kullanılarak, kayıtlı her bir kişiye aitliği "3.3 Sınıflandırma" başlığında anlatıldığı gibi tahmin edilir.

BÖLÜM 4

DVM SINIFLANDIRICININ DONANIM ÜZERİNDE GERÇEKLENMESİ

Bu çalışmada MATLAB üzerinde "Statistics and Machine Learning Toolbox - svmtrain" fonksiyonuyla geliştirilen ve eğitilen Destek Vektör Makinelerine ait sınıflandırıcı karar fonksiyonlarının FPGA ve GPU üzerinde gerçeklenmesi ve performanslarının karşılaştırılması hedeflenmiştir.

FPGA ve GPU üzerinde sadece DVM sınıflandırıcısı karar fonksiyonları tasarlanmış olup, öznitelikleri çıkarılan eğitim verilerinin en iyi sonuç elde edilen 30 uzunluklu öznitelik vektörleri ile Destek Vektör Makineleri, PC üzerinde eğitilmiş ve sınıflandırıcıda kullanılacak olan karar fonksiyonuna ait destek vektörleri, ağırlık vektörü ve bias parametreleri elde edilmiştir. Bu tez çalışmasında bire karşı hepsi yöntemiyle, doğrusal kernel fonksiyonlu destek vektör yapısı kullanılmıştır. Bu sebeple üzerinde çalışılan 40 sınıflı yüz sınıflandırma problemi için, her biri, bir sınıfı ayırmak için eğitilmiş Destek Vektör Makinelerine ait, 40 adet 30 uzunluklu destek vektörü (x_i), 30 uzunluklu ağırlık vektörü (α_i) ve bias (b) elde edilmiştir. Yine test edilecek yüz görüntüsünün öznitelikleri PC üzerinde elde edilir ve FPGA veya GPU birimlerine iletilir. FPGA veya GPU birimlerine gelen öznitelik vektörleri, eşitlik (4.1)'e göre gerçeklenen sınıflandırıcılar üzerinde sınıflandırılır.

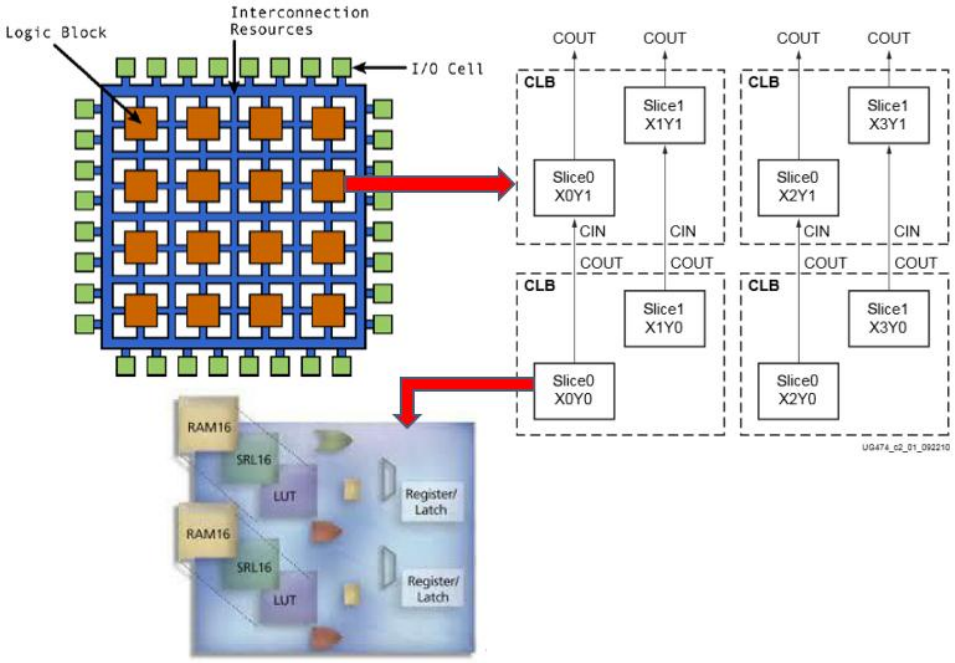
$$\mathrm{c} = \sum_i \alpha_i (x * x_i) + b \tag{4.1}$$

x öznitelik vektörünün ait olduğu sınıfın bulunabilmesi için, her farklı sınıf için eğitilen Destek Vektör Makinesi karar fonksiyonunda sınıflandırma işlemine tabi tutulur. FPGA

ve GPU üzerinde, 40 grup için, her biri bir grubu sınıflandırmak için eğitilmiş Destek Vektör Makinelerine ait karar fonksiyonu ile eşitlik (4.1)'e göre gerçeklenen, 40 adet sınıflandırıcı birimlerine, PC üzerinden gönderilen 30 uzunluklu yüz öznitelik vektörleri eş zamanlı olarak işletilir ve her sınıflandırıcı sonuçları aynı anda elde edilir. Her bir sınıflandırıcıların sonuçları $c \geq 0$ ise, x giriş öznitelik vektörü yoklanan kişiye aittir değilse yoklanan kişiden farklı bir sınıfa aittir denir.

4.1 FPGA Üzerinde DVM Sınıflandırıcısının Gerçeklenmesi

Sahada Programlanabilir Kapı Dizileri (FPGA), tasarımcının ihtiyaç duyduğu her türlü mantık işlevlerini gerçekleştirme amacına yönelik, istenen fonksiyona göre içyapısı kullanıcı tarafından değiştirilebilen ve paralel çalışma olanağı sağlayan programlanabilir tümleşik devrelerdir. Temel olarak programlanabilir mantık blokları, mantık blokları arasındaki programlanabilir bağlantılar, programlanabilir giriş / çıkış birimlerinden oluşur.

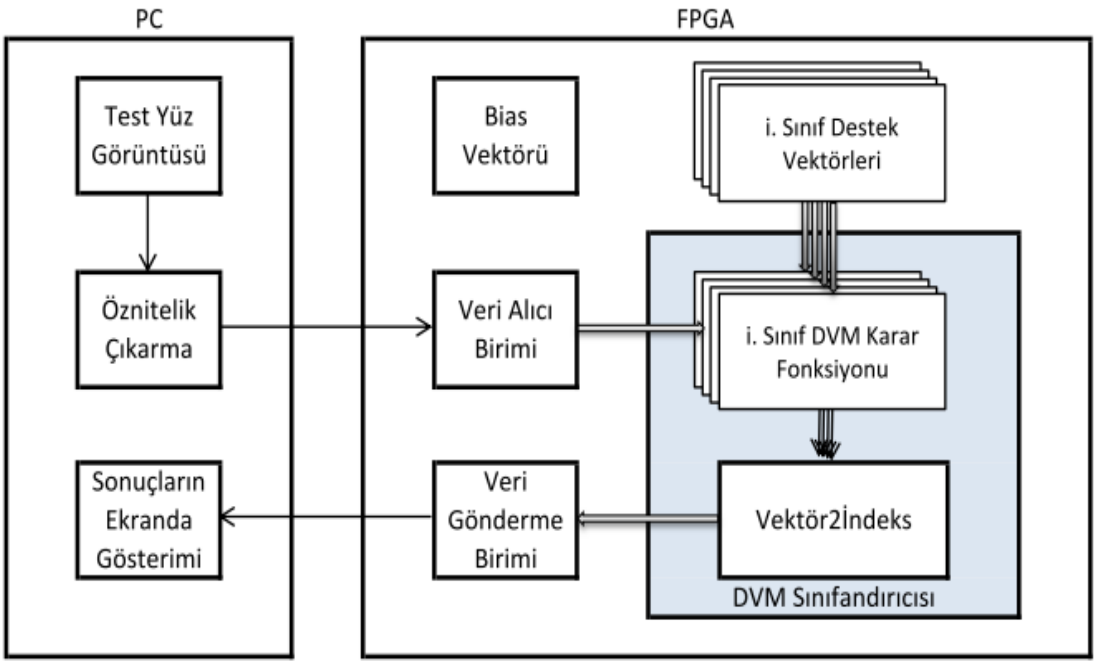


Şekil 4. 1 FPGA yapısı

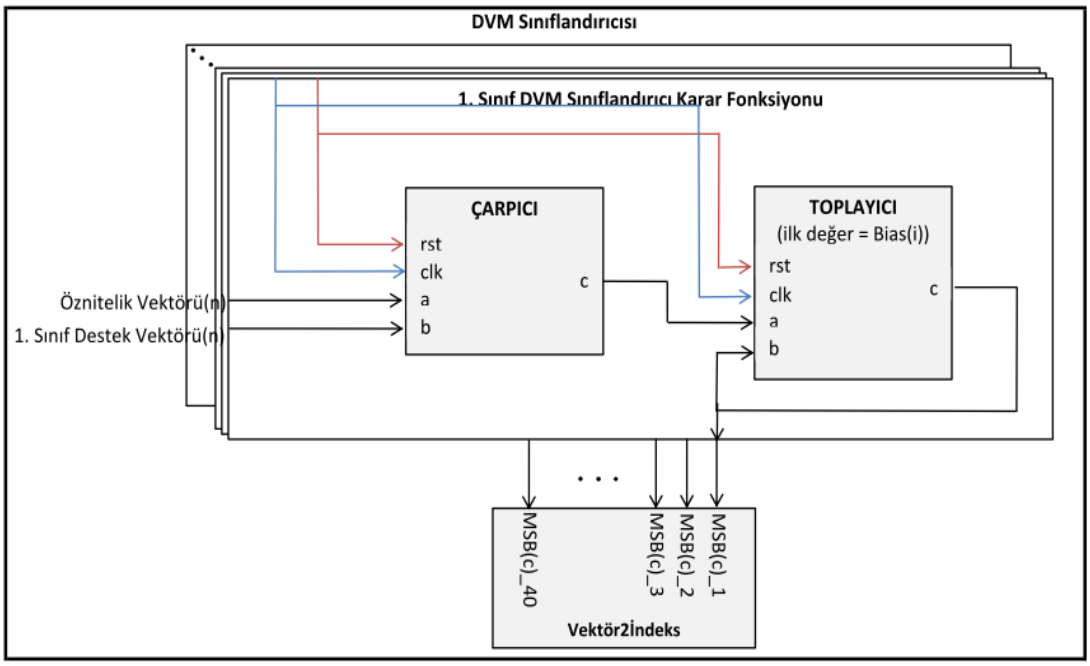
Bu tez çalışmasında, FPGA üzerinde, ORL veritabanına ait yüz verilerini sınıflandırmak üzere eğitilmiş DVM sınıflandırıcısı gerçeklenmiştir. Gerçeklenen bu yapıda; her biri, bir yüz setine ait özniteliklerle eğitilmiş 40 adet sınıflandırıcı karar fonksiyonu birimleri, eş zamanlı sonuç elde edecek şekilde paralel olarak tasarlanmıştır.

Şekil 4. 2'de gösterildiği gibi, sınıflandırılacak test yüz görüntüsünden elde edilen öznitelik vektörleri PC üzerinden FPGA'ya iletir ve FPGA sınıflandırma işlemini gerçekleştirir. Daha sonra elde ettiği sınıflandırma sonucunu tekrar PC'ye iletir.

Şekil 4. 2'de FPGA üzerinde gerçeklenen sistemin genel blok diyagramı, Şekil 4. 3'te sınıflandırıcı yapısına ait blok diyagram yer almaktadır.



Şekil 4. 2 FPGA üzerinde geliştirilen sistemin blok diyagramı



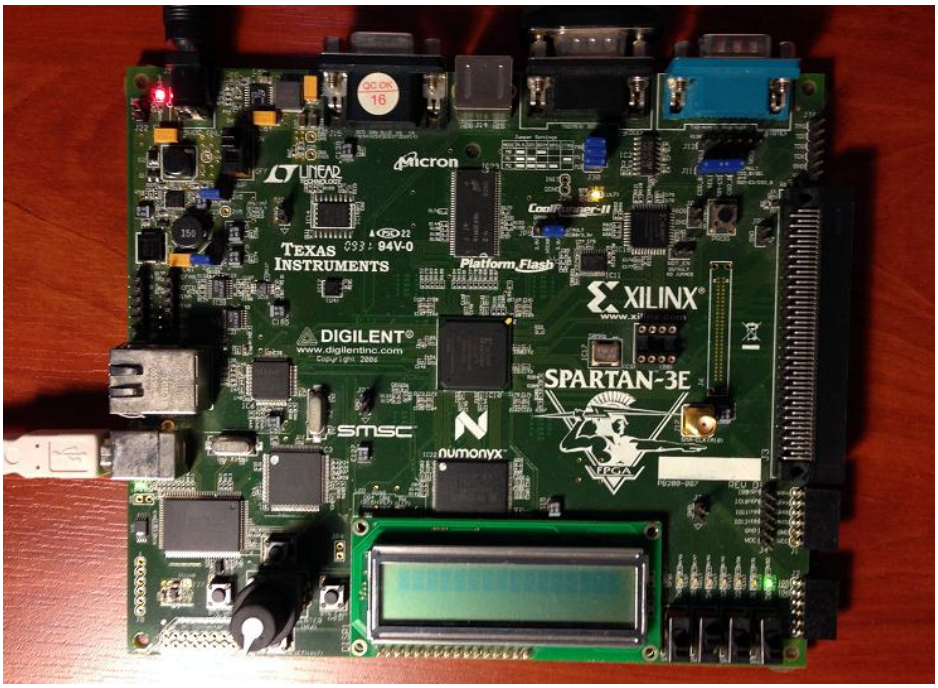
Şekil 4. 3 FPGA üzerinde gerçeklenen DVM sınıflandırıcı blok diyagramı

DVM sınıflandırıcı karar fonksiyonuna ait parametreler ve yüz görüntülerine ait öznitelik vektörleri ondalıklı sayılardan oluşmaktadır. FPGA üzerinde kullanabilmek üzere, bu ondalıklı sayıları ikilik tabanda gösterebilmek için sabit noktalı sayı formatı kullanılmıştır. Bu sebeple FPGA üzerinde tasarlanan aritmetik işlem birimleri sabit noktalı sayı formatı kullanılarak geliştirilmiştir.

DVM karar fonksiyonu çıkışındaki toplama sonucu verisinin en anlamlı biti olan işaret biti, ilgili öznitelik vektörüne ait sınıflandırma sonucunu vermektedir. Tüm sınıflandırıcıların çıkışındaki verilerin en anlamlı bitleri "Vektör2İndex" biriminde değerlendirilerek girdi öznitelik vektörünün hangi sınıfa ait olduğu hesaplanır.

Her bir DVM sınıflandırıcı karar fonksiyonu çıkışı öznitelik vektörlerimizin uzunluğu olan 30 saat çevriminde, toplama biriminden kaynaklanan +1 saat çevrimi ile 31 saat çevriminde tamamlanır. Sınıflandırıcı çıktıları iletilen Vektör2İndex birimi de değerlendirmeyi 1 saat çevriminde tamamlar. FPGA üzerinde geliştirilen sınıflandırıcı sonucu toplamda 32 saat çevrimi sonra elde edilmiş olur.

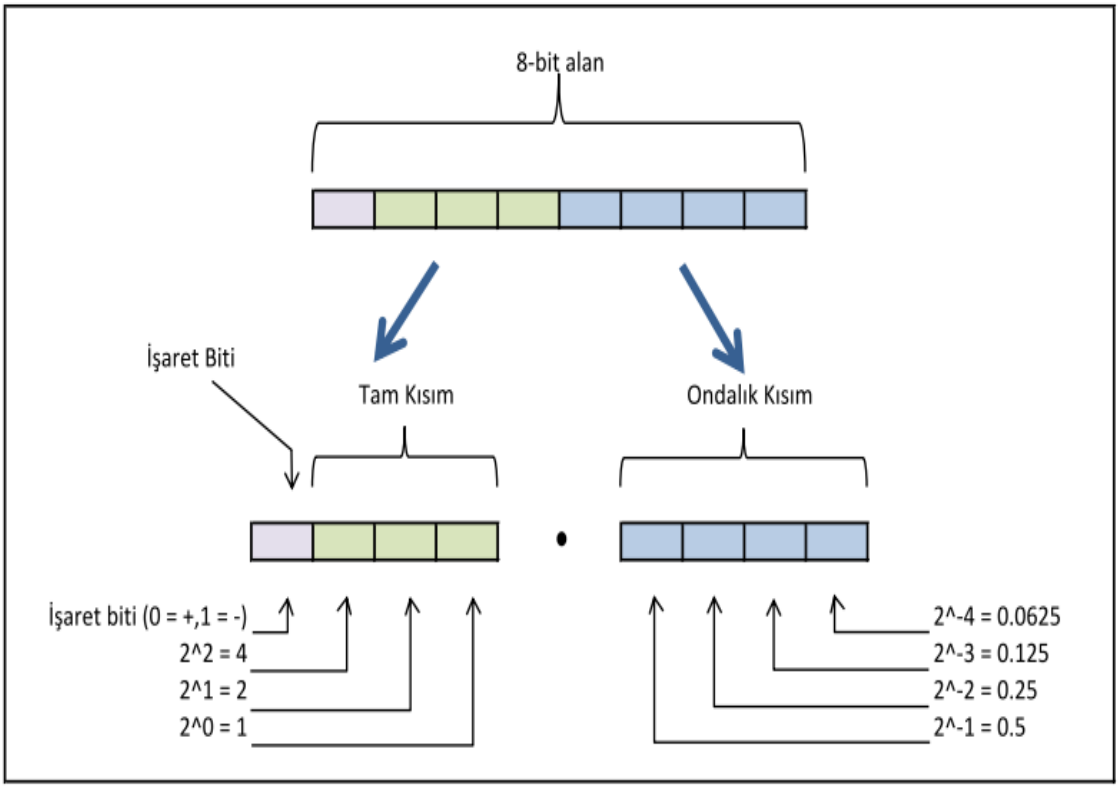
FPGA kodları VHDL ile yazılmış olup, ISE Design Suite 13.1 geliştirme ortamında çalışılmış ve Xilinx Spartan 3E XC3S500E FPGA ve Xilinx Virtex5 XC5VLX50T FPGA üzerinde performansı değerlendirilmiştir.



Şekil 4. 4 Xilinx Spartan 3E XC3S500E FPGA geliştirme kartı

4.1.1 Sabit Noktalı Sayılar

Sabit noktalı sayı formatı $Qm.n$ ile sembolize edilir ve sabit noktalı sayı formatı, Şekil 4. 5'de gösterildiği gibi sayının tam kısmı ve ondalık kısmı ile yazılır. Sayı formatındaki m sayısı tam sayı bit sayısını belirtirken, n ondalık kısmına ait bit sayısını temsil eder. İşaretli sayılarda, sayının işaretini belirtmek üzere tam kısımda yer alan en yüksek ağırlıklı bit tanımlanır. Bu sayı formatında tasarlanacak sisteme göre, istenilen sayının maksimum büyüklüğü ve hassasiyet göz önünde bulundurularak bit uzunlukları belirlenir.

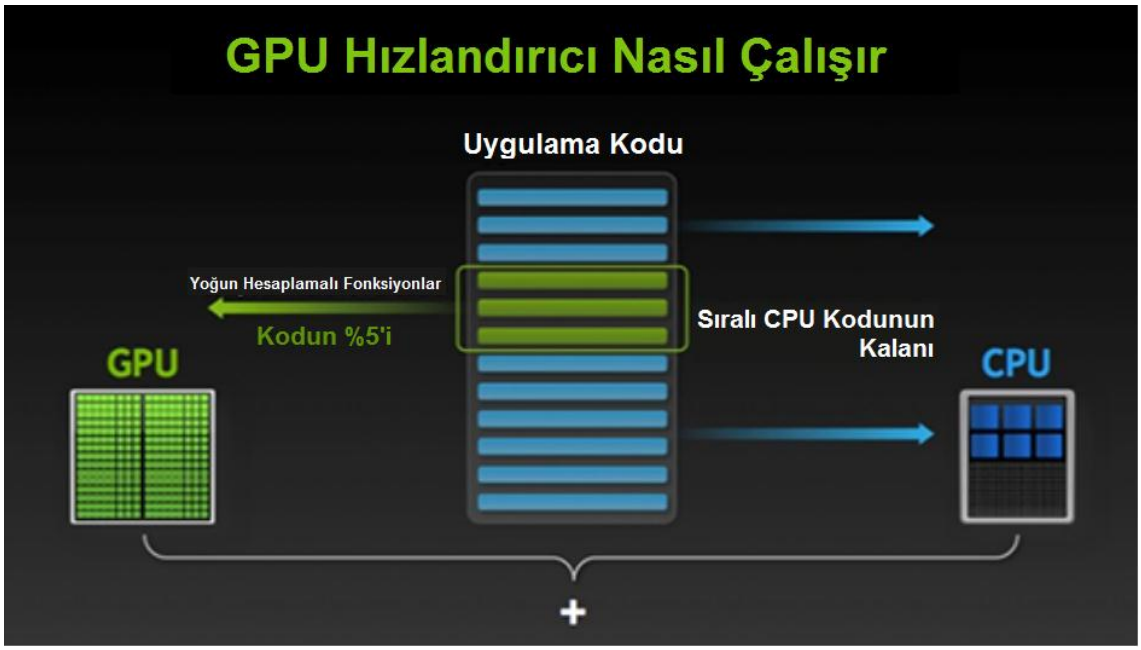


Şekil 4. 5 Örnek olarak $Q4.4$ sabit noktalı sayı formatı gösterimi

FPGA üzerinde tasarlanan DVM sınıflandırıcısında aritmetik işlemler, sabit noktalı sayı formatı kullanılarak tasarlanmıştır. Eğitilen Destek Vektör Makinesi parametreleri ve öznitelik vektörleri, Fisher yüz öznitelikleri kullanıldığında 1 bit işaret biti olmak üzere 16 bit genişliğinde, $Q4.12$ hassasiyetli sabit noktalı sayı formatı ile ifade edilebilirken, özyüz öznitelikleri kullanıldığında 1 bit işaret biti olmak üzere 17 bit genişliğinde, $Q5.12$ hassasiyetli sabit noktalı sayı formatı ile ifade edilir.

4.2 GPU Üzerinde DVM Sınıflandırıcısının Gerçeklenmesi

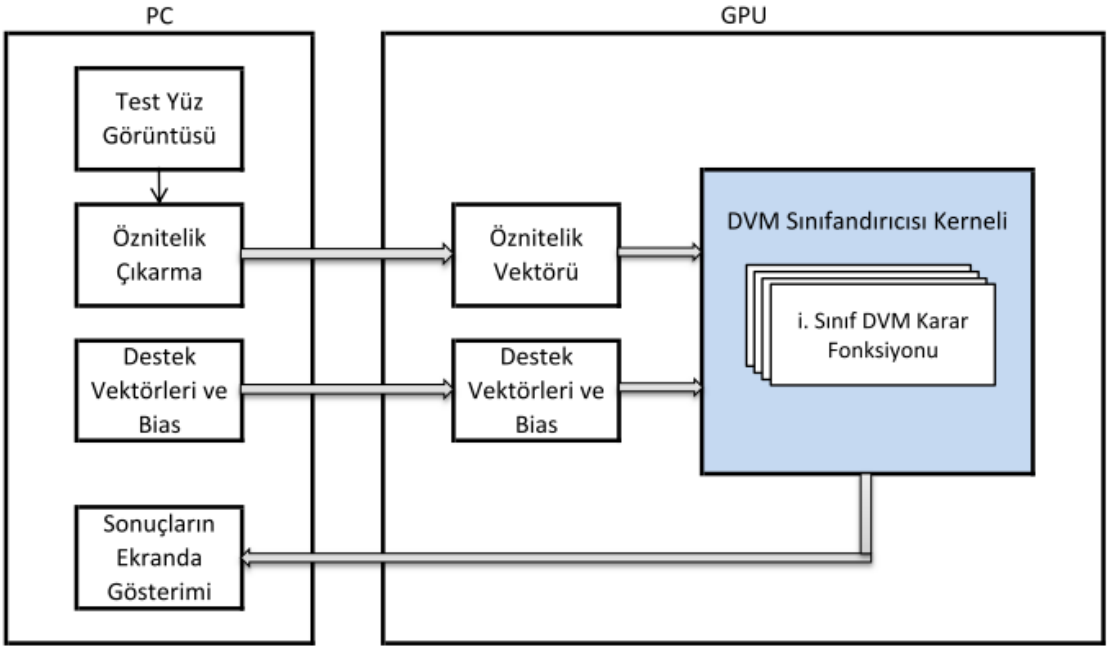
Grafik İşlem Birimi (GPU) grafik oluşturma ve görüntü işlemenin getirdiği işlem yükünü karşılayabilecek şekilde tasarlanmış gömülü bir sistemdir. Özel amaçlı uP denilebilir fakat programlama alanları sınırlıdır. Çok çekirdekli yapısı sayesinde yoğun veri bloklarını CPU'lara kıyasla paralel olarak etkili ve hızlı şekilde işler [31]. Şekil 4. 6'da gösterildiği gibi uygulama kodunun yüksek hesaplama gerektiren kısımları, CPU üzerinden kontrollü olarak GPU'ya aktarılır ve GPU üzerinde işletilerek hızlandırılır.



Şekil 4. 6 GPU üzerinde yoğun hesaplamalı işlemlerin çalıştırılması [31]

Bu tez çalışmasında, GPU üzerinde, ORL veritabanına ait yüz verilerini sınıflandırmak üzere eğitilmiş DVM sınıflandırıcısı gerçeklenmiştir. Gerçeklenen bu yapıda; her biri, bir yüz setine ait özniteliklerle eğitilmiş 40 adet sınıflandırıcı karar fonksiyonu birimleri, eş zamanlı sonuç elde edecek şekilde paralel olarak tasarlanmıştır. Şekil 4. 7'de GPU ile gerçeklenen sistemin genel blok diyagramı gösterilmektedir.

DVM sınıflandırıcının GPU üzerinde koşması için C++ dili ile Visual Studio 2010 üzerinde CUDA kernel fonksiyonu geliştirilmiş ve NVIDIA GeForce GTX 480 üzerinde performansı değerlendirilmiştir. DVM sınıflandırıcısına ait GPU üzerinde koşan CUDA Kernel Fonksiyonu, EK-A da yer almaktadır.



Şekil 4. 7 GPU ile gerçeklenen sistemin blok diyagramı

BÖLÜM 5

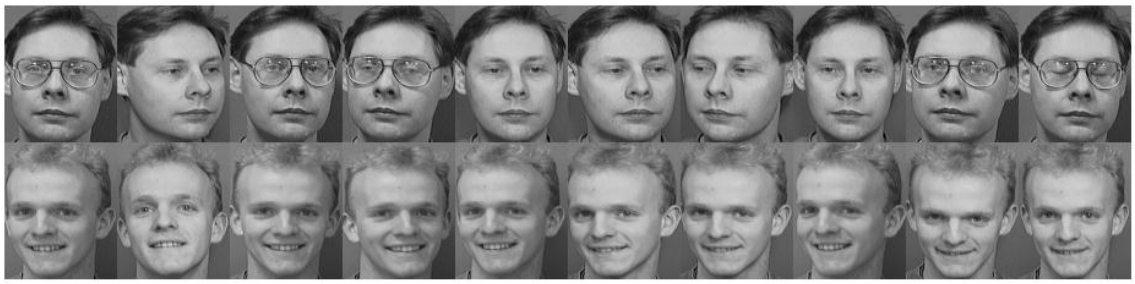
DENEYSEL ÇALIŞMALAR

5.1 ORL Yüz Veritabanı

Önerilen yöntemin performansını ölçebilmek için kullanılan veritabanı yüz görüntüleri, Nisan 1992 ve Nisan 1994 yılları arasında Cambridge'te bulunan Olivetti Araştırma Laboratuvarı'nda yapılan çalışmalar sonucunda elde edilmiştir. Veritabanındaki yüz görüntüleri, 40 farklı kişiye ait farklı özelliklerde çekilmiş 10 görüntüden oluşur. Toplamda 400 görüntü mevcuttur. Aynı kişiler için elde edilen 10 tane görüntü, farklı zamanlarda, farklı ışık şiddetinde, farklı yüz ifadeleri, yüz ayrıntıları (gözlüklü / gözlüksüz) ve farklı kafa yönlenmesi ile çekilen görüntülerden oluşturulmuştur. Her bir görüntünün arka zemini aynı oranda karartılmıştır. Görüntü boyutları 92 X 112 piksel, 8-bit gri ton ve pgm uzantılı olarak kaydedilmiştir [30].

Şekil 5. 1'de ORL yüz veritabanından örnekler görülmektedir.

Şekil 5. 1 ORL yüz veritabanından beş farklı kişiye ait örnek yüz görüntüleri



Şekil 5. 1 ORL yüz veritabanından beş farklı kişiye ait örnek yüz görüntüleri (devamı)

5.2 Sınıflandırıcı Test Sonuçları

Geliştirilen sistemi eğitmek için ORL veritabanında bulunan her bir kişiye ait özel olarak seçilen 4 yüz görüntüsü kullanılmıştır. Her kişiye ait kalan 6 yüz görüntüsü tanıma testi için kullanılmıştır. 30 uzunluklu özyüzler öznitelik vektörleri ile eğitilen Destek Vektör Makinelerinin eğitim verilerini sınıflandırma başarısı, %100, test verilerini sınıflandırma başarısı ise %90 seviyesindedir. 30 uzunluklu Fisher yüz öznitelik vektörleri ile eğitilen Destek Vektör Makineleri eğitim verilerini sınıflandırma başarısı yine %100 iken test verilerini sınıflandırma başarısı %91 seviyesine ulaşmaktadır.

Çizelge 5. 1'de özel olarak (çeşitli yönelim ve aksesuarlarla) seçilen eğitim verileri ile 40 farklı kişinin yüz görüntüsü için eğitilen çeşitli sınıflandırıcıların, 240 adet test yüz görüntüsü kullanılarak MATLAB üzerinde gerçekleştirilen sınıflandırma başarım oranları yer almaktadır.

Çizelge 5. 1 Özel olarak seçilen eğitim verileri ile eğitilen sınıflandırıcıların MATLAB üzerinde sınıflandırma başarım sonuçları

Yöntem	**Test Başarısı (%)**
Özyüzler + DVM (Doğrusal kernel)	90
Fisher Yüzler + DVM (Doğrusal kernel)	91
Özyüzler + YSA	90
Fisher Yüzler + YSA	92
Özyüzler + kNN (k=4)	89
Fisher Yüzler + kNN (k=4)	88

Çizelge 5. 2'de 10 farklı adımda, 40 farklı kişinin yüz görüntülerinden rastgele seçilerek oluşturulan eğitim verileri ile eğitilen çeşitli sınıflandırıcıların, 240 adet yüz görüntüsü

kullanılarak MATLAB üzerinde gerçekleştirilen sınıflandırma başarım oranlarının ortalaması ve 10 adımda harcanan toplam eğitim ve test süresi yer almaktadır.

Çizelge 5. 2 Rastgele olarak seçilen eğitim verileri ile eğitilen sınıflandırıcıların MATLAB üzerinde sınıflandırma başarım sonuçları ve harcanan süre

Yöntem	**Test Başarısı (%)**	**Eğitim + Test Süresi (sn.)**
Özyüzler + DVM (Doğrusal kernel)	78	21.7
Fisher Yüzler + DVM (Doğrusal kernel)	80	17.3
Özyüzler + YSA	83	4726.5
Fisher Yüzler + YSA	83	4471.4
Özyüzler + kNN (k=4)	85	14.6
Fisher Yüzler + kNN (k=4)	84	14.8

Destek Vektör Makinelerinde kullanılan kernel yapısı, sınıflandırma başarım sonucunu verinin yapısına bağlı olarak büyük oranda etkilemektedir.

Çizelge 5. 3'te farklı kernel yapılarının özyüz ve Fisher yüz öznitelik vektörleriyle olan başarım sonuçları yer almaktadır.

Tasarlanan yüz tanıma sisteminde en iyi sonuçların doğrusal kernel yapılı Destek Vektör Makinesinde elde edildiği gözlemlenmiştir.

Çizelge 5. 3 Farklı kernel yapılarındaki DVM sınıflandırıcıların başarım sonuçları

Kernel Tipi	**Özyüzler + DVM Sınıflandırıcı Test Başarısı (%)**	**Fisher Yüzler + DVM Sınıflandırıcı Başarısı (%)**
Doğrusal	90	91
Radyal Tabanlı	7	11
Polinomial	55	81
tanh	6	5

Kullanılan öznitelik sayısı sınıflandırma sonucunu etkilemektedir.

Çizelge 5. 4'te farklı öznitelik sayılarında özyüzler + doğrusal kernelli DVM sınıflandırıcı yapısının sınıflandırma başarım sonuçları yer almaktadır. En iyi sonuç en ağırlıklı 30 özyüz özelliğinin kullanılmasıyla elde edilmiştir.

Çizelge 5. 4 Farklı öznitelik sayılarındaki özyüzler + doğrusal kernelli DVM sınıflandırıcı yapısının sınıflandırma başarım sonuçları

Özyüz Sayısı	**Test Başarısı(%)**
10	65
20	87
30	90
40	89
50	88
80	85
160	83

Çizelge 5. 5'de farklı öznitelik sayılarında Fisher yüzler + doğrusal kernelli DVM sınıflandırıcı yapısının sınıflandırma sonuçları yer almaktadır. En iyi sonuç en ağırlıklı 30 Fisher yüz özniteliğinin kullanılmasıyla elde edilmiştir.

Çizelge 5. 5 Farklı öznitelik sayılarındaki Fisher yüzler + doğrusal kernelli DVM sınıflandırıcı yapısının sınıflandırma başarım sonuçları

Fisher Yüz Sayısı	**Test Başarısı (%)**
10	65
20	88
30	91
40	90
50	91
80	91
160	91

5.3 Donanım Test Sonuçları

Bu tez çalışmasında Destek Vektör Makineleri kullanılarak FPGA ve GPU üzerinde gerçeklenen bir yüz tanıma sistemi önerilmiştir. Bu sistemde ORL veritabanına ait yüz görüntülerinden özyüz ve Fisher yüz öznitelik vektörlerinin sınıflandırılması için eğitilen Destek Vektör Makinelerinin sınıflandırma performansı FPGA üzerinde gerçekleştirilen

sınıflandırıcı ile test edilmiştir. FPGA üzerinde gerçekleştirilen bu sistem ayrıca GPU ile gerçekleştirilerek performansları karşılaştırılmıştır.

5.3.1 FPGA Kaynak Kullanımı

Fisher yüzler + doğrusal kernelli DVM Sınıflandırıcısının, Xilinx Spartan 3E XC3S500E FPGA kaynak kullanımı Çizelge 5. 6'da yer almaktadır. Sistemin hesaplanan maksimum saat frekansı 70.912MHz'dir.

Çizelge 5. 6 Fisher yüzler + doğrusal kernelli DVM sınıflandırıcısı Xilinx Spartan 3E XC3S500E FPGA kaynak kullanımı

Blok Adı	**Kullanılan**	**Var Olan**	**Kullanım Oranı (%)**
Slice Flip Flop	1660	9312	17
Slice 4 giriş LUT	3971	9312	42
Slice	2272	4656	48
IOBs	5	232	2
BUFGMUXs	1	24	4
MULT18X18SIOs	20	20	100

Fisher yüzler + doğrusal kernelli DVM sınıflandırıcısının, Xilinx Virtex5 XC5VLX50T FPGA kaynak kullanımı Çizelge 5. 7'de yer almaktadır. Sistemin hesaplanan maksimum saat frekansı 203.344MHz'dir.

Çizelge 5. 7 Fisher yüzler + doğrusal kernelli DVM sınıflandırıcısı Xilinx Virtex5 XC5VLX50T FPGA kaynak kullanımı

Blok Adı	**Kullanılan**	**Var Olan**	**Kullanım Oranı(%)**
Slice Register	1583	28000	5
Slice LUT	1168	28000	4
Slice	567	7200	7
IOBs	5	480	1
BUFG/BUFGCTRLs	1	32	3
DSP48E	40	48	83

Özyüzler + doğrusal kernelli DVM sınıflandırıcısının, Xilinx Spartan 3E XC3S500E FPGA kaynak kullanımı Çizelge 5. 8'de yer almaktadır. Sistemin hesaplanan maksimum saat frekansı 70.912MHz'dir.

Çizelge 5. 8 Özyüzler + doğrusal kernelli DVM sınıflandırıcısı Xilinx Spartan 3E XC3S500E FPGA kaynak kullanımı

Blok Adı	**Kullanılan**	**Var Olan**	**Kullanım Oranı(%)**
Slice Flip Flop	1731	9312	18
Slice 4 giriş LUT	4162	9312	44
Slice	2398	4656	51
IOBs	5	232	2
BUFGMUXs	1	24	4
MULT18X18SIOs	20	20	100

Özyüzler + doğrusal kernelli DVM sınıflandırıcısının, Xilinx Virtex5 XC5VLX50T FPGA kaynak kullanımı Çizelge 5. 9'da yer almaktadır. Sistemin hesaplanan maksimum saat frekansı 203.344 MHz'dir.

Çizelge 5. 9 Özyüzler + doğrusal kernelli DVM sınıflandırıcısı Xilinx Virtex5 XC5VLX50T FPGA kaynak kullanımı

Blok Adı	**Kullanılan**	**Var Olan**	**Kullanım Oranı(%)**
Slice Register	1623	28000	5
Slice LUT	1268	28000	4
Slice	618	7200	8
IOBs	5	480	1
BUFG/BUFGCTRLs	1	32	3
DSP48E	40	48	83

5.3.2 FPGA ve GPU Performanslarının Karşılaştırılması

Yüz tanıma donanım testlerinde, Xilinx Spartan3E serisi XC3S500E, Xilinx Virtex5 serisi XC5VLX50T FPGA'ler ve NVIDIA GeForce GTX 480 GPU kullanılmıştır.

Spartan3E serisi FPGA kartı giriş saat hızı 50 MHz ve Virtex5 serisi FPGA kartı giriş saati 100 MHz'dir.

Tasarlanan FPGA sınıflandırma bloğu, sınıflandırma işlemi sonucunu, "(öznitelik vektörü boyutu + 2)*saat hızı" kadar sürede elde edebilmektedir. Sınıflandırma testleri sonrası en iyi özyüz ve Fisher yüz öznitelikleri kullanılarak oluşturulan her iki sınıflandırıcıda da öznitelik vektörü sayısı 30'dur. Bu şartlarda iki sınıflandırıcı için sınıflandırma süresi Spartan3E FPGA kartında 0.64 ms iken Virtex5 FPGA kartında 0.32 ms olarak hesaplanmaktadır.

NVDIA Geforce GTX 480 GPU kartı grafik saat hızı 607 MHz ve işlemci saat hızı 1215 MHz'dir. 448 adet CUDA core bulunur. CUDA 32-bit integer ve single precision floating point sayı tiplerini destekler. Tasarlanan CUDA sınıflandırma bloğu hesaplama süresi, en iyi özyüz ve Fisher yüz öznitelikleri kullanılarak oluşturulan her iki sınıflandırıcı için sınıflandırma süresi ortalama 0.031 ms'dir. Farklı donanımlarda elde edilen performansların karşılaştırılması Çizelge 5. 10'da yer almaktadır.

Çizelge 5. 10 FPGA GPU performans karşılaştırması

Donanım Adı	**Sınıflandırma Süresi (ms)**
Spartan3E XC3S500E FPGA	0.64
Virtex5 XC5VLX50T FPGA	0.32
NVDIA Geforce GTX 480 GPU	0.031

BÖLÜM 6

SONUÇ VE ÖNERİLER

Bu tez kapsamında ilk aşamada, özyüz ve Fisher yüz tabanlı öznitelik vektörleri kullanılarak doğrusal kernelli Destek Vektör Makinesi üzerinde sınıflandırma yapılmıştır. Elde edilen başarım sonuçları, çok katmanlı bir Yapay Sinir Ağı ve k En Yakın Komşuluk sınıflandırıcıları kullanılarak elde edilen başarım sonuçları ile karşılaştırılmıştır.

İkinci aşamada, doğrusal kernelli DVM sınıflandırıcısı FPGA ve GPU üzerinde gerçeklenmiş ve performansları karşılaştırılmıştır. Donanım üzerinde gerçeklenen sınıflandırıcıda, doğrusal kernelli DVM yapısı tercih edilmesinin sebebi, düşük karmaşıklık ve düşük işlem yükü gerektirmesindendir. Doğrusal kernelli DVM sınıflandırıcısı Çok Katmanlı Yapay Sinir Ağı ve k En Yakın Komşuluk sınıflandırıcısına göre çok daha basittir ve çok daha hızlı çalışmaktadır.

DVM sınıflandırıcısı ile gerçeklenen sistem, 40 kişi ve her kişiye ait 10 farklı yüz görüntüsünden oluşan ORL yüz veritabanı kullanılarak eğitilmiş ve test edilmiştir. Destek Vektör Makinelerinin eğitimi sırasında her bir kişi için o kişiye ait 4 yüz görüntüsü kullanılmıştır. Kalan 6 yüz görüntüsü ile toplamda 240 adet yüz görüntüsü üzerinde sınıflandırma testi gerçekleştirilmiştir. DVM sınıflandırıcısının başarısı özyüzler kullanıldığında %90'lara ulaşırken, Fisher yüzler kullanıldığında %91 seviyesine ulaşmıştır.

FPGA ve GPU üzerinde, yüz tanıma gerçekleştirmek üzere tasarlanan doğrusal kernelli DVM sınıflandırıcıların sınıflandırma süresi olarak GPU daha yüksek performans

göstermiştir. Yine GPU tasarımı FPGA tasarımına göre çok daha hızlıdır ve bu özelliği GPU kullanımını üstün kılar. Fakat GPU'nun bir işlemci ile kullanılma zorunluluğu ve enerji tüketimi ise FPGA'yı daha avantajlı hale getirir.

KAYNAKLAR

[1] Ahonen, T., Hadid, A. ve Pietikäinen, M., (2006). "Face Description with Local Binary Patterns: Application to Face Recognition", IEEE Transactions on Pattern Analysis and Machine Intelligence, 28, 12.

[2] Turk, M. A. ve Pentland, A. P., (1991). "Face Recognition Using Eigenfaces", In IEEE Computer Society Conference Computer Vision and Pattern Recognition, 586-591.

[3] Belhumeur, P.N., Hespanha, J.P. ve Kriegman, D.J., (1997). "Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection", IEEE Trans. PAMI 19, 711-720.

[4] Suhas, S., Kurhe, A. ve Khanale, Dr.P., (2012). "Face Recognition Using Principal Component Analysis and Linear Discriminant Analysis on Holistic Approach in Facial Images Database", IOSR Journal of Engineering, 15-23.

[5] Sirovich, L. ve Kirby, M., (1987). "Low-Dimensional Procedure for The Characterization of Human Faces", Optical Society of America, 519- 525.

[6] Brunelli, R. ve Poggio, T., (1992). "Face Recognition through Geometrical Features", In: Proc. of ECCV'92, 792-800.

[7] Brunelli, R. ve Poggio T., (1995). "Template Matching: Matched Spatial Filters and Beyond", Tech. Rep., 95-103.

[8] Jiminez, D. G. ve Casto, J. A., (2004). "Frontal Face Authentication through Creaseness-Driven Gabor Jets", ICIAR, 660-667.

[9] Gao, Y. ve Leung, M., (2002). "Face Recognition Using Line Edge Map", IEEE Transactions on Pattern Analysis and Machine Intelligence, 764-779.

[10] Nefian, A.V. ve Hayes, M.H., (1998). "Hidden Markov Models for Face Recognition", In: Acoustics, Speech and Signal Processing, IEEE International Conference on, 2721-2724.

[11] Garcia, C., Zikos, G. ve Tziritas, G., (2000). "Wavelet Packet Analysis for Face Recognition", Image and Vision Computing 18, 289-297.

[12] Goldstein, A. J., Harmon, L. D. ve Lesk, A. B., (1971). "Identification of Human Faces" Proc. IEEE, Vol. 59, No. 5, 748-760.

[13] Kanade T., (1973). “Picture Processing System by Computer Complex and Recognition of Human Faces”, PhD. Thesis. PhD thesis, Kyoto University, Japan.

[14] Manjunath, B.S., Chellappa, R. ve von der Malsburg, C., (1992). “A Feature Based Approach to Face Recogniton”, Technical Report CS-TR-2834, Center for Automated Research. Univ. of Maryland.

[15] Cox, I. J., Ghosn, J. ve Yianilos, P. N., (1996). “Feature-Based Face Recognition Using Mixture-Distance”, Proceedings of the 1996 Conference on Computer Vision and Pattern Recognition (CVPR '96), 209.

[16] Wiskott, L., Fellous, J., Krueuger, N. ve von der Malsburg, C., (1997). “Face Recognition by Elastic Bunch Graph Matching”, IEEE Trans. On Pattern Analysis and Machine Intelligence, 776-779.

[17] Fleming, M. ve Cottrell, G., (1987). “Categorization of Faces Using Unsupervised Feature Extraction”, In Proc. IEEE IJCNN Inter- national Joint Conference on Neural Networks, 65–70, 90.

[18] Jonsson, K., Matas, J., Kittler, J. ve Li, Y., (2000). “Learning Support Vectors for Face Verification and Recognition”, InProc. IEEE International Conference on Automatic Face and Ges-ture Recognition, 208–213.

[19] Ouerhani, Y., Jridi, M. ve Alfalou, A., (2010). “Fast Face Recognition Approach Using a Graphical Processing Unit “GPU””, Imaging Systems and Techniques (IST)”, IEEE International Conference.

[20] Endluri, R., Kathait, M. ve Ray, K.C., (2013). “Face Recognition Using PCA on FPGA Based Embedded Platform”, Control, Automation, Robotics and Embedded Systems (CARE).

[21] Neggazi, M., Bengherabi, M., Boulkenafet, Z. ve Amira, A., (2006). “An Efficient FPGA Implementation of Gaussian Mixture Models Based Classifier: Application to Face Recognition”.

[22] Dimitrios, S. A., Papastergiou, A. ve Hatzigaidas, A., (2011). “Real-Time Face Recognition with GPUs, A DCT-Based Face Recognition System Using Graphics Processing Unit”, Signal Processing and Multimedia Applications (SIGMAP), Proceedings of the International Conference.

[23] Cortes, C. ve Vapnik, V., (1995). “Support-Vector Network”, Machine Learning, 20(3): 273–297.

[24] İTÜ Veri Madenciliği Ders Notları, http://www.ninova.itu.edu.tr/tr/dersler/bilisim-enstitusu/195/bbl-606/ekkaynaklar?g8396 ,10 Eylül 2015.

[25] Vapnik, V.N., (2000). The Nature of Statistical Learning Theory, 2. Baskı, Springer-Verlag, New York.

[26] Kavzoğlu, T. ve Çölkesen, İ., (2010). “Destek Vektör Makineleri ile Uydu Görüntülerinin Sınıflandırılmasında Kernel Fonksiyonlarının Etkilerinin İncelenmesi”, Harita Dergisi, 144: 73-82.

[27] Fletcher, T., (2009). Support Vector Machines Explained, http://www.tristanfletcher.co.uk/SVM Explained.pdf, 07 Şubat 2015.

[28] Gökmen, M., Kurt, B., Kahraman, F. ve Çapar, A., (2007). "Çok Amaçlı Gürbüz Yüz Tanıma", Proje No: 104E121, TÜBİTAK.

[29] Zhao, W., Chellappa, R., Krishnaswamy, A., Sweats, D. L. ve Weng, J., (1998). "Discriminant Analysis of Principal Components for Face Recognition", Face Recognition, 73-85.

[30] Cambridge University, C. L., (1994). The Olivetti Research Ltd Yüz Veritabanı, http://www.uk.research.att.com/facedatabase.html, 05 Ocak 2015.

[31] What is GPU Accelerated Computing?, http://www.nvidia.com/object/what-is-gpu-computing.html, 12 Ekim 2015.

EK-A

CUDA KERNEL FONKSİYONU

```
__global__ void SVMClassifier_CUDA(float *c, float *imageList, const float *SVM_BIAS,
const float *SVM_SUPPORT_VECTORS)
{   int j = blockDim.x * blockIdx.x + threadIdx.x;
   int n = blockDim.x;
   if (j<GROUP_COUNT)
   {  float _c = 0.0f;
     for (int k = 0; k<n; k++)
    {
      _c+=imageList[k]*SVM_SUPPORT_VECTORS[j*n+k];
    }
    c[j]=_c+SVM_BIAS[j];
   }
}
```

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı : Hilal GÜNEREN

Doğum Tarihi ve Yeri : 14.11.1988 / Merzifon

Yabancı Dili : İngilizce

E-posta : hilalguneren@gmail.com

ÖĞRENİM DURUMU

Derece	Alan	Okul/Üniversite	Mezuniyet Yılı
Lisans	Elektronik ve Haberleşme Mühendisliği	Yıldız Teknik Üniversitesi	2010
Lise	Fen Bilimleri	Merzifon Anadolu Lisesi	2006

İŞ TECRÜBESİ

Yıl	Firma/Kurum	Görevi
2010 - Devam Ediyor	TÜBİTAK BİLGEM-BTE	Araştırmacı