**KOCAELİ ÜNİVERSİTESİ MÜHENDİSLİK FAKÜLTESİ**

**SES TEMELLİ MALZEME TANIMA**

**LİSANS TEZİ**

**Ender GÖREN**

**Bölümü: Elektronik ve Haberleşme Mühendisliği**

**Danışman: Doç. Dr. Kemal GÜLLÜ**

**KOCAELİ, 2016**

TEŞEKKÜR

Tez çalışmam boyunca bana yardımcı olan ve önerileri ile beni yönlendiren tez danışmanım Doç. Dr. Kemal GÜLLÜ ‘e katkılarından dolayı çok teşekkür ederim.

Mayıs 2016, KOCAELİ Ender GÖREN

İÇİNDEKİLER

[TEŞEKKÜR i](#_Toc451856365)

[İÇİNDEKİLER ii](#_Toc451856366)

[ŞEKİLLER DİZİNİ iii](#_Toc451856367)

[TABLOLAR DİZİNİ iv](#_Toc451856368)

[SİMGELER DİZİNİ VE KISALTMALAR v](#_Toc451856369)

[SES TEMELLi MALZEME TANIMA vi](#_Toc451856370)

[VOICE RECOGNİTİON BASED MATERİALS vii](#_Toc451856371)

[1. GİRİŞ 8](#_Toc451856372)

[2. SESTEN ÖZNİTELİK ÇIKARTIMI 11](#_Toc451856373)

[2.1 Çerçeveleme 11](#_Toc451856374)

[2.2 Pencereleme 12](#_Toc451856375)

[2.3 Hızlı Fourier Dönüşümü 12](#_Toc451856376)

[2.4 Mel-Frekansına Çevirme 13](#_Toc451856377)

[2.5 Kepstrum Katsayıları 14](#_Toc451856378)

[2.6 MFCC Enerji Tespiti 15](#_Toc451856379)

[3. DESTEK VEKTÖR MAKİNELERİ 16](#_Toc451856380)

[3.1 Doğrusal Destek Vektör Makineleri 16](#_Toc451856381)

[3.2 Doğrusal Ayrılma Durumu 17](#_Toc451856382)

[3.3 Çok Sınıflı Destek Vektör Makineleri (Multiclass SVM) 20](#_Toc451856383)

[3.3.1 Bire Karşı Bir Yaklaşımı 20](#_Toc451856384)

[3.3.2 Bire Karşı Hepsi Yaklaşımı 21](#_Toc451856385)

[4. PROJENİN GERÇEKLENMESİ 22](#_Toc451856386)

[4.1 Projenin Matlab’da Gerçeklenmesi 22](#_Toc451856387)

[4.2 Projenin Gömülü Sistemde Gerçeklenmesi 23](#_Toc451856388)

[4.2.1 ST Firması 23](#_Toc451856389)

[4.2.2 Stm32f4 Discovery 23](#_Toc451856390)

[4.2.3 İde Seçimi 25](#_Toc451856391)

[4.2.4 Stm32f4 USART 26](#_Toc451856392)

[4.2.5 Stm32f4 Ses Tanıma 27](#_Toc451856393)

[SONUÇLAR 28](#_Toc451856394)

[EKLER 29](#_Toc451856395)

[KAYNAKLAR 30](#_Toc451856396)

[ÖZGEÇMİŞ 31](#_Toc451856397)

# ŞEKİLLER DİZİNİ

Şekil 1.1 Ses Tanıma Sistemin Blok Diyagramı 10

Şekil 2.1 MFCC aşamaları 11

Şekil 2.2 Ses sinyalinin çerçevelenmesi 12

Şekil 2.3 Mel-frekans süzgeç öbeği 13

Şekil 2.4 Kepstrum katsayılarının elde edilişi 15

Şekil 2.5 MFCC 1.katsayısının sesin enerji ile olan benzerliği 15

Şekil 3.1 İki sınıflı veri setini ayıran farklı düzlemlere ilişkin örnekler [6] 16

Şekil 3.2 DVM için iki sınıflı problem örneği [6] 17

Şekil 3.3 OVO Yaklaşımı [7] 21

Şekil 3.4 OVA Yaklaşımı [7] 21

Şekil 5.1 Stm32f4-Discovery kitinin üzerindeki yapılar 24

# TABLOLAR DİZİNİ

[Tablo 1.1 Ses tanıma sistemleri için geliştirilen yöntemler 9](#_Toc451118625)

[Tablo 4.1 Verilerin Test Sonuçları 22](#_Toc451118626)

# SİMGELER DİZİNİ VE KISALTMALAR

ANN Artificial Neural Network

DFT Discrete Fourier Transform

DMA Direct Memory Access

DTW Dynamic Time Warping

FFT Fast Fourier Transform

HMM Hidden Markov Model

IDFT Inverse Discrete Fourier Transform

KKT Karush-Kuhn\_Tucker

LPC Linear Predictive Coding

MFCC Mel Frequency Cepstrum Coefficient

OVA One vs All

OVO One vs One

SVM Supoort Vector Machine

VC Vapnik Chervonenkis

UART Universal Asynchronous Receiver Transmitter

USART Universal Synchronous Asynchronous Receiver Transmitter

# 

# SES TEMELLi MALZEME TANIMA

**Ender GÖREN**

**Anahtar Kelimeler:** MFCC Çıkarımı, SVM ile Sınıflandırma

**Özet:** Malzemenin tanımlana bilmesi için gelen sesin sesli ve sessiz kısımları ayırt edilmiş ve sesin bulunduğu kısım üzerinden 13 adet MFCC katsayısı çıkarılarak öznitelik vektörleri oluşturulmuştur.

MFCC katsayısılarının 1.elemanıyla sesin enerjisi arasındaki ilişki karşılaştırılmıştır. Sesin yarım saniye boyunca oluşan öznitelik vektörlerinin ortalama ve varyans hesabı yapılmış ve çıkan sonuçlar üzerinden linear SVM sınıflandırma yöntemi kullanılarak sınıflandırma yapılmıştır.

Sınıflandırmayı test etmek için one vs all (OVA) yöntemi kullanılmış ve sonuçlar kıyaslanmıştır. Sistem Matlab ortamında gerçeklenmiş ve gömülü sistem üzerine aktarılmaya ­çalışılmıştır.

# 

# VOICE RECOGNİTİON BASED MATERİALS

**Ender GÖREN**

**Keywords**: Extract MFCC, Classification with SVM

**Abstract:** The identification of the material that can be made to sound, voice and have been distinguished through quiet parts and parts that sound found 13 MFCC coefficients by subtracting the feature vectors have been created.

  The relationship between the energy of the sound of 1 element with MFCC coefficients were compared. The mean and variance calculation of the feature vector consisting of sound made during the second half and was classified using linear SVM classification method over the results.

To test one vs. all classification (OVA) was used and the results were compared. In the MATLAB environment and the system has been tried to on embedded systems..

# GİRİŞ

Bu projede geliştirilen ses tanıma siteminin sesin bulunan kısımlarında işlem yapması sağlanmıştır. Sesten öznitelik çıkartılması aşamasında sesin bulunduğu 40 çerçeve boyunca ortalama ve varyans hesabı yapılarak işlemin doğruluğu arttırılmaya ve ses yokken yapılan fazladan hesaptan arındırılmıştır. Sınıflandırma linear yapılmış olup sonuçların doğruluk yüzdeleri incelenmiştir.

Ses tanıma teknolojileri giderek önemini artırmaya devam etmektedir. Hayatı kolaylaştırmak, zamandan tasarruf etmek ve güvenliği arttırmak için ses tanıma teknolojilerinden yararlanılmaktadır. Ses tanıma teknolojisinin tarihinden kısaca bahsedersek, bu teknoloji 18.yüzyılın ikinci yarısına kadar uzanmaktadır. O sıralar ses tanımaktan çok sesin makineler ile çıkartılmasıyla uğraşılmıştır. 1773’te Rus bilim adamı Kratzenstein, kilisi orglarının boruları ve rezonans tüplerini bağlayarak sesli harfler çıkarmayı başarmıştır. Ardından Viyanalı Von Kempelen, Akustik-Mekanik ses makinesi icat etmiştir. Ses tanıma adına büyük adımlar 20.yüzyılda, birçok teknolojinin doğduğu Bell Laboratuvarı’nda atılmıştır. Homer Dudley ve Harley Fletcher konuşmada çıkan sesi tanımlamada sinyal aralığının önemli olduğu bulunmuştur. Bu noktadan sonra ses tanıma kapasitesi daha çok ve daha karmaşık işlemlere ulaşmıştır. Ses tanıma sistemleri için geliştirilen yöntemler Tablo 1.1’de verilmiştir.

Ses tanıma işleminden sonra gelen diğer bir işlem sınıflandırma problemidir. Sınıflandırma probleminin çözümüme ilişkin yapılan çalışmalar önemli yer tutmaktadır. Özellikle, bankacılık ve sigortacılık (riskli gruptaki müşterilerin tahmin edilmesi), tıp (hastalık teşhisi), biyoloji (canlı türlerinin sınıflandırılması), kimya (belirli bir hastalık için ilacın etkilerinin belirlenmesi), sosyal medya, endüstriyel üretim sistemleri (ortaya çıkan kusurlu ürünlerin belirlenmesi) gibi alanlarda sınıflandırma problemleriyle sıkça karşılaşılmaktadır. Dolayısıyla, son yıllarda sınıflandırma problemlerinin çözümü, makine öğreniminin önemli çalışma alanlarından biri olmuştur.

Son yıllarda, sınıflandırma problemlerinin çözümü için geliştirilmiş en başarılı makine öğrenimi algoritmalarından biri Destek Vektör Makineleridir. Destek Vektör Makineleri, birçok sınıflandırma probleminin çözümünde başarıyla uygulanmış ve genelleme performansı yüksek ve etkin makine öğrenimi algoritmalarından biri olarak kendisine yer ayırmıştır.

Tablo 1.1 Ses tanıma sistemleri için geliştirilen yöntemler

|  |  |
| --- | --- |
| GELİŞME | TARİH |
| Doğrusal Önkestirim Kodlama (LPC) | 1969-1970 |
| Dinamik Zaman Uydurma (DTW) | 1970’ler |
| Gizli Markov Modeller (HMMs) | 1975 |
| Mel-Frekans Cepstrum Katsayıları (MFCCs) | 1980 |
| Dil Modelleri (Language Models) | 1980’ler |
| Yapay Sinir Ağları (ANNs) | 1980’ler |
| Destek Vektör Makineleri (SVMs) | 1995, 1998 |
| Çekirdek-tabanlı Sınıflandırıcılar | 1998 |
| Dinamik Bayesian Ağları | 1999 |

Destek vektör makinelerinin en önemli avantajı, sınıflandırma problemini kareli optimizasyon problemine dönüştürüp çözmesidir. Böylece problemin çözümüne ilişkin öğrenme aşamasında işlem sayısı azalmakta ve diğer teknik/algoritmalara göre daha hızlı çözüme ulaşılmaktadır. Teknik bu özelliğinden dolayı, özellikle büyük hacimli veri setlerinde büyük avantaj sağlamaktadır. Ayrıca optimizasyon temelli olduğundan sınıflandırma performansı, hesaplama karmaşıklığı ve kullanışlılık açısından diğer tekniklere göre daha başarılıdır.

Mel kepstrum

FFT

Pencereleme

Çerçeveleme

Ses

SınıflandırıcınınTest Edilmesi

Sınıflandırıcı Tasarımı

Ortalama+Varyans Hesabı

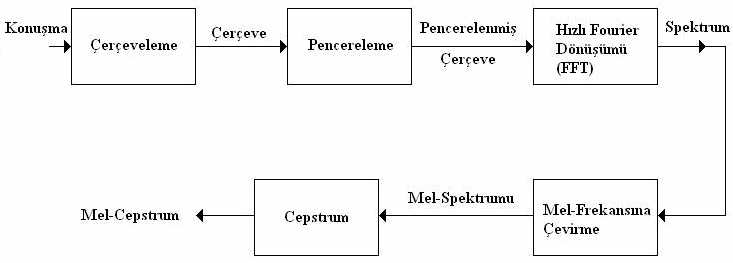
Şekil 1.1 Ses Tanıma Sistemin Blok Diyagramı

Ses verileri 168 adet olup 4 farklı sınıftan oluşmaktadır. Bu sınıflar cam, karton, metal ve plastik sınıflarıdır. Her bir ses verisi için yapılan işlem şöyledir; Ses verisi yarıya birbiri ile örtüşen 1024’lük ses verilerine ayrılmaktadır. Ses verilerinin 512 birim örtüşmesinin sebebi insan kulağının 10-20ms periyod aralığındaki sesleri duyabilmesidir. Ses verileri 44100Hz olduğundan her bir veri arası 22.67us olmaktadır. 512 birim uzunluğundaki ses verisi de 11.6ms’e eşit olmaktadır. Bir sonraki adım pencereleme aşamasıdır. Bu işlemin yapılma sebebi frekans sızıntılarını önlemek ve süreksizliği ortadan kaldırarak istenilen kısımda işlem yapmaya hazır hale getirmektir. Her bir çerçeve FFT 1024 uygulanarak ses zaman düzleminden frekans düzlemine geçirilmiş olur ve frekans düzleminde yapılmak istenilen öznitelik vektörleri çıkartılır. Çıkartılan öznitelik vektörleri 40 çerçeve boyunca tutulmaktadır. 40 tane kepstrum katsayısından ses var veya yok analizi yapılmaktadır. Ses var ise ortalama-varyans hesabı çıkartılmaktadır. Ortalama ve varyans hesabı daha doğru bir hesap yapmamıza gürültüden kaynaklı hatalı çıkabilecek sonuçlardan bizi kurtarmaktadır.

Çıkartılan ortalama ve varyans hesapları ile sınıflandırıcı oluşturulmakta ve daha sonra bu sınıflandırıcı test edilmektedir. Test edilen verilen incelendikten ve doğruluk oranlarının yeterli olduğu görüldükten sonra ses tanıma sitemi Matlab ortamından gömülü sisteme aktarılmaktadır.

# SESTEN ÖZNİTELİK ÇIKARTIMI

Mel ölçekli kepstrum katsayıları (MFCC), ses işaretini temsil eden özellikler arasında en çok bilinen özelliklerdir. MFCC özellikleri, ses işaretinin düşük frekans bileşenlerinin taşıdığı bilgi miktarının insanlar açısından yüksek frekans bileşenlerinin taşıdığı bilgi miktarına göre daha önemli olduğu temeline dayanmaktadır. MFCC özellikleri kısa dönem analizinden sonra her bir çerçeveden bu özelliklerin elde edilmesi şeklinde olmaktadır. MFCC özelliklerinin elde edilmesi sırasında izlenen adımlar Şekil 2.1.’ de gösterilmektedir.



Şekil 2.1 MFCC aşamaları

## Çerçeveleme

Bu aşamada ses sinyali üzerinde kısa zamanlı spektral analiz yapabilmek için sürekli ses sinyali, N adet örnekten oluşan çerçevelere bölünmektedir. İlk çerçeve N örnekten oluşurken ikinci çerçeve ilk çerçeveden K örnek sonra başlar ve ilk çerçevenin N-K örnek kadar üzerine binmektedir. Aynı şekilde üçüncü çerçeve ilk çerçeveden 2K örnek, ikinci çerçeveden K örnek sonra başlar ve ilk çerçevenin N-2K örnek kadar üstüne binmektedir. Bu işlem tüm ses sinyali boyunca yapılmaktadır. Aşağıdaki şekilde çerçeveleme işlemi görülmektedir. K insan ses üretim mekanizmasındaki yapıların değişim hızı göz önüne alınarak seçilir. Genel olarak pratik sistemlerde K, 10-20ms’ ye karşılık gelecek şekilde seçilmektedir.



N

K

Ses Sinyali

t

Şekil 2.2 Ses sinyalinin çerçevelenmesi

## Pencereleme

Ses sinyali içerisinde çok fazla bilgi içermeyen kısımların kesilmesi için kullanılmaktadır. Pencerelemenin amacı çerçeve başında ve sonundaki süreksizlikleri ortadan kaldırmaktır. Bu işlemde x(n) sinyali ile çerçevelenmiş sinyal w(n) çarpılmakta ve pencerelenmiş sinyal elde edilmektedir. Bu işlem ile sinyalin işlenmek istenilen bölümü ele alınır. İdeal olarak kullanılan pencerenin frekans tepkisi çok dar bir ana lobunun ve hiçbir yan lobunun olmaması gerekmektedir. Ana lobun dar olması frekans çözünürlüğünü arttırmaktadır. Yan lobların olmaması ise frekans sızıntısını engellemektedir. Böyle bir pencere pratikte mümkün değildir. Bu sebepten bu parametrelere uygun en iyi pencere seçile bilmektedir. Bu aşamada en çok kullanılan pencereleme fonksiyonlarından birisi Hamming fonksiyonudur. Hamming fonksiyonu, aşağıdaki gibi tanımlanmaktadır.

*w*(*n*) = 0.54-0.46 *cos*[2π*n/*(N-1)] , 0 ≤ n ≤ N-1 (2.1)

## Hızlı Fourier Dönüşümü

Bu aşamada, N örnekli her çerçevenin zaman bölgesinden, frekans bölgesine çevrilmesi için Hızlı Fourier Dönüşümü (FFT) uygulanmaktadır. FFT, Ayrık Fourier Dönüşümü (Discrete Fourier Transform, DFT) uygulamak için hızlı bir algoritmadır. N örnekli bir set için şöyle tanımlanmaktadır:

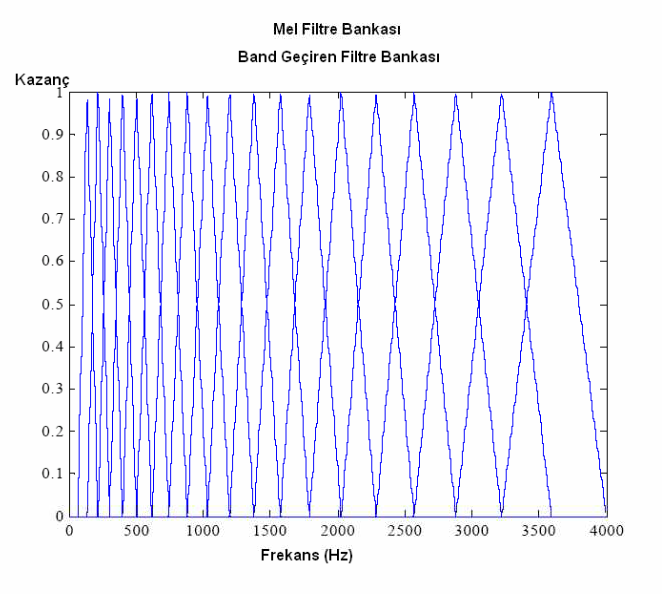
, n=0,1,2… N-1 (2.2)

## Mel-Frekansına Çevirme

Mel birimi, insan kulağının algısal özelliğini taklit edecek şekilde tasarlanmış bir birimdir. Yapılan araştırmalar, ölçümler, 1 kHz’e kadar doğrusal, daha yüksek değerlerde ise logaritmik olmaktadır. Mel ölçeği ile frekans ölçeği arasındaki dönüşüm aşağıda verilen eşitlikle sağlanmaktadır.

mel(f) = 2595 \* log (1 + f / 700) (2.3)

Bu algısal spektrumu uygulamanın bir yolu mel skalasında düzgün olarak yerleştirilmiş bir süzgeç öbeği kullanmaktır. Bu süzgeç öbeği %50 oranında birbiri üstüne getirilen, üçgen bant geçiren ve aralıkları ve bant genişliği sabit mel frekansı aralığına bağlı bir banktır. 10Hz’den 1000Hz’e kadar lineer olarak 10 filtre atanır. Frekans her iki katına çıktığında logaritmik olarak 5 filtre bu frekans aralığına atanır. Süzgeç bant genişliği 3dB noktası, mevcut dalga ile bir önceki veya bir sonraki dalganın ortasına gelecek şekilde seçilmektedir. Genellikle 20 Süzgeç kullanılmaktadır.



Şekil 2.3 Mel-frekans süzgeç öbeği

## Kepstrum Katsayıları

Sesler arasındaki farkların ve benzerliklerin nedeni insan gırtlak yapısından kaynaklanmaktadır. Kepstrum katsayılar sesler arasındaki fark ve benzerlikleri ortaya çıkarmada kullanılan yöntemlerden biridir.

Kepstrum kavramı ilk olarak 1963’de Bogert, Healy ve Tukey tarafından kullanılmıştır. Kepstrum, homomorfik sinyal işleme teknikleri içinde yer alır. Homomorfik sistemler doğrusal olmayan sistemlerin bir sınıfı olarak kabul edilmektedir. Doğrusal sistemler homomorfik sistemlerin özel bir durumudur. Homomorfik sistemler uyarım sinyalini, ses yolunun şeklinden ayırabilecek bir yol sunmalarından dolayı konuşma işleme için kullanışlı bulunmaktadır. Ayırma işlemi katlama olarak anılmakta ve aşağıdaki gibi ifade edilmektedir.

s(n) = u(n)\* h(n) (2.4)

Burada s(n)konuşma sinyali, u(n)uyarı sinyali, yani, h(n)ses yolu filtresinin dürtü tepkisidir.

Frekans bölgesindeki gösterim ise,

S(w) = U(w).H(w) (2.5)

Eğer iki tarafında logaritmasını alacak olursak eşitlik,

Log (S(w)) = Log(U(w).H(w)) (2.6)

Log (S(w)) = Log(U(w))+ Log(H(w)) (2.7)

Böylece çarpımsal ifade toplamsal bir ifadeye dönüşmüştür.

u(n)\*h(n)=Log(U(w))+Log (H(w)) (2.8)

Bu tip dönüşümler homomorfik dönüşüm olarak isimlendirilmektedir. Kepstrum, homomorfik bir dönüşümdür ve ayrık zamanda aşağıdaki gibi tanımlanmaktadır.

Konuşma sisteminde ses yolu filtresi ve uyarı sinyali zamanla değişmektedir. Bu değişimden dolayı kepstrum analizi, konuşma sinyalinin durağan olduğu kısa zamanlı aralıklarda yapılmaktadır. Bu işleme ait blok diyagram Şekil 2.3’de verilmiştir.

Pencereleme

DFT

LOG

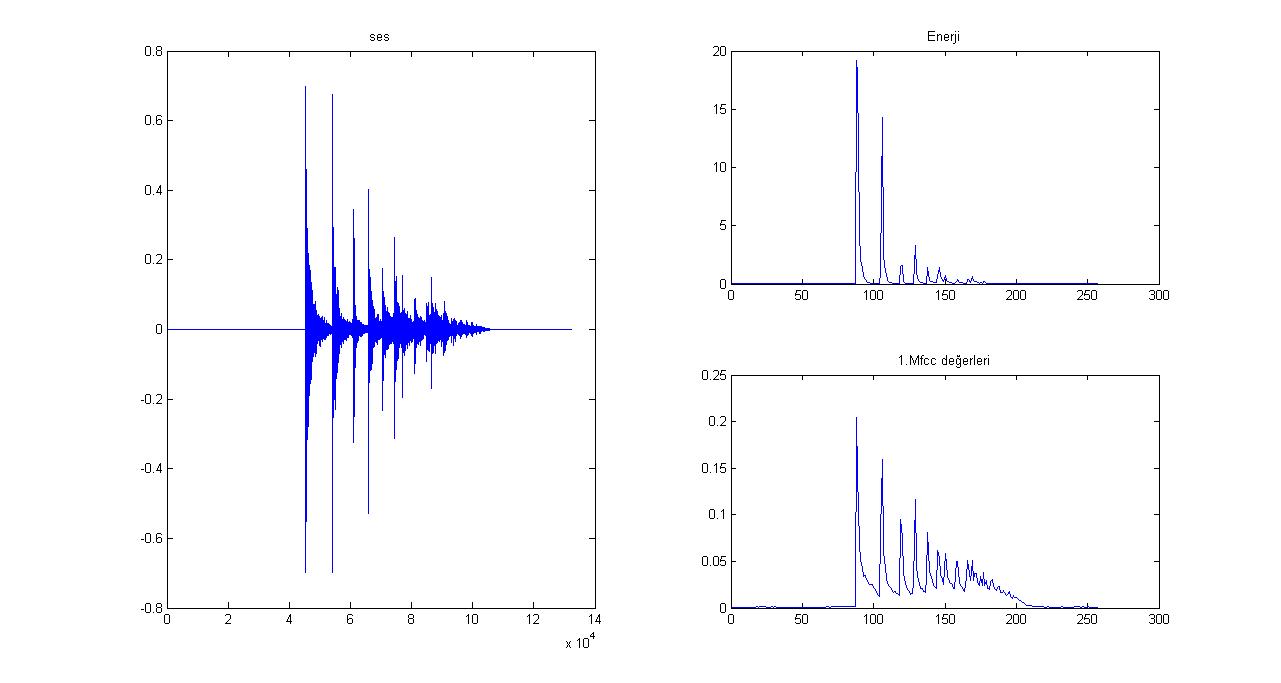
IDFT

Şekil 2.4 Kepstrum katsayılarının elde edilişi

Öncelikle ses sinyali pencerelenmekte, sonra DFT (Discrete Fourier Transform ) uygulanmaktadır. Alınan ses örneğinin frekans bileşen genliklerinin logaritması alınmaktadır. En sonunda elde edilen değerin IDFT'i alınmaktadır. Böylece ses sinyalini kepstrum değerleri elde edilmiştir.

## MFCC Enerji Tespiti

13 adet MFCC’den 1.MFCC’nin sesin enerjisi ile bağlantılı olduğu görülmüştür ve ayrı bir enerji hesabı yapmak yerine MFCC’nin 1.katsayısı kullanılmıştır. MFCC 1’in ters logaritması alınmıştır. Bu bağlantı Şekil 2.5’de gösterilmektedir.



Şekil 2.5 MFCC 1.katsayısının sesin enerji ile olan benzerliği

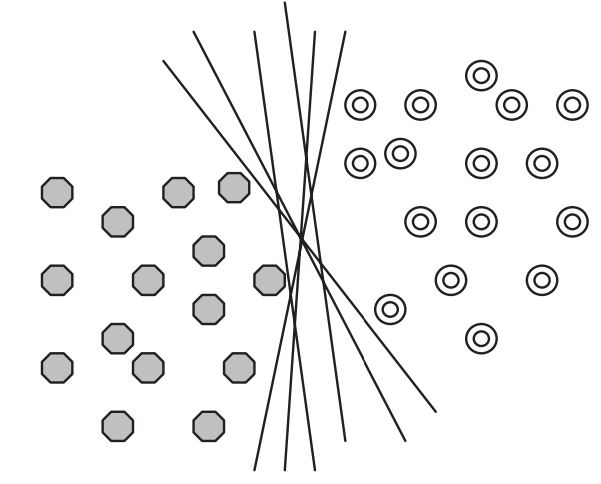
# DESTEK VEKTÖR MAKİNELERİ

## Doğrusal Destek Vektör Makineleri

Her örneği, p adet niteliğe sahip girdi, {-1,1} örneklerin ait olduğu sınıfı temsil eden çıktı ve x yüksek boyutlu girdi vektörü olmak üzere; (xi, yi) ikililerinden oluşan n hacimli bir eğitim kümesi S verildiğinde, farklı sınıflara ait örneklerin birbirinden en iyi şekilde ayıracak,

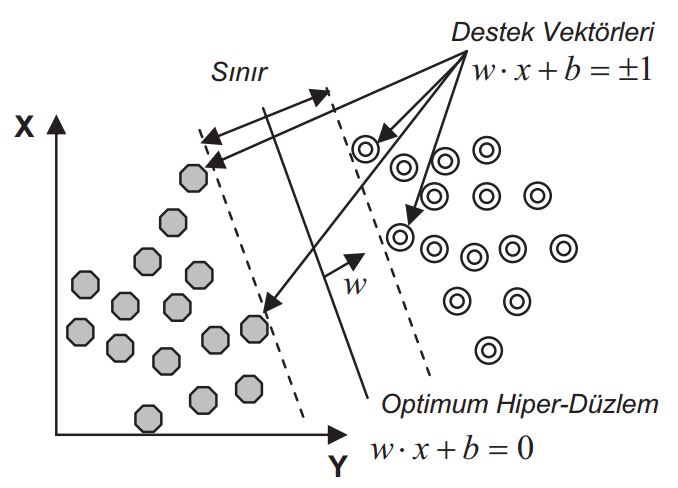
(3.1)

Doğrusal hiper düzleminin bulunmasına yardımcı olan denetimli öğrenme algoritmaları sınıfına ait makine öğrenimi algoritmasıdır. Burada w, hiper düzlemin normali aynı zamanda ağırlık vektörü ve b sabit olarak tanımlanmıştır.



Şekil 3.1 İki sınıflı veri setini ayıran farklı düzlemlere ilişkin örnekler [6]

Veri setini ayıran düzlemlere ilişkin geometrik gösterim Şekil 3.1‘de verilmiştir. Şekilde görüldüğü gibi farklı sınıflara ait örnekleri birbirinden ayıran birçok doğrusal düzlem olmaktadır. Ancak SVM, farklı sınıflara ait destek vektörleri arasındaki uzaklığı maksimize eden ayırma hiper düzleminin bulunmasını amaçlamaktadır.



Şekil 3.2 DVM için iki sınıflı problem örneği [6]

İki sınıflı ve iki boyutlu bir sınıflandırma problemi için doğrusal DVM’nin geometrik gösterimi Şekil 3.2’de verilmiştir. Destek vektörleri, ayırma hiper düzlemine en yakın olan her iki sınıfa da ait olarak ifade edilmekte ve Şekil 3.2’de gösterilmiştir. Söz konusu destek vektörleri, ait olduğu sınıfın sınırını belirler ve ayırma hiper düzlemine paralel bir düzlem üzerinde yer almaktadır. Destek vektörlerinin üzerinde bulunduğu ve kesikli çizgilerle gösterilmiş düzlemlere sınır düzlemleri denir. Sınır düzlemlerinin tam ortasından geçen ve her iki düzleme eşit uzklıkta bulunan düzlem ise hiper düzlem olarak ifade edilmektedir.

## Doğrusal Ayrılma Durumu

Eğitim veri setinin doğrusal olarak ayrılabilme durumunda SVM, en büyük sınırına sahip ayırma hiper düzlemini bulmaya çalışmaktadır. Söz konusu ayırma hiper düzleminin bulunabilmesi için veri setindeki tüm örneklerin,

(3.2)

(3.3)

Eşitsizliklerini sağlaması gerekmektedir. Bu eşitsizlikler, [3.4]’te verildiği gibi tek bir eşitsizlikte birleştirilmektedir.

(3.4)

Bir hiper düzlem, w normali ve orijinden dik uzaklığı olan, w.x+b=0 düzlemidir.

Eşitsizlik (3.2) ve (3.3) dikkate alındığında, sırasıyla, w normali ve orijinden dik uzaklığı olan düzlemi ile w normali ve orijinden dik uzaklığı olan düzlemi paralel düzlemlerdir. Dolayısıyla ve sınır düzlemleri hiper düzleme eşit uzaklıkta yer almaktadır. ve sınır düzlemleri arasında herhangi bir eğitim örneği yer almamaktadır. Ancak, düzlemlerin üzerinde var olan eğitim örnekleri destek vektörleridir ve hiper düzleme en yakın olan eğitim örnekleridir. Ayırma hiper düzlemi, her iki sınıfın destek vektörleri arasındaki uzaklığı diğer bir ifadeyle sınırı maksimum yapan ve sınırın ortasından geçen düzlemdir. Bir eğitim örneğinin hiper düzleme uzaklığı,

(3.5)

Olmak üzere, sınır değeri () eşitlik (3.6)’da verilmiştir.

Burada ||w||, ağırlık vektörü olarak adlandırılan w normal düzleminin normudur. Dolayısıyla hiper düzleme en yakın olan örneklerin hiper düzleme olan ters uzaklığı sahip olduğu ağırlık vektörünün normuna eşit olmak zorundadır. Bu teoremden yola çıkarak, eğitim örneklerinin en iyi şekilde ayıran hiper düzlem,

(3.7)

Eşitliğini en küçükleyen düzlemdir. Eşitlik (3.7)’nin en küçüklenmesi, probleme ilişkin VC boyutunun üst sınırının en küçüklenmesi anlamına gelmektedir. Böylece VC boyutunun en küçüklenmesiyle modelin yanlış sınıflandırılma olasılığı düşürmektedir.

Bu bilgiler ışığında, Eşitlik (3.7)’deki en küçükleme tipindeki optimizasyon problemi Eşitlik (3.4)’deki kısıt altında kareli optimizasyon problemi olarak Eşitlik (3.8)’de verildiği gibi formüle edilmektedir.

(3.8)

Eşitlik (8)’deki optimizasyon probleminin çözülmesi sonucunda, sınıflara ait destek vektörleri arasındaki uzaklığı maksimize edecek optimal ayırma hiper düzlemi elde edilmektedir.

SVM’de optimal ayırma hiper düzleminin bulunması için kareli optimizasyon probleminin çözümünde Lagrange çarpanlarından yararlanılmaktadır. Lagrange çarpanları en küçükleme tipindeki problemi dual probleme dönüştürerek problemin daha kolay çözülmesine olanak vermektedir. Problemin çözümünde kullanılan Lagrange fonksiyonu Eşitlik (3.9)’da verilmiştir.

(3.9)

Bu eşitlikte olmak üzere, her bir Lagrange çarpanı olarak ifade edilir. , w ağılık vektörü ve b sabitini en küçükleyen ve negatif olmayan dual değişken en büyükleyen bir fonksiyondur.

Lagrange fonksiyonunun w ve b’ye göre kısmi türevler alınarak Eşitlik (3.10) ve Eşitlik (3.11)’deki Karush Kuhn Tucker (KKT) koşulları elde edilmektedir.

Elde edilen eşitlikler Lagrange fonksiyonunda ilgili yerlere koyularak problem en büyükleme tipindeki dual Lagrange problemine (LD (α)) dönüşmektedir. Söz konusu probleme ilişkin model Eşitlik (3.12)’de gösterilmiştir.

Optimal hiper düzlemin belirlenmesi için, Eşitlik (3.12)’de verilen model çözülerek dual Lagrange LD (α)'ı maksimum yapan αi değerleri elde edilmektedir. αi Lagrange çarpanlarından sıfırdan büyük değer alan eğitim örnekler “destek vektörleri” olarak ifade edilmektedir. Optimal ayırma hiper düzlemi, sıfırdan büyük değer alan bu Lagrange çarpanları ile belirlenmektedir. αi’nin çözümü ile optimal hiper düzlemin Eşitlik (3.13) ve Eşitlik (3.14)’te verilen ağırlık vektörü w ve b sabit parametreleri belirlenmektedir.

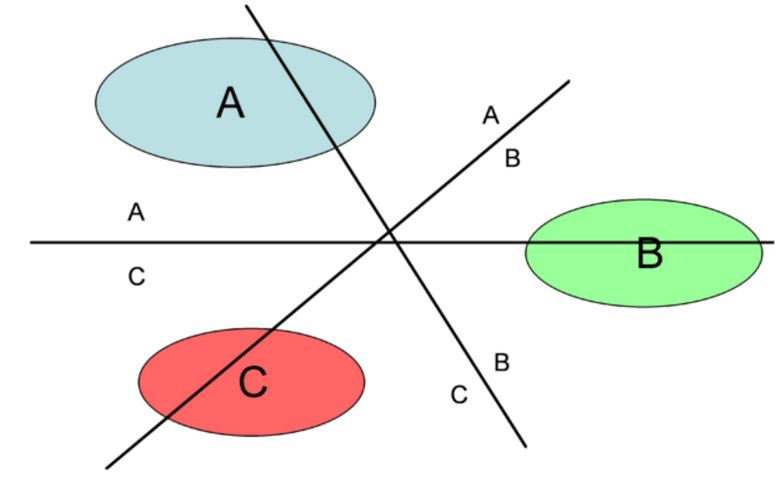
Sonuç olarak elde edilen hiper düzeleme bağlı olarak sınıflandırıcı Eşitlik (3.15)’te verilmiştir.

## Çok Sınıflı Destek Vektör Makineleri (Multiclass SVM)

Destek vektör makineleri iki sınıfın ayrılmasında kullanılmaktadır. SVM ile iki sınıf belirli bir tolerans değeri ile ayrılmaktadır. Çok sınıflı da ise sınıflandırılacak grupların ikiden fazla olmasıdır. Bu durum en çok kullanılan bire karşı bir (OVO) ve bire karşı hepsi (OVA ya da OVR) yaklaşımları kullanılmaktadır.

### Bire Karşı Bir Yaklaşımı

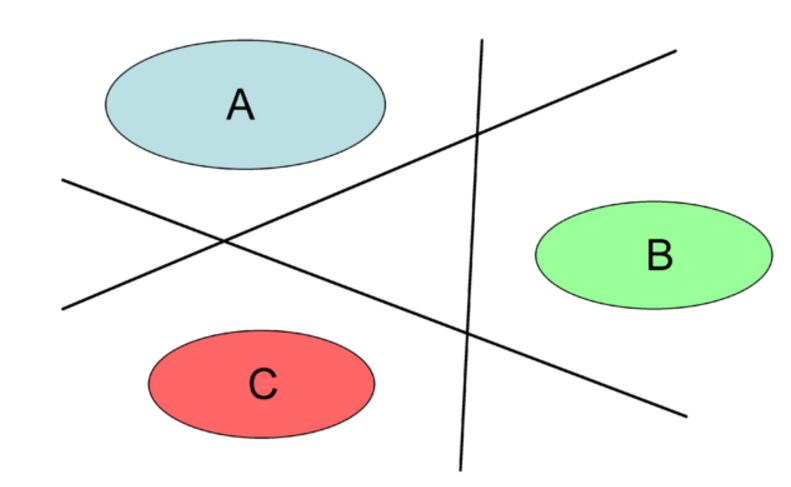
Bire karşı bir yaklaşımında sınıflandırılacak olan veri eğitilmiş sınıfların her biri ile ayrı ayrı eğitimi yapılmaktadır. Eğitimden çıkan sonuç daha yakın olduğu sınıfa göre belirlenmektedir. Yapılacak eğitim sayısı K sınıf sayısı ise K\*(K-1)/2 ile bulunmaktadır.



Şekil 3.3 OVO Yaklaşımı [7]

### Bire Karşı Hepsi Yaklaşımı

Bire karşı hepsi yaklaşımında her bir örnek veri seti bir sınıf, geriye kalan sınıfların tamamı bir sınıf olarak kabul edilmektedir. Şekil 3.4‘de görüldüğü gibi A sınıfı diğer B ve C sınıflarına karşı, B sınıfı da A ve C sınıflarına karşı, C sınıfı da A ve B sınıflarına karşı bir sorgulama yapılmaktadır.



Şekil 3.4 OVA Yaklaşımı [7]

# PROJENİN GERÇEKLENMESİ

## Projenin Matlab’da Gerçeklenmesi

Bu tezde MFCC yöntemi kullanılarak 168 adet malzeme verisinin SVM ile sınıflandırılmaya çalışılmıştır. Data olarak 40 adet cam, 44 adet karton, 34 adet metal ve 50 adet plastik verisinden yararlanılmıştır. Her veri birbiri %50 oranında üst üste binen 1024 birim uzunluğundaki çerçeveler ile işleme tabi tutulmuş ve hamming ile pencerelenmiştir. Daha sonra FFT ve MFCC hesabı yapıldığında yaklaşık yarım saniye boyunca ses var ise MFCC’ler üzerinden ortalama ve varyans hesabı yapılmakta eğer ses yok ise işlem yapılmamakta ve diğer yarım saniyelik kısma bakılmaktadır. MFCC hesabının Matlab kodlarını [10]’da bulunmaktadır.

Her veri için MFCC‘nin 13 adet ortalama ve 13 adet varyans bilgileri one vs all sınıflandırma ile SVM eğitimi yapılmakta ve sonuçlar gözlemlenmektedir. Crossvalidation ile yapılan OVA SVM sınıflandırmaya ait Matlab kodu Ekler [1]’de bulunmaktadır.

Çıkan verileri test için ve sistemi kitte gerçeklemek için destek vektörleri, alpha, bias ve destek vektörü indislerinden yararlanılmaktadır. Bu işlemin Matlab kodunu Ekler [2]‘de bulunmaktadır. Sınıflara göre doğruluk oranları Tablo 4.1’de gösterilmiştir.

Tablo 4.1 Verilerin Test Sonuçları

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **test** | **cam** | **karton** | **metal** | **plastik** |
| %10 | 96 | 94 | 88 | 76 |
| %20 | 95 | 89 | 86 | 76 |
| %25 | 95 | 89 | 86 | 75 |
| %50 | 93 | 88 | 86 | 75 |

## Projenin Gömülü Sistemde Gerçeklenmesi

### ST Firması

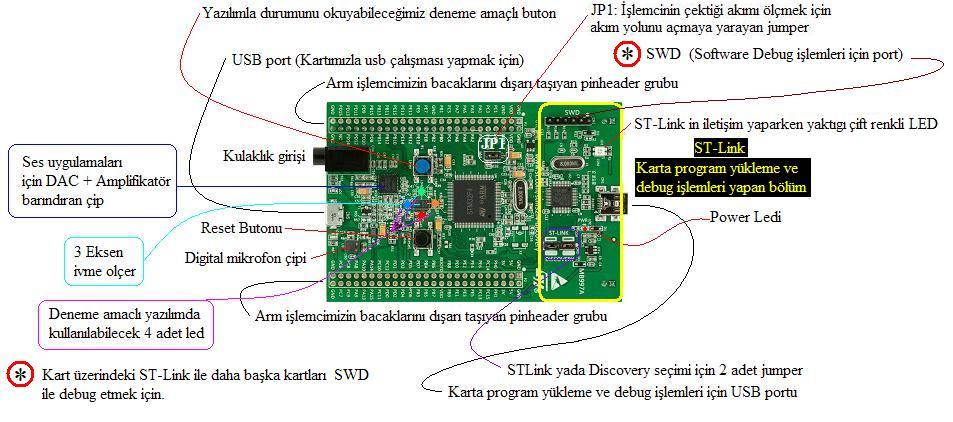
STMicroelectronics Cenevre, İsviçre merkezli bir Fransız-İtalyan uluslu elektronik ve yarı iletken üreticisidir. Genellikle ST olarak adlandırılır. Avrupa’nın en büyük yarı iletken üreticisidir. STMicroelectronics şirket merkezi ve EMEA bölgesi için merkezi Cenevre’de holding merkezli iken, STMicroelectronics NV Amsterdam Hollanda’da kayıtlıdır. ST duyu ve güç teknolojileri, otomotiv ürünleri ve gömülü işleme çözümleri ürün stratejisine odaklanmaktadır.

### Stm32f4 Discovery

Mikro denetleyicilerde günümüz ve geleceğin endüstriyel mimarisi olarak kabul edilen ARM ve uygulamalarını öğrenmek bir ayrıcalıktır. ARM’nin 32 bitlik mimarisi; yüksek hız, performans ve görsellik (TFT, GLCD uygulamalar gibi) gerektiren uygulamalarda 8-bit mimarilere göre kıyaslanmayacak üstünlüktedir.

STM32F407, ARM–M4 cortex mimarilidir. Donanım üstünlüğü, hesaplı oluşu ve çok zengin internet uygulama paylaşım platformlarından dolayı tercih edilmektedir. Üzerinde STM32F407VGT6 model işlemci bulunmaktadır. Ayrıca bir debugera ihtiyaç duyulmamaktadır. Çünkü üzerinde ST-LINK/V2 ile birlikte gelmektedir. Yani debugerı zaten kartın üzerine yerleştirilmiş durumdadır. Üretici olarak gömülü sistem tasarlayan insanlar ürünlerinin üzerine debugger yerleştirmemektedir. Çünkü ayrı bir maliyet içermekte ve kodu sadece kendileri bildiği için ürün üretimdeyken bir kez kod yüklenmekte ve sonsuza kadar o kod ile çalışmaktadır. Fakat geliştiriciler için debuggerın üzerinde bulunması, yazdığımız kodları kolayca kartımıza yüklemek için çok güzel bir özelliktir olmaktadır.

Üzerinde donanımsal olarak iki adet ST MEMS, dijital akselerometre(ivmeölçer), dijital mikrofon, bir adet “audio DAC with integrated class D speaker driver”, LED’ler, basma buton ve bir USB OTG micro-AB konnektör bulunmaktadır. Fazla sayıda ücretsiz, çalıştırılmaya hazır uygulama örnekleri bulunmaktadır ve internet üzerinden ulaşabileceğiniz birçok doküman ve program bulunmaktadır.



Şekil 4.1 Stm32f4-Discovery kitinin üzerindeki yapılar

STM32F4-DISCOVERY’ni başlıca özellikleri:

* STM32F407VGT6 mikroişlemci üzerinde 32-bit ARM Cortex-M4F çekirdek, 1 MB Flash, 192 KB RAM bulunan LQFP100 paketi
* Kart üzerinde yer alan ST-LINK/V2 kullanım modunu değiştirilebilen switch ile tek başına da kullanılabilen ST-LINK/V2 (SWD konnektör ile programlama ya da debug için)
* Kartın güç kaynağı: USB veri yolu üzerinden veya harici bir 5 V besleme gerilimi
* Harici uygulama güç kaynağı: 3 V ve 5 V
* LIS302DL veya LIS3DSH ST MEMS 3 eksenli akselerometre
* MP45DT02, ST MEMS ses sensörü, çok yönlü dijital mikrofon
* CS43L22, D sınıfı hoparlör sürücüsü ile ses DAC
* Sekiz tane LED: USB iletişim için LD1 (kırmızı / yeşil), LD2 (kırmızı) 3,3 V güç ON, Dört adet kullanıcı ledi, LD3 (turuncu), LD4 (yeşil), LD5 (kırmızı) ve LD6 (mavi), 2 USB OTG LED LD7 (yeşil) VBus ve LD8 (kırmızı) aşırı akım
* İki buton (kullanıcı ve reset)
* Mikro-AB konnektör ile USB OTG FS
* Uzatma header ile hızlı bağlantı için tüm LQFP100 I / O için başlık

### İde Seçimi

Bir STM32F4-Discovery kartı satın aldığımızda ilk yapmamız gereken debugger kurmaktır. Debugger kurulmazsa karta USB bağlantısı yaptığımızda programımızı karta download edilememektedir. STLINK V2 debugger kartımızın üzerinde dahili olarak mevcuttur ancak driverı internetten kurmamız gerekmektedir. Şayet driver kurmadan kartı takarsak taktığımız USB portundan driver kursak bile bir daha download yapılamamaktadır. Bunu çözebilmek için driver kurulumundan sonra USB bağlantısı yapıldıktan sonra Windows aygıt yöneticisinden STLINK driverını silmemiz gerekmektedir. Aygıtları yeniden taratınca download edebilmemiz mümkün hale gelmektedir.

Driver işleminden sonra Stmicroelectronics sitesinden examples, libraries vs. içeren firmware dosyası indirilmektedir. Bu klasörü çok kullandığımız bir yere almakta, yazılım olarak her şey bu dosyada olmaktadır. Satın aldığım kitle birlikte bağlantı kabloları gelmemektedir. USB - Mini USB kablosu ve USB dişi - Micro USB (A veya B) kablosunu temin etmek gerekir. USB - Mini USB kablosu zaruri olarak ihtiyaç duyulmakta çünkü kodları kitimize atıp debug etmek için bu kabloya ihtiyacımız olmaktadır.

IDE derleme, karta yükleme, hata ayıklama gibi işlemleri yapan paket yazılımlardır. STM32 ile uyumlu port edilmiş yazılımlardan birini kullanmak gerekir. İndirdiğimiz firmware dosyasındaki örnekler bir çok IDE ile projeleri hazır halde eklenmiştir. Kullanıcıya bunlardan birini seçme olanağı tanınmaktadır.

Firmware example dosyasına girdiğimizde (örnek olarak STM32F4-Discovery\_FW\_V1.1.0 ->> Project ->>) karşımıza birden fazla klasör çıkmaktadır. KEIL ve IAR en yaygın bilinen idelerden biridir. Ayrıca ilk olarak STM32'yi destekleyen ide olduğundan daha çok doküman ya da port edilmiş donanımları bulunabilmektedir. Sitesinde ücretsiz olarak iki sürümü bulunmaktadır. 30 günlük zaman kısıtlı sürüm ve 32kb kod limitli sürümüdür. ARM için olanı indirdiğinizde bunlardan birini seçilmektedir. Kurulum yapıldıktan sonra örnek projeler açılıp proje workspace dosyaları açılarak kartımıza debug download işlemleri yapılmaktadır.

Projelerin bulunduğu yerde bir de master workspace adında bir klasör bulunmaktadır. Bu klasör STM32F4-Discovery ile idemizi kullanabilmek için ayarları önceden yapılmış bir klasördür. Yeni bir proje oluşturup ayarlarını yapmak yerine bu dosya kullanılmaktadır. Yeni bir proje oluşturmak zor bir iş olduğundan, bu adımı atlamanın yollarına bakılmaktadır. IAR ve KEIL için master workspaceler vardır. Bunun sebebi bu idelerde yeni proje oluşturmak zordur. Başlangıçta bunları ayarlamak çoğunlukla sonradan bilmemiz gereken bilgiler içermektedir. IAR kullanmayı biliyorsanız yapmanız gereken ayarlar olabilmektedir. STM32F4-Discovery debugger olarak SWD kullanılmaktadır. Eğer JTAG ayarlıysa bunu değiştirmek gerekmektedir. Ayrıca işlemci ayarlı değilse bunu ayarlamak da gerekmektedir.

### Stm32f4 USART

USART(Universal Synchronous Asynchronous Receiver Transmitter) UART’ın senkronize iletişimi destekleyenidir. Normalde TX ve RX ile haberleşmeyi sağlarız ama senkron modunda bir tane de clock pini ilave edilmektedir ve datalar clock ile senkron gönderilmektedir. USART çift yönlü asenkron veya senkron haberleşme, genellikle 8 bitlik veriler kullanılmaktadır. Kullanım şekli sırasıyla start biti, data, parity biti ve stop biti şeklinde olmaktadır.

USART veri gönderme işlemi verinin en düşük değerlikli bitten en yüksek değerlikli bite doğru gönderilmektedir. TX veri göndermezken lojik 1 seviyesinde, veri gönderilirken lojik 0’dır. Stop biti veri gönderimi bittikten sonra hattı tekrar lojik 1 seviyesine çıkarmak için gönderilmektedir. Baudrate bir bitin ne kadar sürede gönderileceğini ve alınacağını belirleyen birimdir. Baudrate değerleri 1200, 2400, 4800, 9600, 19200 ve 115200 şeklindedir. Veriyi iletebilmek için alıcı taraf ve gönderen tarafın baudrate değerleri aynı olmak zorundadır.

Stm32f4 6 tane UART birimine USART 1-6 (APB2=84MHz) ve USART 2-3-4-5 (APB1=42MHz) birimlerine sahiptir. Verinin iletimi için kitteki TX dönüştürücünün RX’ine, kitteki RX ise dönüştürücünün TX’ine bağlanmaktadır.

### Stm32f4 Ses Tanıma

Stm32f4 USART 8 bitlik veri iletişimini desteklediği için float tipindeki ses verilerini kite gönderebilmek için öncelikle bu verilerin floattan integer veriye dönüştürülmesi gerekmektedir. Float sayı negatif ve pozitif olabildiğinden negatif kısmı kadar veya daha fazlası sayıya eklenmektedir. Sayının virgülden sonra 4 basmağa kadar olan verileri de olabilmesi için 1000’e bölünmektedir. Elde ettiğimiz sayı 16 bitlik olduğundan iki adet 8 bit şekline çevirmemiz gerekmektedir. Bu işlem için sayı 256’a bölünür ve tamsayı kısmı alınır, ikinci 8 bitlik veriyi bulmak için 16 bitlik veriden 256 ile ilk elde edilen 8 bitlik sayının çarpımı çıkarılmaktadır. Verileri bilgisayardan kite göndermek için realterm seri port programı kullanılmıştır. Bu program TXT dosyası ASCII kodunda gönderdiği için verileri Matlab’da ASCII koduna çevirip TXT dosyasına yazılmaktadır.

Gönderilen bilgilerin kitte Matlab’da yaptığımız işlemin tam tersi yapılarak ilk veri 256 ile çarpılır ve ikinci veri ile toplanır, daha sonra 10000’e bölünüp çıkarttığımız offset kadar eklenir ve veri geri elde edilmektedir. IAR idesinde gelen veri bir diziye atılıp watch ekranından doğru test edilmiştir. Verinin doğru geldiği görülmüştür ve Matlab’da gerçekleştirilen sistem C diline çevrilmiştir. Bu dönüşüm kısmında C kütüphaneleri ve ARM DSP Kütüphanesi [11] kullanılmıştır. ARM kütüphanesinin kullanılma sebebi kayan nokta işlemlerini daha kısa sürede tamladığı için diğer kütüphanelere göre çok daha avantajlı olmaktadır.

Gönderilen veri USART kesmesi kullanılarak işaretsiz 8 bitlik veri tipinde bir diziye atılmıştır. Dizi dolduğunda ise ses tanıma işlemi gerçekleştirilmiştir. Ses tanıma kısmında Matlab’da bulunan öz nitelik çıkartım işlemleri ve ortalama varyans kısmına kadar olan kısım C diline çevrilmiştir. Daha sonra test için Matlab’da bulunan destek vektörleri, destek vektörlerini indisleri, alpha değerleri ve biasları dizi şeklinde tanımlanmış ve Ekler [2]’nin C dilinde test edilmiştir. Çıkan sonuç 4 farklı sınıf için 4 farklı LED’e atanmıştır. Çıkan sonucun Matlab ile aynı olduğu görülmüştür.

# SONUÇLAR

Ses temelli malzeme tanıma sisteminde MFCC yöntemi kullanılarak sesin öz nitelik vektörleri bulunmuştur. Öznitelik vektörlerinden sesli bölge için ortalama ve varyans hesapları yapılmış ve bu sonuçlar ile bir sınıflandırma oluşturulmuştur. Daha sonra malzemelerin sınıflara göre başarım oranları bulunmuştur.

Sistemin gerçek zamanlı olarak gömülü sistemde çalışmasını görmek için ses sensörü aracılığıyla timer tetikleyicili adc, dma uygulaması gerçekleştirilmiştir. Fakat sistemde kullanılan ses veriler başka bir sensörden alınmış olduğundan ve bu veriler üzerinden sınıflandırıcı tasarlandığı için sistem gerçekleştirilememiştir. Ses tanıma işlemini gömülü sistemde gerçekleştirmek için ses sensörünüze ait olan verileri kullanmanız gerekmektedir.

Sistemi daha iyi duruma getirmek için sınıflandırıcıda doğrusal ayrım yerine sigmoid, paraboloid ve RBF çekirdeklerini kullanılmaktadır. MFCC hata oranlarından kurtulmak için veya daha kesin bir sonuç elde etmek için ortalama ve varyans hesabının dışında diğer başka hesaplar da yapılmaktadır. Sadece ortalama ve varyans hesabının yapılması gömülü sistemdeki kitin hızının yetersiz kalmaması için seçilmiştir.

# EKLER

[1]

for numclass=1:4 %%one vs all yaklaşımı

B=[numclass,5];

groups = ismember(class,B);

k=10; % %10 eğitim oranı

cvFolds = crossvalind('KFold', groups, k);

cp = classperf(groups);

for i = 1:k

testIdx = (cvFolds == i);

trainIdx = ~testIdx;

G1vAll=(class(trainIdx)==numclass);

svmModel = fitcsvm(data(trainIdx,:), G1vAll);

pred = predict(svmModel, data(testIdx,:));

cp = classperf(cp, pred, testIdx);

end

end

[2]

for i=1:size(data,1)

buffer = 0;

for j=1:size(SVMModel.SupportVectors,1) buffer=buffer+sum(sum(data(i,:).\*SVMModel.SupportVectors(j,:))

\*SVMModel.Alp ha(j)\*SVMModel.SupportVectorLabels(j));

end

class(i) = buffer + SVMModel.Bias;

class\_label(i) = class(i) > 0;

end

# KAYNAKLAR

1. Baygün, M. K., (2006) Türkçe Komutları Tanıyan Ses Tanıma Sistemi Geliştirilmesi, Yüksek Lisans Tezi, Pamukkale Üniversitesi Fen Bilimleri Enstitüsü, Denizli, 69s.
2. Cristianini, N. and Taylor, J. S. (2000) An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods, Cambridge University Press, s.172.
3. Demirci, M. D. (2005) Bilgisayar Destekli Ses Tanıma Sistemi Tasarımı, Yüksek Lisans Tezi, İstanbul Üniversitesi Fen Bilimleri Enstitüsü, İstanbul, 45s.
4. İskendere, Ö. 2007. İstatiksel modelleme ile konuşmacı tanıma. Doktora tezi. Uludağ Üniversitesi, 218 s. Bursa.
5. Sevgi A. (2014) “Destek Vektör Makineleriyle Sınıflandırma Probleminin Çözümü için Çekirdek Fonksiyon Seçimi” Eskişehir Osmangazi Üniversitesi İİBF Dergisi, 175s-198s
6. Taşkın K. & İsmail Ç. (2010) Destek Vektör Makineleri ile Uydu Görüntülerinin Sınıflandırılmasında Kernel Fonksiyonlarının Etkilerinin İncelenmesi, Harita Dergi Temmuz, Gebze Yüksek Teknoloji Enstitüsü, Kocaeli, s.76-77
7. Kijsirikul, B. & Ussivakul, N. (2002) Multiclass support vector machines using adaptive directed acyclic graph. Proceedings of International Joint Conference on Neural Networks (IJCNN 2002), 980-985.
8. Abe, S. (2003) Analysis of Multiclass Support Vector Machines. International Conference on Computational Intelligence for Modelling Control and Automation (CIMCA 2003), 385-396.
9. Vapnik, V. (1998). Statistical Learning Theory. Wiley-Interscience, NY.
10. Theodoros Giannakopoulos and Aggelos Pikrakis. (2014) "Introduction to Audio Analysis, A MATLAB® Approach", Academic Press, 9780080993881.
11. Cmsis Dsp Library http://www.keil.com/pack/doc/CMSIS/DSP/html/index.html

# ÖZGEÇMİŞ

1992 yılında Bolu’da doğdu. İlk, orta ve lise öğrenimini Bolu’da tamamladı. 2011 yılında Kocaeli Üniversitesi Mühendislik Fakültesi Elektronik ve Haberleşme Mühendisliği Bölümü’ne başladı. 2016 yılında mezun olma durumundadır