T.C.
İSTANBUL ÜNİVERSİTESİ-CERRAHPAŞA
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

YÜKSEK LİSANS TEZİ

Yetişkinlerden Alınan Ses Sinyallerinden Kişinin Anlık Duygusal Durumunun Derin Öğrenme Tekniği Kullanılarak Tahmini

ALI ALGHURABI

DANIŞMAN
Doç. Dr. ABDURRAHİM AKGÜNDOĞDU

Biyomedikal Mühendisliği Anabilim Dalı

Biyomedikal Mühendisliği Programı

İSTANBUL-2022

Bu çalışma, 11.02.2022 tarihinde aşağıdaki jüri tarafından Biyomedikal Mühendisliği Anabilim Dalı, Biyomedikal Mühendisliği Programında Yüksek Lisans tezi olarak kabul edilmiştir.

Tez Jürisi

Doç. Dr. **ABDURRAHİM AKGÜNDOĞDU** (Danışman)
İstanbul Üniversitesi-Cerrahpaşa
Mühendislik Fakültesi

Dr. Öğr. Üyesi Hakan SOLMAZI
Bahçeşeşir Üniversitesi
Mühandislik ve Doğa Bilimleri Fakültesi

Doç. Dr. Yasin ÖZÇELEP
İstanbul Üniversitesi-Cerrahpaşa
Mühendislik Fakültesi

Unvan ı
Bir öğe seçin.
Bir öğe seçin.

Unvan Bir öğe seçin.
Bir öğe seçin.
Bir öğe seçin.

İ. ÖNSÖZ

Yüksek lisans öğrenimim boyunca, desteğini ve bilgi esirgemeyen, bir sorunumla ilgilenip fikirleriyle çalışmamı ve çözüm bulmaya çalışan yönlendiren kıymetli 2 yıllık danışman hocam Doç. Dr. ABDURRAHİM AKGÜNDOĞDU' ya sonsuz teşekkürlerimi sunarım.

Yüksek lisans dönemi çalışmalarım boyunca her daim yanımda olan Kuzenim Serap Algorabi' ye teşekkürü bir borç bilirim.

En değerli varlığım olan aileme maddi, manevi destekleri ve güzel bir hayata sahip olmam için harcadıkları sonsuz emek ve sonsuz sevgileri için çok teşekkür ederim. Bugünlere gelmemde emeği olan tüm aileme teşekkür ederim.

Ocak 2022 ALI ALGHURABI

İİ. İÇİNDEKİLER

Sayfa No

ÖNSÖZ ……iv
İÇİNDEKİLER ……..v
ŞEKİL LİSTESİ ……...vii
TABLO LİSTESİ……..ix
SİMGE VE KISALTMA LİSTESİ …… x
ÖZET……xi
SUMMARY……xiii
1. GİRİŞ……1
2. GENEL KISIMLAR……3
2.1. Ses...…….. 5
2.1.1. Spektrum …….....7
2.1.2. Zaman Alanı ve Frekans Alanı …….....8
2.1.3. Spektrogramlar…….....8
2.1.4. Spektrogram Oluşturma …… ……...10
2.1.5. Ses Derin Öğrenme Modelleri ……..……..…….....10
2.1.6. Ses Sınıflandırması …….....…..11
2.1.7. Ses Ayırma ve Segmentasyon ……...…..11
2.1.8. Müzik Türü Sınıflandırması ve Etiketleme …… ….…… …… ..12
2.2. Özellik çıkarma ……. ……..….....13
2.2.1. Ses dosyasına genel bakış …… ……..… …..…. 14
2.2.2. Ses İşleme Uygulamaları: …… ……..… ….…. 14
2.2.3. Python Kullanarak Ses Verileri İşleme …… ……... 15
2.2.3.1. (Librosa) …… … ……..…… 15
2.2.3.2. IPython.display. …… ……..…... 15
2.2.4. Spektrogram elde edilmesi ……..... 17
2.2.5. Ses sinyalinden özellik çıkarma ……... 18
2.2.5.1. Spektral Merkez ……19
2.2.5.2. Spektral Devrilme(Spectral Rolloff)..…….....19
2.2.5.3. Spektral Bant Genişliği(Spectral Bandwidth)……..20
2.2.5.4. Sıfır Geçiş Oranı (Zero-Crossing Rate)……..…….…….21
2.2.5.5. Mel-Frekans Cepstral Katsayıları (MFCC'ler) …….……..22
2.2.5.6. Renk özelliği (Chroma feature)…….……..23
2.3. Çapraz doğrulama…….....24
2.3.1. Holdout Yöntemi ……...……...24
2.3.2. K-Katlama Çapraz Doğrulama …….....25
2.3.3. Tabakalı K-Katlama Çapraz Doğrulama …….....26
2.4. Derin Öğrenme …….....26
2.4.1. Kayıp Fonksiyonu (Loss)…….....28
2.4.2. Optimizer ……...……...…….....28
2.4.3. Devir (Epochs)…….....28
2.4.4. Küme boyutu (Batch Size)…….....29

2.4.5. Dropout……...……29
2.5. Evrişimli Sinir Ağıları …….…….29
2.5.1. Evrişimsel Katmanlar (Convolutional Layers) …….…..……..30
2.5.2. Havuzlama Katmanı (Pooling layer)……..……..…….…..30
2.5.3. Tam Bağlantılı Katman (Fully Connected Layer)…….……33
2.6. Derin Öğrenme Teknikleri……..……..……..34
2.6.1. Transfer Öğrenimi……..……..…….34
2.6.2. Veri Büyütme……...…...……...……..34
2.6.3. Softmax Sınıflandırıcı …..…….34
3.MALZEME VE YÖNTEM…….....36
3.1. Veri kümeleri ……... .. 37
3.2. Duygular…….…...39
3.3. Mel Frekans kepstral katsayıları (MFCC) ……42
3.4. Evrişimli Sinir Ağı Özelikleri …… 43
3.4.1. Hiperparametreler …….…….43
3.4.2. Aktivasyon Fonksiyonu ……43
3.4.3. Seyreltme (Dropout) Değeri ve Seyreltme Uygulanacak Katmanlar ……44
3.4.4. Optimizasyon Algoritmasının Seçimi …….…….. 45
3.4.5. Katman Sayısı ve Gizli Katmanlardaki Nöron Sayısı ……. 47
3.4.6. Evrişimli Sinir Ağı (CNN) Kernel Boyutu ……47
3.4.7. Evrişimli Sinir Ağı (CNN) "Pooling" yöntemi: max, min, mean …..…….. 48
3.5. Kullanılan yazılım…….…….……48
4. BULGULAR ……...51
4.1. CNN Değerlendirme Sonuçları…..…….…….....51
4.2. Confusion Matrix Sonuçları (Karışıklık matrisi)…….……....55
5. TARTIŞMA VE SONUÇ 57
KAYNAKLAR.....60
EKLER……...66
ÖZGEÇMİŞ.....……74

İİİ. ŞEKİL LİSTESİ

Sayfa No

Şekil 2.1: Genlik ve Zamanı gösteren basit yinelenen sinyal ……..…….5
Şekil 2.2: Karmaşık bir yinelenen sinyale sahip müzikal dalga biçimi..…….....6
Şekil 2.3: Düzenli zaman aralıklarında numune ölçümleri [3]……6
Şekil 2.4: Bir ses sinyali oluşturan frekansları gösteren spektrum ……...7
Şekil 2.5: Zaman Alanı ve Frekans Alanı [2]……..8
Şekil 2.6: Ses sinyali ve Spektrogramı ……..…….……9
Şekil 2.7: Sesli derin öğrenme modelleri tarafından kullanılan tipik ardışık düzen…….……10
Şekil 2.8: Sıradan seslerin sınıflandırılması …….……11
Şekil 2.9: Bireysel konuşmacıları bir videodan ayırma [6]…….……12
Şekil 2.10: Müzik Türü Sınıflandırması ve Etiketleme …….……..12
Şekil 2.11: Kırmızı renkte bir ses dalgaları, dijital olarak mavi renkte (örnekleme ve 4 bit nicelemeden sonra) temsil edilir ve sonuçtaki dizi sağda gösterilir. [43]…….……...14
Şekil 2.12: Bu, bir ses widget'ı döndürür……..…16
Şekil 2.13: Burada, bir dalga biçiminin genlik zarfının grafiğine sahibiz. [43]……..……16
Şekil 2.14: Ses Dalgalarının Frekansa Göre Sınıflandırılması [43]……..…...17
Şekil 2.15: Dikey eksen frekansları (0'dan 10kHz'e kadar) gösterir ve yatay eksen klibin zamanını gösterir. Tüm eylemin spektrumun altında gerçekleştiğini gördüğümüzden, frekans eksenini logaritmik bir eksene çevirebiliriz. [43]……...……..18
Şekil 2.16: burada S(k), k bin frekansındaki spektral büyüklük, f(k) k bindeki frekanstır.[43]……..18
Şekil 2.17: Başlangıçta spektral merkezde bir artış var. [47]……..……19
Şekil 2.18: librosa.feature.spectral_rolloff, bir sinyaldeki her çerçeve için rolloff frekansını hesaplar: [47]…….20
Şekil 2.19: Spektral Bant Genişliği (Spectral Bandwidth). [43]……..……..20
Şekil 2.20: librosa.feature.spectral_bandwidth sipariş-p spektral bant genişliğini hesaplar[43]……...21
Şekil 2.21: Sıfır geçiş oranını [43]…….……..21
Şekil 2.22: sıfır geçiş oranını hesapla …….....22
Şekil 2.23: sıfır geçiş oranını hesapla yakınlaştırma[43]…… ……..…..22
Şekil 2.24: Mel-Frekans Cepstral Katsayıları (MFCC'ler) [43]…… ……...23
Şekil 2.25: Renk özelliği(Chroma feature)…….....23
Şekil 2.26: Çapraz doğrulama [45]…….…… ……..…..24
Şekil 2.27: Holdout Yöntemi [45] …….25
Şekil 2.28: Yapay Nöron [46] …….27
Şekil 2.29: Yapay Sinir Ağı[50]…… ……....27
Şekil 2.30: Aktivasyon Fonksiyonu [52]…… ……..…27
Şekil 2.31: Görüldüğü gibi tipik "S" şekli nedeniyle bu şekilde adlandırılır. [49]…….…...28
Şekil 2.32: Standart sinir ağı [49]. …….…29
Şekil 2.33: Dropout yaptıktan sonra ağı……..……..…29
Şekil 2.34: Evrişimli Sinir Ağı Yapısı [48]. ……..30
Şekil 2.35: farklı filtreler uygulayarak farklı özellikleri tespit etmek mümkündür [52]. ……31
Şekil 2.36: 2D Evrişim Katmanı [48]…….…..31
Şekil 2.37: Pooling layer [48]……..…32
Şekil 2.38: havuzlama katmanı operasyon yaklaşımları [48]…….....32
Şekil 2.39: Tam Bağlantılı Katman [52]……...33
Şekil 2.40: Softmax Sınıflandırma Özellikleri [60]……..35
Şekil 2.41: Softmax Özellikleri[60]……35

Şekil 3.1: Derin öğrenme sistemini kullanarak Konuşmadaki duyguları tanımak için akış şeması tasarımı……..……..36
Şekil 3.2: RAVDESS veri seti [41]……..37
Şekil 3.3: Oluşumsal duygu teorileri [62]……40
Sekil 3.4: duyguda dalga ve spektrogram görüntüleri……41
Şekil 3.5: MFCC çıkarma işlemi [61]. ……..42
Şekil 3.6: Hiperparametreler - "düğmeler" veya "kadranlar" metaforu [61]. ……. 43
Şekil 3.7: En bilinen aktivasyon fonksiyonlarının grafikleri [61]. ……..44
Şekil 3.8: Örnek dropout uygulaması ……44
Şekil 3.9:Belli başlı optimizasyon algoritmalarının MNIST veriseti üzerindeki çalışma zamanı grafiği45
Şekil 3.10: Optimizasyon algoritmalarının hız performans gösterimi (Resim Kaynak : Alec Radford) [61]……. 46
Şekil 3.11: veri kümesi hakkında bulgular…..……..…..49
Şekil 3.12: # Eğitim verlerin yüklenmiş ……....50
Şekil 4.1: doğruluk sonuç……...52
Şekil 4.2: Eğitim kısmı %70 ve test kısmı %30…….....52
Şekil 4.3: 5 saniye konuştuktan sonraki kodun sonucu…….....53
Şekil 4.3: 5 saniye konuştuktan sonraki kodun sonucu……...53
Şekil 4.5: Etkinlik Şeması ve Kullanım Durumu Şeması ……..54

İV. TABLO LİSTESİ

Sayfa No

Tablo 4.1: RAVDESS'in veri seti eğitim-test doğruluk oranları	51
Tablo 4.2: f1-score, recall, precision and macro ortalamanın sonucunu	55
Tablo 4.3: karışıklık matrisinin (Confusion matrix) sonucu.	56

V. SİMGE VE KISALTMA LİSTESİ

Simgeler	Açıklama
kW	: Kilowatt
m	: Metre
km	: Kilometre
Hz	: Hertz frekans
MHz	: Mikrohertz frekans
MSE	: Karesel ortalama hata
GHz	: Gigahertz frekans
RMSE	: Karesel ortalama hatanın karakökü

Kısaltmalar	Açıklama
AIR	: Sesli Bilgi Alma
YSA	: Yapay sinir ağı
ASR	: Otomatik konuşma tanıma
AUC	: Eğri altındaki alan
SVM	: Destek Vektör Makinesi
CNN	: Evrişimli sinir ağı
DNN	: Derin sinir ağı
EER	: Eşit hata oranı
ER	: Hata oranı
FNN	: İleri beslemeli sinir ağı
FPR	: Yanlış pozitif oran
KNN	: K-en yakın komşu
MFCC	: Mel frekansı cepstral katsayısı
RNN	: Tekrarlayan sinir ağı
ROC	: Alıcı çalışma özelliği
SEC	: Ses olayı sınıflandırması
SED	: Ses olayı algılama
SGD	: Stokastik gradyan inişi
SIF	: Spektrogram görüntü özelliği

Vİ. ÖZET

YÜKSEK LİSANS TEZİ

Yetişkinlerden Alınan Ses Sinyallerinden Kişinin Anlık Duygusal Durumunun Derin Öğrenme Tekniği Kullanılarak Tahmini

ALI ALGHURABI

İstanbul Üniversitesi-Cerrahpaşa

Lisansüstü Eğitim Enstitüsü

Biyomedikal Mühendisliği Anabilim Dalı

Danışman : Doç. Dr. ABDURRAHİM AKGÜNDOĞDU

II. Danışman : Doç. Dr. Yasin ÖZÇELEP

Duygular, insan merkezli bilgi işlemede önemli bir rol oynamaktadır. Bu çalışma, ses dalgası biçiminden üst düzey temsilleri öğrenmek amacıyla doğrudan ham ses kaydına derin sinir ağını uygulayan uçtan uca bir derin öğrenme yaklaşımı önermektedir. Ancak, duyguların ses kaydından otomatik olarak tanınması, duygusal ifadelerin karmaşıklığı nedeni ile zor bir iştir.

Yaptığımız çalışma sonucunda, ses kaydı ile duygu tanıma için derin öğrenme modelindeki son gelişmeler ile olumlu sonuç elde etmemin mümkün olduğunu görmekteyiz. Bu gelişmeler nedeni ile duygu tanımaya yönelik son yaklaşımlar, çeşitli derin öğrenme modellerini kullanmaktadır. Bu modeller ise, gelişmiş temsiller veya spektrogram ve ses kaydının zaman-frekans gösterimi gibi önceden işlenmiş konuşma kayıtları kullanılarak oluşturulmuştur. Bu çalışmada duygu tanıma için kullanılan ileri beslemeli Derin Sinir Ağı (DNN)'nın performansında %76.79'lık bir doğruluk elde edilmiştir.

Yaptığımız çalışmalar, DNN modelinin duygu sınıflandırmasında çok iyi performans sergilediğini göstermektedir. Önerilen modelin performansı, EmoDB veri kümeleri üzerinde değerlendirilmiştir.
Bu çalışmanın amacı, ayrık konuşma duygu tanıma alanında bir araştırma sağlamaktır.
Ocak 2022, 60 sayfa.

Anahtar kelimeler: Konuşma duygu tanıma, Derin sinir ağı, Veri büyütme, Özellik çıkarma

Vİİ. SUMMARY

M.Sc. THESIS

Estimation of the Instant Emotional State of the Person from the Sound Signals Received from Adults Using Deep Learning Technique

ALI ALGHURABI

Istanbul University-Cerrahpasa

Institute of Graduate Studies

Department of Biomedical Engineering

Supervisor : Assoc. Prof. Dr. ABDURRAHİM AKGÜNDOĞDU

Co-Supervisor : Assist. Prof. Dr. Yasin ÖZÇELEP

Emotions play a prominent role in the efficient and natural human-centered computing. This work proposes a deep learning path which applies deep neural network on a raw audio recording of speech directly to learn high level representations from the audio waveform. However, automatic recognition of emotions from the speech is a difficult task due to the complication of emotional expressions. In our experiment the recent advancements in deep learning model for emotion recognition from speech we find that getting better accuracy is possible. Due to these improvements, new approaches to emotion recognition use several deep learning models. Spectrogram and time-frequency representation of speech have been used to built advanced representations or pre-processed speech recordings in deep learning models. the performance of a feed forward Deep Neural Network (DNN) for speech emotion recognition obtained an overall an accuracy of 76.79%. Our experiments show that DNN model performs very well on

emotion classification. The performance of the proposed model was evaluated on RAVDESS datasets. The goal of this study is to detecte the emotion from sound signals with deep learning.

January 2022, 60 pages.

Keywords: Speech emotion recognition, Deep neural network, Data augmentation, Feature extraction

1. GİRİŞ

Sesten duygu tanıma yöntemi, İnsan-Bilgisayar etkileşimi (HCI) için önemli bir gelişmedir [1-3]. Bu yapı, sözlü içeriği anlamaya katkı olarak geleneksel aygıtları kullanmak yerine, doğrudan sesle işbirliği yolu ile makinelerle normal iletişim ile çalışmaya yardımcı olur ve insan izleyici üyelerinin yanıt vermesini kolaylaştırır [4-6]. Bazı uygulamalar, lehçelerde iletişim için değiş tokuş çerçevelerini içerir, örneğin, çağrı odak tartışmaları, yerel olarak mevcut araç sürüş çerçevesi ve klinik uygulamalarda söylemden duygu tasarımlarının kullanımı gibi [7]. HCI yapısında (frameworks), özellikle bu yapılar laboratuvar testlerinden gerçek uygulamaya geçerken, uygun şekilde yönlendirilmesi gereken çok sayıda sorunla karşılaşılmaktadır [8-10]. Sonuç olarak, bu tür sorunlarla başarılı bir şekilde ilgilenmek ve makineler tarafından daha iyi duygu tanımayı gerçekleştirmek için azimli çalışmaya ihtiyaç vardır. İnsanların duygusal durumuna karar vermek özel bir görevdir ve herhangi bir duygu kabul modeli için bir standart olarak kullanılabilir [11]. Bu duyguların düzeni için kullanılan çeşitli modeller arasında ayrık bir duygusal metodoloji temel yaklaşımlardan biri olarak kabul edilir. Öfke, can sıkıntısı, iğrenme, şaşkınlık, korku, sevinç, mutluluk, tarafsızlık ve üzüntü gibi farklı duygular kullanılmaktadır [12, 13]. Konuşma duygu tanıma (SER) süreci, özellik çıkarma ve özellik sınıflandırma aşaması olarak bilinen iki ana aşamadan oluşur [14]. Bazı araştırmacılar kaynak tabanlı uyarma özellikleri, prosodik özellikler, ses traksiyon faktörleri ve diğer hibrit özellikler gibi konuşma işleme alanında çeşitli özellikler elde etmişlerdir [15].

Özelliklerin sınıflandırılması ise doğrusal ve doğrusal olmayan sınıflandırıcıları içeren ikinci aşamadır. Duygu tanıma için en yaygın olarak kullanılan doğrusal sınıflandırıcılardan bazıları Maksimum Olabilirlik İlkesi (MLP), Bayes Ağları (BN) ve Destek Vektör Makinesidir (SVM). Genel olarak, konuşma sinyalinin durağan olmadığı kabul edilir. Bu nedenle, doğrusal olmayan sınıflandırıcıların SER için etkin bir şekilde hareket ettiği düşünülmektedir. SER, Gizli Markov Modeli (HMM) ve Gauss Karışım Modeli (GMM) dahil olmak üzere birçok doğrusal olmayan sınıflandırıcı içerir [16]. Bunlar, temel seviye özelliklerden türetilen bilgilerin sınıflandırılması için yaygın olarak kullanılmaktadır. Mel-Frekans Cepstrum Katsayıları (MFCC), Mel Enerji Spektrumu Dinamik Katsayıları (MEDC), Doğrusal Tahmin Katsayıları (LPC) ve Algısal Doğrusal Tahmin cepstrum katsayıları (PLP) gibi enerji tabanlı özellikler, ses kaydından etkili duygu tanıma için sıklıkla kullanılmaktadır. Temel Bileşen Analizi (PCA), K-En Yakın Komşu

(KNN) ve Karar ağaçları gibi diğer sınıflandırıcılar da duygu tanıma için kullanılmaktadır [17]. Derin Öğrenme, yapay zekada yükselen bir araştırma alanı olarak kabul edilmiş ve son zamanlarda dikkatleri üzerine çekmiştir. SER için Derin Öğrenme stratejilerinin geleneksel yöntemlere göre birkaç avantajı vardır. Bunlar; manuel özellik çıkarma ve ayarlama gereksinimi olmadan yapı ve özellikleri tanımlama kapasiteleri, verilen ham bilgiden düşük seviyeli özelliklerin çıkarılmasına yönelik eğilim ve isimlendirilmemiş bilgileri yönetme kapasitesidir. Derin Sinir Ağları (DNN), girdi ve çıktı arasında bir veya daha fazla temel gizli katmandan oluşan ileri besleme yapılarına dayanmaktadır. Derin Sinir Ağları (DNN) ve Evrişimli Sinir Ağları (CNN) gibi gelişmiş besleme mimarileri, video ve görüntü işleme için etkili sonuçlar sağlamaktadır. Öte yandan, Tekrarlayan Sinir Ağları (RNN) ve Uzun Süreli Bellek (LSTM) gibi tekrarlayan mimariler, Doğal Dil İşleme (NLP) ve SER gibi konuşma tabanlı sınıflandırmada çok etkilidir [18].

Güçlü sınıflandırma yöntemlerinin yanı sıra, bu modellerin birkaç engeli vardır. Örneğin; CNN'lerin olumlu yanı, yüksek boyutlu girdi bilgilerinden özellikleri alması, diğer yandan küçük çeşitlerden ve sakatlama olayından da özellikleri alması ve dolayısı ile büyük kapasite yeteneği gerektirmesidir. LSTM tabanlı RNN'ler değişken girdi bilgileriyle ilgilenebilir ve uzun menzilli ardışık içerik bilgilerini modelledir.

Konuşulan dil aracılığıyla insan iletişimi, Bilgi alışverişinin temelidir ve ilk insan yerleşimlerinden bu yana toplumun ana yönüdür. Aynı şekilde duygular da, bugün bildiğimiz ve ilk doğal iletişim stratejisi sayılabilecek konuşma dilinden önceki ilkel bir içgüdüye kadar uzanır. Proje, konuşmacının konuşurken ortaya çıkardığı duyguların tespiti ile ilgilidir. Bu projede, RAVDESS adı verilen 8 farklı duygu sesi karakterine sahip bir veri seti kullanılmış ve Mel-frequency cepstrum (MFCC) elde etmek için python'daki (Librosa) kütüphanesi kullanılarak bu veri kümesinden öznitelik çıkarımı uygulanmıştır. Softmax sınıflandırmasını kullanarak sesin 8 duygusal karakterini algılayabilen bir derin öğrenme sistemi oluşturmak için MFCC öznitelik çıkarımına Evrişimli Sinir Ağı (CNN) yöntemi uygulanmıştır.

2. GENEL KISIMLAR

Konuşma, bilgi aktarımının ana ve doğrudan yoludur. Çok çeşitli bilgiler içerir ve bu duygular aracılığı ile zengin duygusal bilgileri ifade edebilir ve nesnelere, sahnelere veya olaylara tepki olarak görselleştirebilir [19]. Duyguları konuşarak tanımak, konuşma işlemede ve insan-bilgisayar etkileşimine dayalı uygulamalarda aktif araştırma temalarından biri haline gelmiştir. Bu tez, insan konuşmasından duyguları tanıma üzerine deneysel bir çalışma yürütmektedir. Deneyler için dikkate alınan duygular arasında nötr, sakin, mutlu, üzgün, kızgın, korkulu, iğrenme ve şaşırma yer almaktadır. Derin öğrenme yöntemleri gibi modern makine öğrenimi tekniklerinin, gerçek yaşam koşullarında sağlam performansa sahip SED sistemleri geliştirmek için kullanılıp kullanılamayacağını araştırmaktayız [19].

Bu tez, öznitelik çıkarma ve sınıflandırma motoru olarak adlandırılan iki aşamalı bir yaklaşımla konuşma sinyallerine dayalı bir duygu tanıma sistemi önermektedir. İlk yaklaşımda, iki özellik grubu incelenir; ilk olarak, Mel Frekans Cepstral Katsayıları (MFCC), Sıfır Geçiş Hızı (ZCR), Harmonik Gürültü Hızı (HNR), Tonnetz katsayıları ve Spektral-Kontrast dahil olmak üzere ses özelliklerinin boyutsal bir vektörü çıkarılmaktadır. İkinci olarak da, daha önce çıkarılan parametrelerden ilgili parametrelerin seçimi için Auto-Encoder yönteminin kullanılması önerilmektedir.
İkinci yaklaşımda, bir sınıflandırıcı yöntemi olarak Derin Sinir Ağlarını (DNN) kullanılmıştır.

İnsan-bilgisayar etkileşiminde duyguların incelenmesi son yıllarda artmıştır. Bunun nedeni, kullanıcıların duygusal durumunu tespit edebilen bilgisayar uygulamalarına artan taleptir [20]. İnsanlar arasındaki günlük etkileşimden hareketle, bu alandaki araştırmaların büyük bir kısmında yüz ve ses bilgilerinden duyguları tespit etme yer almaktadır. Kontrollü durumlarda bu tür bilgilere dayanan mevcut duygu tespit bilgisayar sistemleri, duyguları önemli bir doğrulukla sınıflandırabilir [21]. Ancak duygular her zaman yüz ifadeleri ve sesli bilgiler ile kendini göstermez. Psikologlar, fizyolojik uyarılma, davranışsal ifade ve duyguların bilinçli deneyimi arasında ayrım yapmaktadır. Yüz ve ses bilgisi, yalnızca bilinçli olarak kontrol edilebilen ve değiştirilebilen ve yorumlamanın genellikle öznel olduğu davranışsal bir ifadeyle ilgilidir. Bu nedenle, kalp hızı, cilt iletkenliği ve göz bebeği genişlemesi gibi farklı fizyolojik bilgilere odaklanan duyguyu tespit etmek için başka yaklaşımlar önerilmiştir [21]. Duygusal beyin-bilgisayar etkileşiminde nispeten yeni bir araştırma alanı olan elektroensefalogramları

(EEG) kullanılarak duygular tespit edilmeye çalışılmaktadır [19, 22, 23]. Duygular sinir sistemi üzerinde doğrudan bir etkiye sahip olduğu için kalp atış hızı da duygulardan etkilenmektedir. Bu durumda kişinin duygusal durumu hakkında bilgi almak için bir kişinin kalp atış hızı da ölçülebilir [22,23].

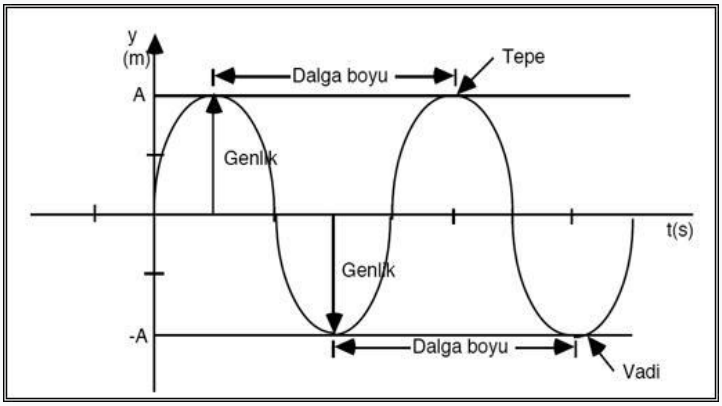
Kalp atış hızı da konuşmayı etkilediğinden, konuşma sinyallerinin de konuşmacının kalp atış hızını temsil ettiğini belirtmektedir. [24]'teki çalışmada, negatif duyguya neden olan negatif bir uyaran varsa, kalp atış hızının pozitif uyaranlara göre daha aktif olarak yavaşladığını söylemektedir. İnsanları makinelerden ayıran özelliklerden biri, başkalarının duygularını tam olarak yorumlamak zor olsa da, bu duyguları anlama yeteneğinin olmasıdır.

Duyguları konuşma yoluyla tespit etmek için birçok yöntem vardır, farklı yöntemler farklı ses parametrelerini kullanır; perde, frekans, tını ve daha fazlası. Farklı parametrelerin değerleri arasında ayrım yapılarak duygu tespit edilebilir. Bu parametreler konuşmadan duyguyu tespit etmede hayati bir rol oynar, bu parametrelerdeki değişiklikler duyguda bir değişikliğe neden olur. Duygu tespiti basit bir sınıflandırma problemi değildir; duyguların ifadesi sadece kültürden kültüre değil, aynı kültür içinde kişiden kişiye de değişebilmektedir. Aynı konuşma, konuşmanın bağlamına bağlı olarak farklı duygularla iletilebilir. Bu nedenle, uygun konuşma özelliklerinin çıkarılması, duygu tanıma sistemlerinde çok önemli bir adımdır. Mel Frekans Cepstrum Katsayısı (MFCC) ve Doğrusal Tahmin Cepstrum Katsayısı (LPCC) özellikleri, etkinlikleri nedeniyle otomatik konuşma tanıma (ASR) sistemlerinde yaygın olarak kullanılmaktadır [25]. Bununla birlikte, bu özelliklerin duyguları sınıflandırmada etkili olup olmayacağı hala tartışma konusudur. Bazı çalışmalarda konuşma sinyallerinden uygun öznitelikleri çıkarmak için Gauss Karışımları Modeli (GMM) ve Gizli Markov Modeli (HMM) kullanılmıştır. Bununla birlikte araştırmacılar, duygu tespiti için konuşma özelliklerinin seçiminde bir standart ortaya koyamamaktadır. İyi bir duygu algılama sistemi için bir diğer önemli görev de sınıflandırıcının seçimidir. Recurrent Neural Network (RNN) kullanılan yaklaşımlardan biridir [26]. Başka bir yaklaşımda, Wieman ve ark. [27] İkili Karar Ağacının en alakalı özellikleri belirleyebildiğini bulmuştur. Ancak deneyleri küçük bir veri setinde yapılmıştır. Başka yaklaşımlarda da, birleşik sistemler önerilmiştir [28-30]. Çalışmamız, duygu tespiti için yaygın olarak kullanılan Mel Frekans Cepstrum Katsayısı (MFCC) olarak bilinen bir ses işleme tekniğine odaklanmaktadır [31, 32]. Mel Frekans Cepstrum (MFC), Mel Frekans

Cepstrum Katsayıları (MFCC)'nın toplu olarak bir MFC oluşturduğu herhangi bir sesin kısa vadeli güç spektrumudur. Ses verilerinin kısa vadeli güç spektrumu özelliği, insanların sesi nasıl algıladığına benzemektedir. MFCC'nin bu özel karakteristiği, duygu tespiti amacıyla arzu ettiğimiz özelliğimizdir. Farklı duyguları sınıflandırmak için Lojistik Model Ağacı (LMT) olarak bilinen bir sınıflandırıcı ile girdi olarak 13 boyutlu MFCC özellikleri kullanılmıştır. LMT, yapraklarında doğrusal regresyon modelleri bulunan bir sınıflandırıcıdır [33]. Bir çoğunluk oylama mekanizması uygulayarak tam bir ifadenin duygusunu etiketlemek için çıkarılan kılavuz düzeyindeki özellikler kullanılmıştır. Araştırmacıların çoğu [28], tam sinyali sınıflandırmak için kılavuz özellikleri üzerinde derin öğrenme algoritmaları uygulayarak tam ifadenin istatistiksel bilgilerini çıkarma eğilimindedir. Öte yandan, bazı çalışmalar [34] daha iyi sonuç elde etmek için yüz ifadelerinden yardım almayı içermektedir.

2.1. Ses

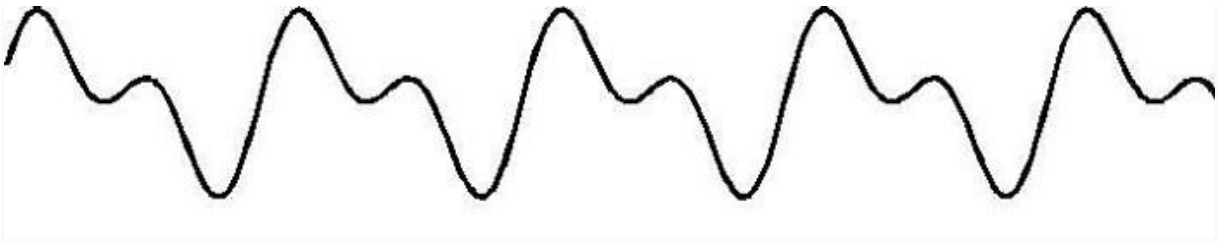
Hepimiz okuldan, hava basıncındaki değişikliklerden bir ses sinyalinin üretildiğini hatırlıyoruz. Basınç değişimlerinin yoğunluğunu ölçebilir ve bu ölçümleri zaman içinde çizebiliriz. Ses sinyalleri, her dalganın aynı şekle sahip olması için genellikle düzenli aralıklarla tekrarlanır. Yükseklik, sesin yoğunluğunu gösterir ve genlik olarak bilinir (Şekil 2.1).



Şekil 2.1: Genlik ve Zamanı gösteren basit yinelenen sinyal [35].

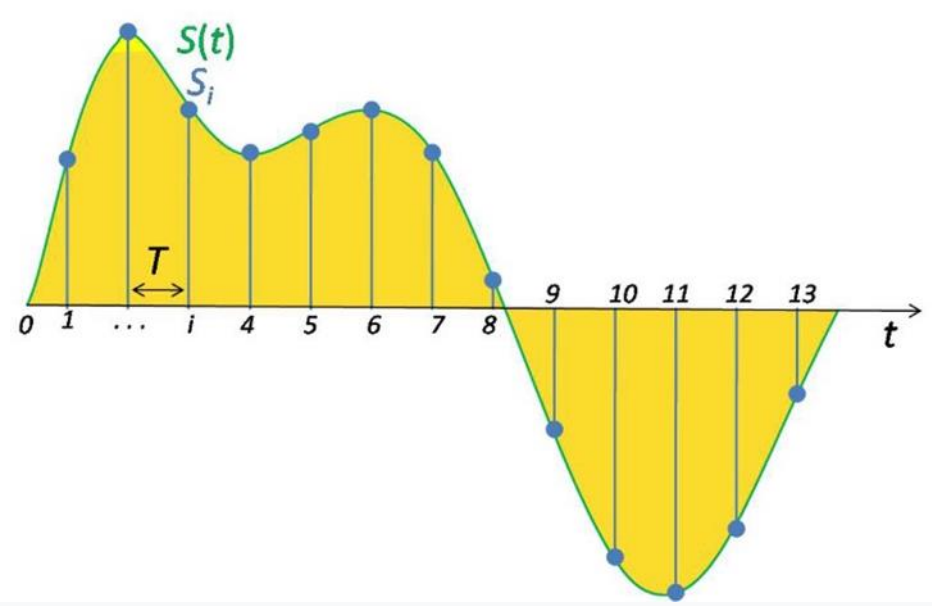
Sinyalin bir tam dalgayı tamamlaması için geçen süre periyottur. Sinyalin bir saniyede yaptığı dalga sayısına frekans denir. Frekans, periyodun tersidir. Frekans birimi Hertz'dir.

Karşılaştığımız seslerin çoğu bu kadar basit ve düzenli periyodik kalıpları takip etmeyebilir. Ancak, daha karmaşık yinelenen desenlere sahip bileşik sinyaller oluşturmak için farklı frekanslardaki sinyaller bir araya toplanabilir. Kendi insan sesimiz de dahil olmak üzere duyduğumuz tüm sesler, bunun gibi dalga biçimlerinden oluşur. Örneğin, Şekil 2.2'de gösterilen bir müzik aletinin sesidir.



Şekil 2.2: Karmaşık bir yinelenen sinyale sahip müzikal dalga biçimi [35].

İnsan kulağı, tını olarak da bilinen sesin 'kalitesine' dayalı olarak farklı sesleri ayırt edebilir. Bir ses dalgasını sayısallaştırmak için sinyali bir dizi sayıya çevirmeliyiz ki onu modellerimize girebilelim. Bu, sesin genliğini sabit zaman aralıklarında ölçerek yapılır [35].



Şekil 2.3: Düzenli zaman aralıklarında numune ölçümleri [37].

Bu tür her ölçüme örnek adı verilir ve örnekleme hızı saniyedeki örnek sayısıdır (Şekil 2.3). Örneğin, yaygın bir örnekleme hızı saniyede yaklaşık 44.100 örnektir. Bu, 10 saniyelik bir müzik klibinde 441.000 örnek olacağı anlamına gelir.

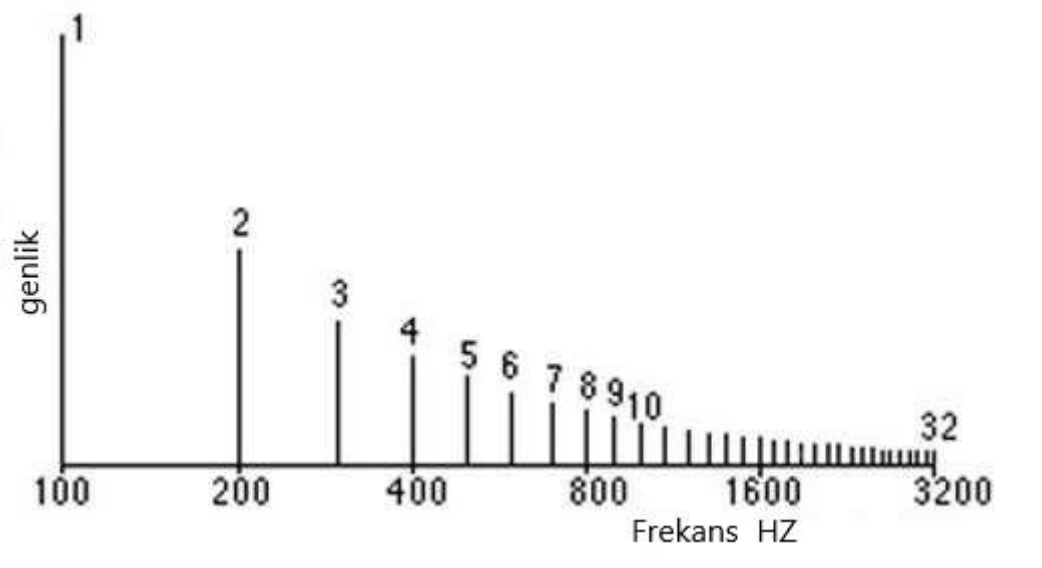
Derin öğrenme modeli için ses verilerinin hazırlanması: Birkaç yıl öncesine kadar, Derin Öğrenmeden önceki günlerde, Computer Vision'ın makine öğrenimi uygulamaları, özellik mühendisliği yapmak için geleneksel görüntü işleme tekniklerine güveniyordu. Örneğin, köşeleri, kenarları ve yüzleri algılamak için algoritmalar kullanarak el yapımı özellikler üretilirdi. NLP uygulamalarıyla da, N-gramları çıkarma ve Terim Frekansını hesaplama gibi teknikler kullanılırdı [35].

Benzer şekilde, ses makinesi öğrenme uygulamaları, özellikleri çıkarmak için geleneksel dijital sinyal işleme tekniklerine bağlıydı. Örneğin, insan konuşmasını anlamak için ses sinyalleri,

fonemler gibi öğeleri çıkarmak için fonetik kavramlar kullanılarak analiz edilebilir. Tüm bunlar, bu sorunları çözmek ve sistemi daha iyi performans için ayarlamak için çok fazla alana özel uzmanlık gerektiriyordu. Bununla birlikte, son yıllarda Derin Öğrenme giderek daha yaygın hale geldikçe, ses işlemede de muazzam bir başarı elde etti. Derin öğrenme ile geleneksel ses işleme tekniklerine artık ihtiyaç yoktur ve çok sayıda manuel ve özel nesil özellik gerektirmeden standart veri hazırlığına güvenebiliriz. Daha da ilginç olan, derin öğrenme ile ses verileriyle aslında ham haliyle ilgilenmememizdir. Bunun yerine, kullanılan yaygın yaklaşım, ses verilerini görüntülere dönüştürmek ve ardından bu görüntüleri işlemek için standart bir CNN mimarisi kullanmaktır [35].

2.1.1 Spektrum

Gerçek dünyada meydana gelen herhangi bir sesi temsil eden bileşik sinyaller oluşturmak için farklı frekanslardaki sinyaller bir araya toplanabilir. Bu, herhangi bir sinyalin birçok farklı frekanstan oluştuğu ve bu frekansların toplamı olarak ifade edilebileceği anlamına gelir. Spektrum, bir sinyal üretmek için bir araya getirilen frekanslar kümesidir. Örneğin. Resim bir müzik parçasının spektrumunu göstermektedir. Spektrum, sinyalde bulunan tüm frekansları, her frekansın gücü veya genliği ile birlikte çizer (Şekil 2.4).

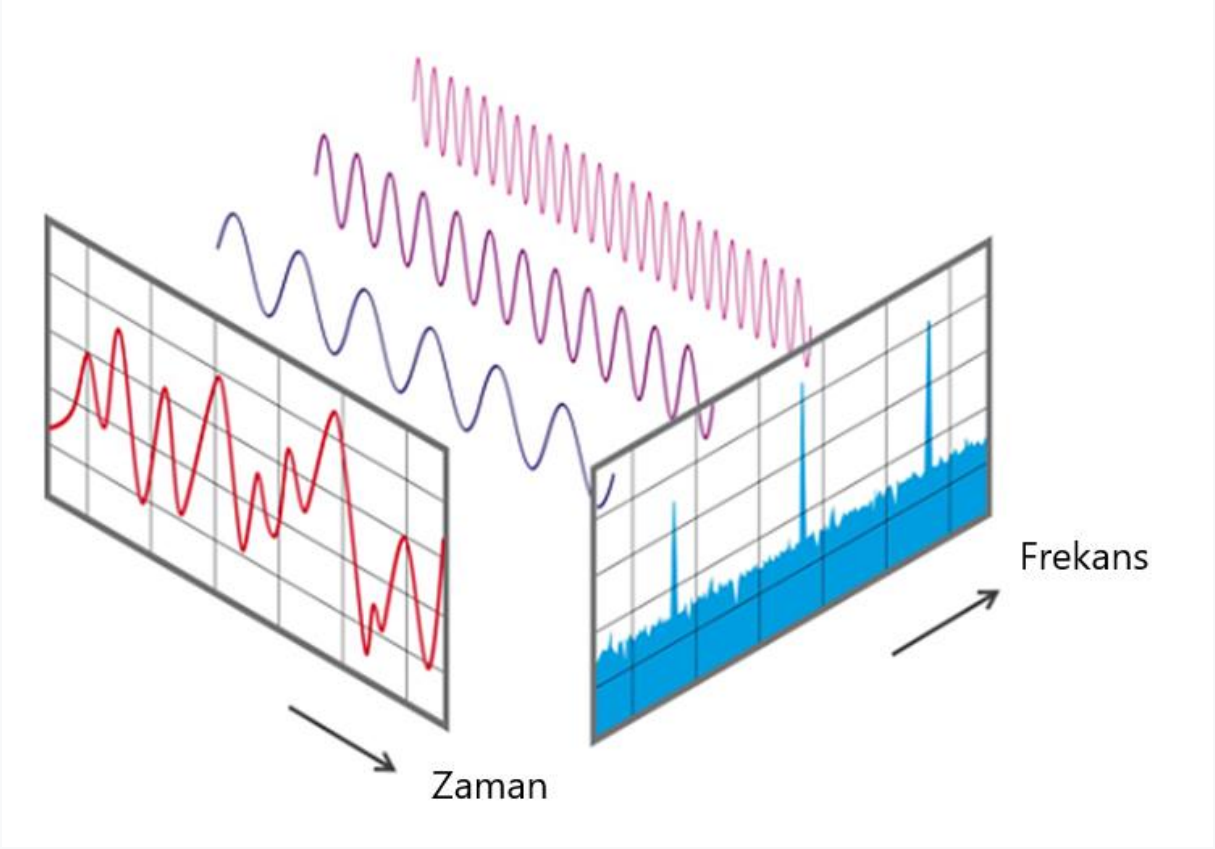


Şekil 2.4: Bir ses sinyali oluşturan frekansları gösteren spektrum [35].

Temel frekans olarak adlandırılan bir sinyaldeki en düşük frekans. Temel frekansın tam sayı katları olan frekanslar harmonik olarak bilinir. Örneğin, temel frekans 200 Hz ise, harmonik frekansları 400 Hz, 600 Hz vb.'dir.

2.1.2 Zaman Alanı ve Frekans Alanı

Daha önce zamana karşı genliğini gördüğümüz dalga biçimleri, bir ses sinyalini temsil etmenin bir yoludur. X ekseni sinyalin zaman değerlerinin aralığını gösterdiğinden, sinyal zaman alanında izlenir. Spektrum, aynı sinyali temsil etmenin alternatif bir yoludur. Frekansa karşı Genlik gösterir ve x ekseni sinyalin frekans değerlerinin aralığını gösterdiğinden, zamanda bir anda sinyali Frekans Alanında izlenir (Şekil 2.5).



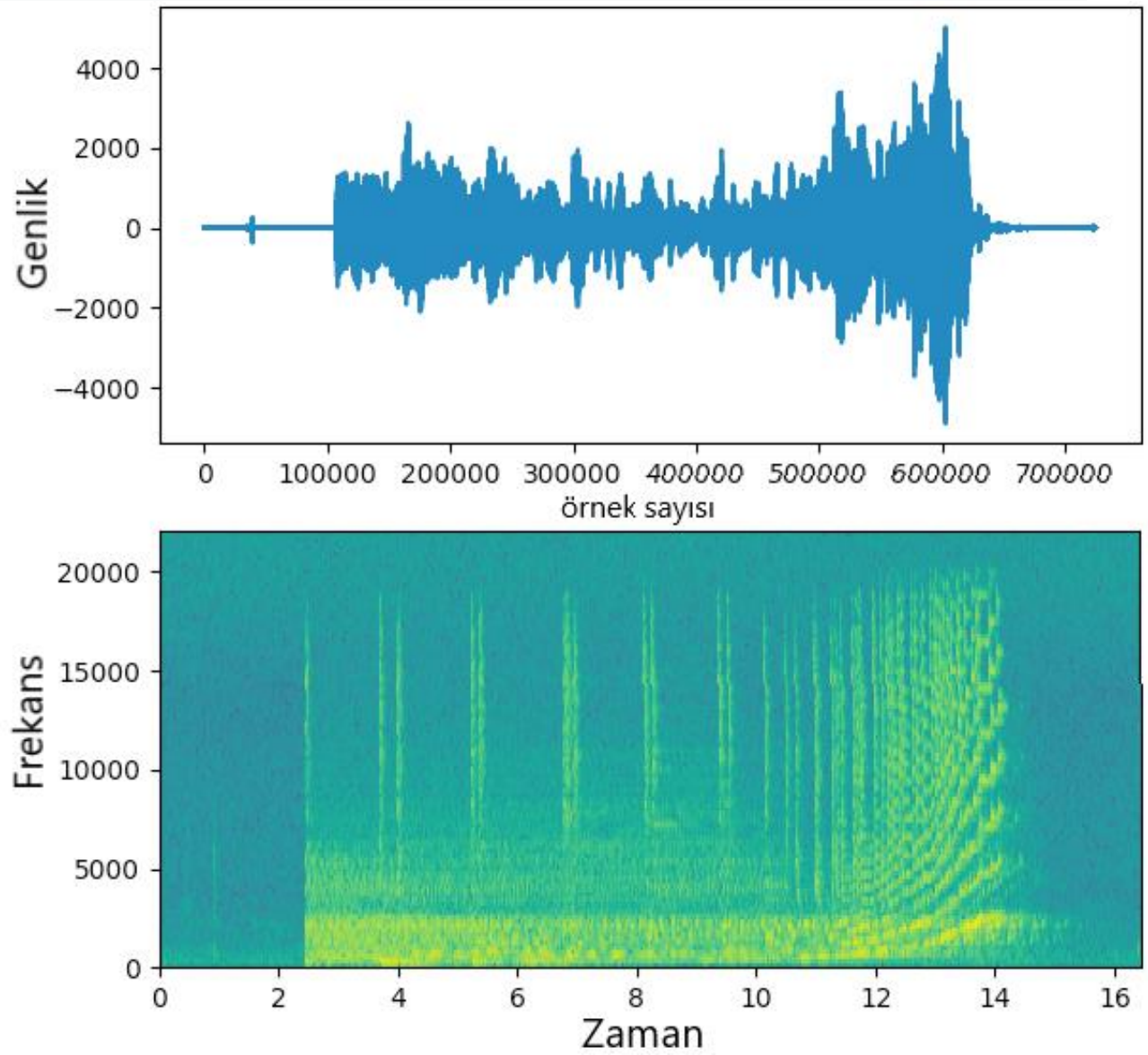
Şekil 2.5: Zaman Alanı ve Frekans Alanı [34].

2.1.3 Spektrogramlar

Bir sinyal zamanla değiştikçe farklı sesler ürettiğinden, onu oluşturan frekanslar da zamanla değişir. Başka bir deyişle, Spektrumu zamanla değişir. Bir sinyalin Spektrogramı, Spektrumunu zaman içinde çizer ve sinyalin bir "fotoğrafı" gibidir. Zamanı x ekseninde ve Frekansı y ekseninde çizer. Sanki Spectrum'u zaman içinde farklı durumlarda tekrar tekrar almış ve sonra hepsini tek bir olay örgüsü içinde birleştirmişiz gibi. Her frekansın Genliğini veya gücünü belirtmek için farklı renkler kullanır. Renk ne kadar parlaksa, sinyalin enerjisi o kadar yüksek

olur. Spektrogramın her dikey dilimi, esasen o andaki sinyalin Spektrumudur ve sinyal gücünün o anda sinyalde bulunan her frekansta nasıl dağıldığını gösterir [35].

Şekil 2.6'da üstteki resim, sinyalin genliğinin zaman alanındaki değerleri herhangi bir zamanda bu klibin ne kadar gürültülü veya sessiz olduğu konusunda bir fikir verir, ancak hangi frekansların mevcut olduğu hakkında bize çok az bilgi verir.



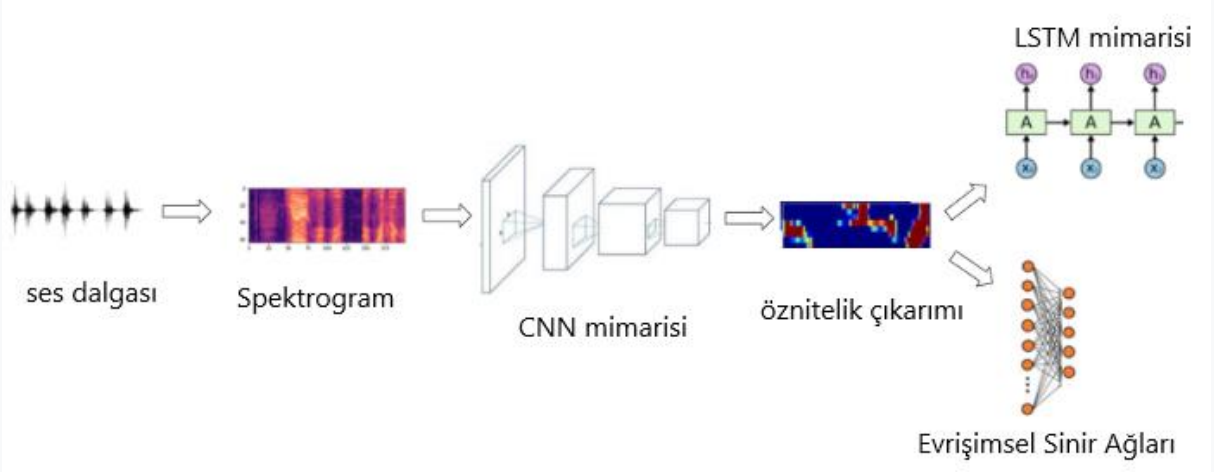
Şekil 2.6: Ses sinyali ve Spektrogramı (İkinci resim Spektrogramdır ve sinyali Frekans alanında gösterir) [4].

2.1.4 Spektrogram Oluşturma

Spektrogramlar, herhangi bir sinyali kurucu frekanslarına ayrıştırmak için Fourier Dönüşümleri kullanılarak üretilir. Bizim için tek bir adımda spektrogram oluşturabilen çok kullanışlı Python Kütüphanesi fonksiyonları vardır [38].

2.1.5 Ses ile Derin Öğrenme Modelleri

Spektrogram, bir şekilde sinyalin "parmak izi" gibi, bir ses sinyalinin eşdeğer bir kompakt temsilidir ve ses verilerinin temel özelliklerini bir görüntü olarak yakalamanın zarif bir yoludur [38].



Şekil 2.7: Sesli derin öğrenme modelleri tarafından kullanılan tipik ardışık düzen [38].

Bu nedenle, çoğu derin öğrenme ses uygulaması, sesi temsil etmek için Spektrogramları kullanır. Genellikle şöyle bir prosedür izlenir (Şekil 2.7):

• Wave dosyası biçimindeki ham ses verileri alınır.

• Ses verileri ilgili spektrograma dönüştürülür.

• İsteğe bağlı olarak, spektrogram verilerini artırmak için basit ses işleme teknikleri kullanılabilir. (Spektrogram dönüştürmeden önce ham ses verilerinde bir miktar büyütme veya temizleme de yapılabilir)

• Artık görüntü verisine sahip olunduğuna göre, bunları işlemek ve spektrogram görüntüsünün kodlanmış bir temsili olan özellik haritalarını çıkarmak için standart CNN mimarileri kullanılabilir. Sonraki adım, çözmeye çalışılan soruna bağlı olarak bu kodlanmış gösterimden çıktı tahminleri oluşturmaktır.

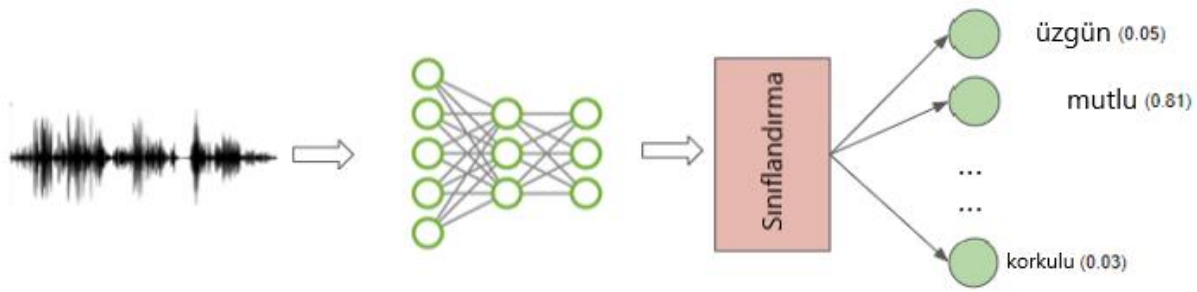
• Örneğin, bir ses sınıflandırma problemi için, genellikle tamamen bağlantılı bazı doğrusal katmanlardan oluşan bir sınıflandırıcıdan geçirilebilir [38].

• Bir konuşmadan metne geçiş için, bu kodlanmış gösterimden metin cümleleri çıkarmak için bazı RNN katmanları kullanılabilir [38].

Aşağıdaki kısımlarda, tüm bu adımlar ve kullanılan mimariler hakkında çok daha fazla ayrıntıya girilecektir. Günlük yaşamdaki ses verileri, insan konuşması, müzik, hayvan sesleri ve diğer doğal sesler gibi sayısız biçimlerde olabileceği gibi, arabalar ve makineler gibi insan faaliyetlerinden kaynaklanan insan yapımı sesler de olabilir. Seslerin hayatımızdaki yaygınlığı ve ses türlerinin çeşitliliği göz önüne alındığında, sesi işlememizi ve analiz etmemizi gerektiren çok sayıda kullanım senaryosu olması şaşırtıcı değildir. Artık derin öğrenme çağı geldiğine göre, bir dizi kullanım örneğini çözmek için uygulanabilir [39].

2.1.6 Ses Sınıflandırması

Bu, en yaygın kullanım durumlarından biridir ve bir ses almayı ve onu birkaç sınıftan birine atamayı içerir. Örneğin, görev, sesin türünü veya kaynağını belirlemek olabilir. Açıkçası, olası uygulamalar çok geniştir. Bu, ürettiği sese dayalı olarak makine veya ekipmanın arızasını tespit etmek için veya bir gözetim sisteminde güvenlik ihlallerini tespit etmek için uygulanabilir, Sıradan seslerin sınıflandırılması (Şekil 2.8).

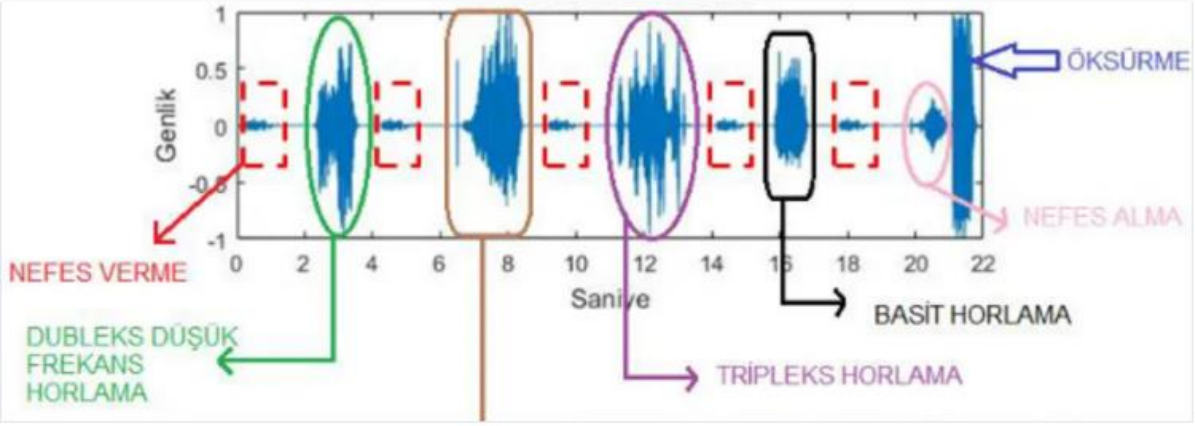


Şekil 2.8: Sıradan seslerin sınıflandırılması [39].

2.1.7 Ses Ayırma ve Segmentasyon

Ses ayırma, daha sonraki işlemler için kullanılabilecek şekilde, bir sinyal karışımından ilgilenilen bir sinyalin izole edilmesini içerir. Örneğin, bireysel insanların seslerini birçok arka plan gürültüsünden veya keman sesini müzik performansının geri kalanından ayırmak isteyebilirsiniz. Ses segmentasyonu, ses akışından ilgili bölümleri vurgulamak için kullanılır.

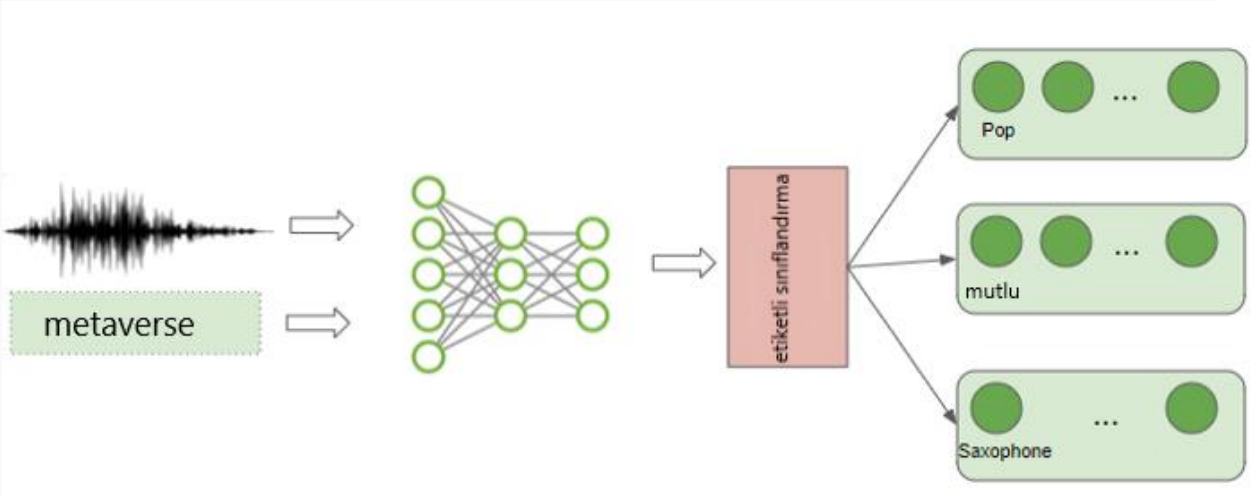
Örneğin, insan kalbinin farklı seslerini tespit etmek ve anormallikleri tespit etmek için teşhis amaçlı kullanılabilir (Şekil 2.9).



Şekil 2.9: Bireysel konuşmacıları bir sestan ayırma [40].

2.1.8 Müzik Türü Sınıflandırması ve Etiketleme

Müzik akışı hizmetlerinin popülaritesi ile çoğumuzun aşina olduğu bir diğer yaygın uygulama, müziği sese göre tanımlamak ve sınıflandırmaktır. Müziğin içeriği, ait olduğu türü bulmak için analiz edilir. Bu, çok etiketli bir sınıflandırma problemidir, çünkü belirli bir müzik parçası birden fazla türün altına düşebilir. Örneğin. rock, pop, caz, salsa, enstrümantal ve 'eskiler', 'kadın vokalist', 'mutlu', 'parti müziği' vb. gibi diğer yönler. Sesin kendisine ek olarak, müziğe zengin bir etiket kümesi eklemek için kullanılacak şarkıcı, yayın tarihi, besteci, şarkı sözleri vb. gibi müzikle ilgili meta veriler vardır. Bu, müzik koleksiyonlarını ses özelliklerine göre indekslemek, kullanıcının tercihlerine göre müzik önerileri sağlamak veya dinlediğiniz şarkıya benzer bir şarkıyı aramak ve almak için kullanılabilir (Şekil 2.10).



Şekil 2.10: Müzik Türü Sınıflandırması ve Etiketleme [39].

2.2. Özellik çıkarma

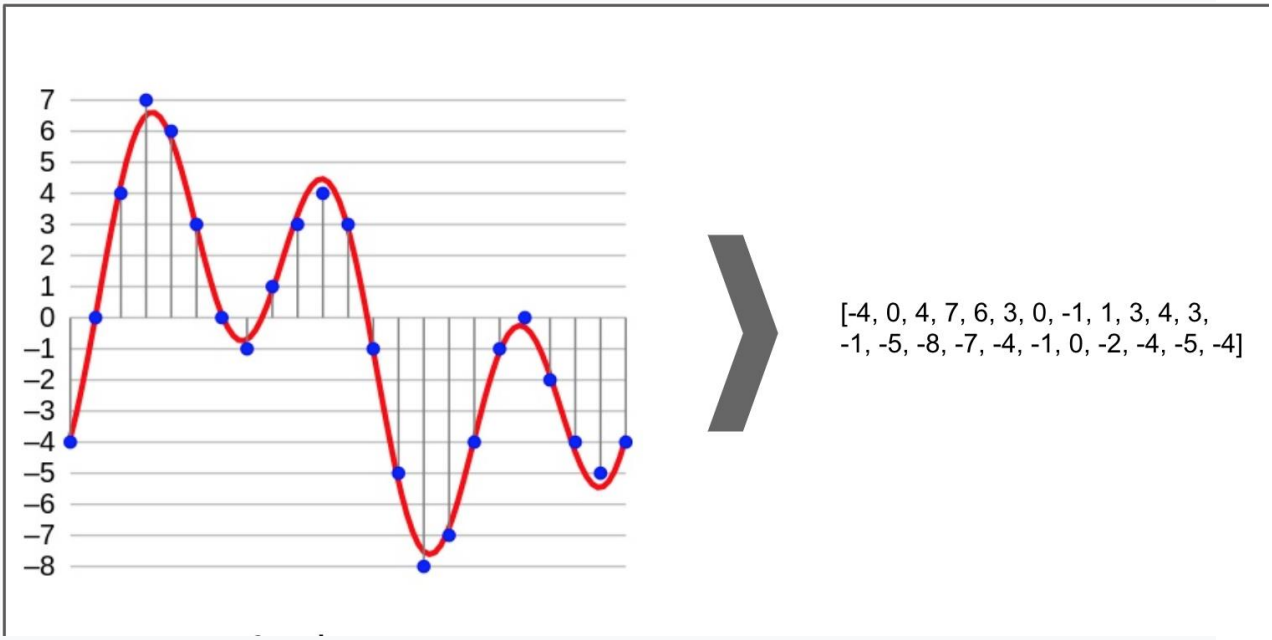
Özellik çıkarma, ilk ham veri setinin işlenmek üzere daha yönetilebilir gruplara indirgendiği bir boyutluluk azaltma işlemidir. Bu büyük veri kümelerinin bir özelliği, işlenmesi için çok fazla bilgi işlem kaynağı gerektiren çok sayıda değişkendir. Öznitelik çıkarma, değişkenleri seçen ve/veya özelliklerde birleştiren, işlenmesi gereken veri miktarını etkili bir şekilde azaltan ve yine de orijinal veri kümesini doğru ve eksiksiz olarak tanımlayan yöntemlerin adıdır. Özellik çıkarma işlemi, önemli veya ilgili bilgileri kaybetmeden işleme için gereken kaynak sayısını azaltmanız gerektiğinde yararlıdır. Özellik çıkarma, belirli bir analiz için gereksiz veri miktarını da azaltabilir. Ayrıca, verilerin azaltılması ve makinenin değişken kombinasyonlar (özellikler) oluşturma çabaları, makine öğrenme sürecinde öğrenme ve genelleme adımlarının hızını kolaylaştırır. Derin öğrenme konusundaki literatürün ve popülerliğin çoğu bilgisayarla görü ve doğal dil işleme (NLP) ile ilgili olsa da, otomatik konuşma tanıma (ASR), dijital sinyal işleme ve müzik sınıflandırma, etiketleme ve oluşturmayı içeren bir alan olan ses analizi, derin öğrenme uygulamalarının büyüyen alt alanı. Ses verisi analizi, kurumsal, sağlık, üretkenlik ve akıllı şehirlerdeki sayısız uygulama ile dijital cihazlar tarafından yakalanan ses sinyallerini analiz etmek ve anlamakla ilgilidir. Uygulamalar, müşteri destek çağrılarından müşteri memnuniyeti analizi, medya içerik analizi ve alımı, tıbbi teşhis yardımcıları ve hasta izleme, işitme bozukluğu olan kişiler için yardımcı teknolojiler ve kamu güvenliği için ses analizini içerir [44].

Özellik Çıkarmanın Pratik Kullanımları

3. Otomatik kodlayıcılar – Otomatik kodlayıcıların amacı, verimli veri kodlamanın denetimsiz olarak öğrenilmesidir. Öznitelik çıkarma, yenilerini türetmek için orijinal veri kümesinin kodlamasından öğrenerek kodlama için verilerdeki temel özellikleri belirlemek için burada kullanılır [43].
4. Kelime Torbası – Bir cümlede, belgede, web sitesinde vb. kullanılan kelimeleri (özellikleri) çıkaran ve bunları kullanım sıklığına göre sınıflandıran doğal dil işleme tekniği. Bu teknik görüntü işlemeye de uygulanabilir[43].
5. Görüntü İşleme – Algoritmalar, dijital bir görüntü veya videodaki şekil, kenar veya hareket gibi özellikleri algılamak için kullanılır[43].

2.2.1 Ses dosyasına genel bakış

Ses alıntıları, .wav formatındaki dijital ses dosyalarıdır. Ses dalgaları, örnekleme hızı olarak bilinen ayrık aralıklarla örneklenerek sayısallaştırılır (CD kalitesinde ses için tipik olarak 44.1kHz, yani örnekler saniyede 44.100 kez alınır). Her örnek, belirli bir zaman aralığındaki dalganın genliğidir; burada bit derinliği, örneğin ne kadar ayrıntılı olduğunu belirler, aynı zamanda sinyalin dinamik aralığı olarak da bilinir (tipik olarak, bir örneğin 65.536 genlik değerinden değişebileceği anlamına gelen 16 bit). Sinyal işlemede örnekleme, sürekli bir sinyalin bir dizi ayrık değere indirgenmesidir. Sıklık veya örnekleme oranı, belirli bir zaman aralığında alınan örneklerin sayısıdır. Yüksek bir örnekleme frekansı, daha az bilgi kaybı ancak daha yüksek hesaplama masrafı ile sonuçlanır ve düşük örnekleme frekansları, daha yüksek bilgi kaybına sahiptir, ancak hesaplanması hızlı ve ucuzdur (Şekil 2.11).



Şekil 2.11: Kırmızı renkte bir ses dalgaları, dijital olarak mavi renkte (örnekleme ve 4 bit nicelemeden sonra) temsil edilir ve sonuçtaki dizi sağda gösterilir [43].

2.2.2 Ses İşleme Uygulamaları:

1. Müzik koleksiyonlarını ses özelliklerine göre indeksleme.
2. Radyo kanalları için müzik önerme
3. Ses dosyaları için benzerlik araması
4. Konuşma işleme ve sentez — konuşma aracıları için yapay ses oluşturma

2.2.3. Python Kullanarak Ses Verileri İşleme

Ses, frekans, bant genişliği, desibel, vb. gibi parametrelere sahip olan bir ses sinyali biçiminde temsil edilir. Tipik bir ses sinyali, Genlik ve Zamanın bir fonksiyonu olarak ifade edilebilir.
Bu sesleri yakalamanıza ve bilgisayar tarafından okunabilir bir biçimde temsil etmenize yardımcı olacak yerleşik cihazlar vardır. Bu formatlara örnek olarak [43].

• wav (Dalga Biçimi Ses Dosyası) biçimi
• mp3 (MPEG-1 Ses Katmanı 3) formatı
• WMA (Windows Media Audio) formatı

Tipik bir ses işleme süreci, eldeki görevle ilgili akustik özelliklerin çıkarılmasını ve ardından algılama, sınıflandırma ve bilgi birleştirmeyi içeren karar verme şemalarını içerir. Neyse ki bu görevi kolaylaştıran bazı kullanışlı python kitaplıklarımız var. Python, Librosa ve PyAudio gibi ses işleme için bazı harika kitaplıklara sahiptir. Bazı temel ses işlevleri için yerleşik modüller de vardır [43].
Ses toplama ve oynatma için esas olarak iki kitaplık kullanacağız:

2.2.3.1. Librosa

Genel olarak ses sinyallerini analiz eden ancak daha çok müziğe yönelik bir Python modülüdür. Bir MIR (Müzik bilgisi alma) sistemi oluşturmak için somunları ve cıvataları içerir. Birçok örnek ve öğretici ile birlikte çok iyi belgelenmiştir[43].

Kurulum: Aşağıda belirtilen kodlar Ek 1.1'de mevcuttur.

Ses kod çözme gücünü artırmak için, birçok ses kod çözücüyle birlikte gelen ffmpeg yüklenmelidir.

2.2.3.2. **IPython.display**.

SesIPython.display.Audio, sesi doğrudan bir jupyter not defterinde çalmanıza olanak tanır. Bir ses dosyası yükleme:
Kurulum: Aşağıda belirtilen kodlar Ek 1.2'de mevcuttur.

Bu, 22KHZ mono varsayılan örnekleme hızına (sr) sahip bir numpy dizisi olarak bir ses zaman serisini döndürür. Bu davranışı 44.1KHz'de yeniden örnekleyerek değiştirebilir [43].
veya yeniden örneklemeyi devre dışı bırakmak için aşağıdaki kod kullanılabilir.

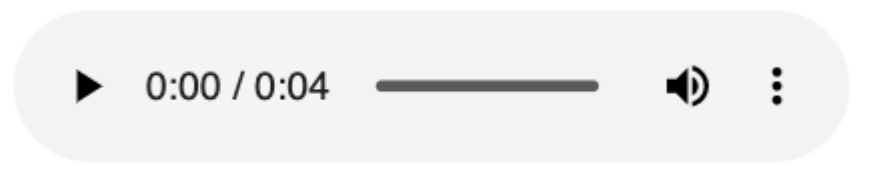
Kurulum: Aşağıda belirtilen kodlar Ek 1.3'de mevcuttur.

Örnekleme hızı, Hz veya kHz cinsinden ölçülen, saniyede taşınan ses örneklerinin sayısıdır.

Ses Çalma:

IPython.display.Audio'yu kullanarak sesi jupyter notebook'unuzda çalabilirsiniz.

Kurulum: Aşağıda belirtilen kodlar Ek 1.4'de mevcuttur.

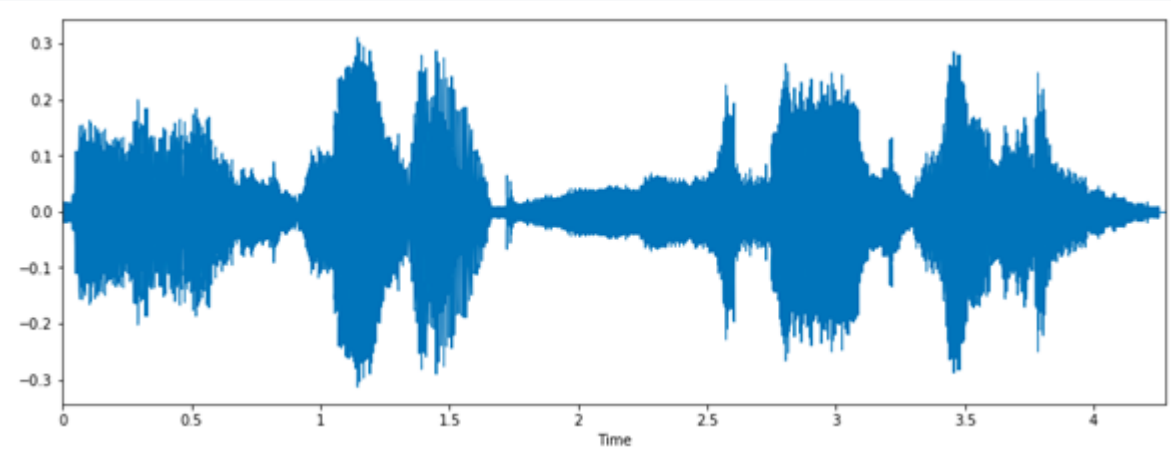


Şekil 2.12: Bu, bir ses widget'ı döndürür.

Sesi Görselleştirme:

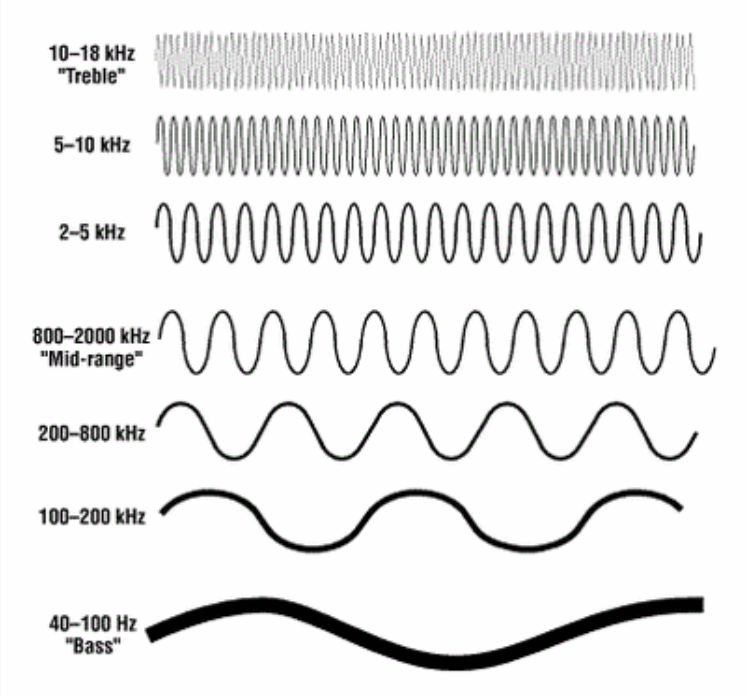
Ses dizisini librosa. display. waveplot kullanarak çizebiliriz:

Kurulum: Aşağıda belirtilen kodlar Ek 1.5'de mevcuttur.



Şekil 2.13: Yukarıdaki figürde bir dalga biçiminin genlik zarfının grafiği gösterilmektedir [43].

Tüm dalgaların olduğu gibi ses dalgalarının da bir frekansı vardır (Şekil 2.14).



Şekil 2.14: Ses Dalgalarının Frekansa Göre Sınıflandırılması [43].

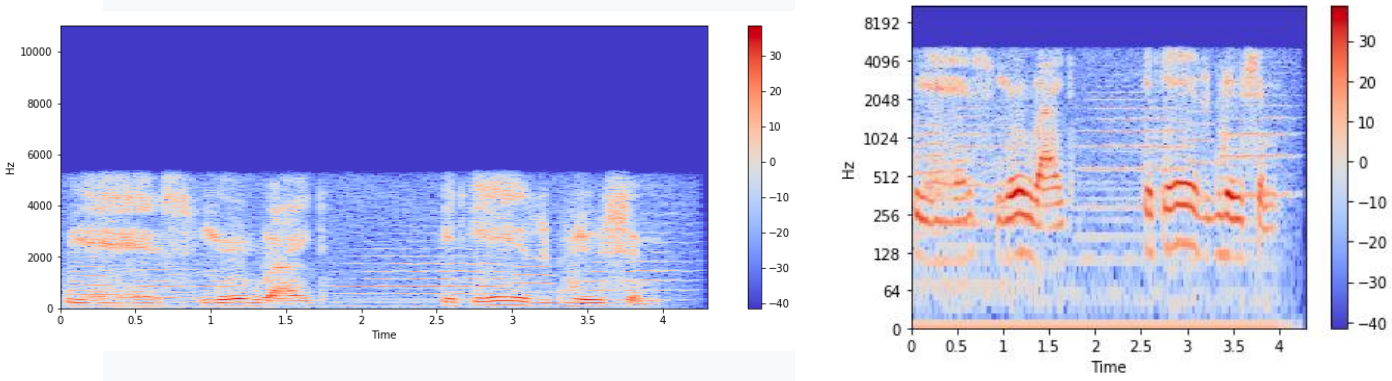
2.2.4. Spektrogramın elde edilmesi

Spektrogram, belirli bir dalga biçiminde bulunan çeşitli frekanslarda zaman içinde bir sinyalin sinyal gücünü veya "yüksekliğini" temsil etmenin görsel bir yoludur. Örneğin 2 Hz'de 10 Hz'de daha fazla veya daha az enerji olup olmadığını görmekle kalmaz, aynı zamanda enerji seviyelerinin zaman içinde nasıl değiştiğini de görebilir. Bir spektrogram genellikle bir ısı haritası olarak, yani rengin veya parlaklığın değiştirilmesiyle gösterilen yoğunluğa sahip bir görüntü olarak tasvir edilir [43].

Kullanarak bir spektrogram gösterebiliriz. librosa. display. specshow.

Kurulum: Aşağıda belirtilen kodlar Ek 1.6'de mevcuttur.

.stft(), verileri kısa vadeli Fourier dönüşümüne dönüştürür. STFT, belirli bir zamanda verilen frekansın genliğini bilebileceğimiz şekilde sinyalleri dönüştürür. STFT kullanarak, bir ses sinyalinin belirli bir zamanında çalan çeşitli frekansların genliği belirlenir. .specshow, bir spektrogramı görüntülemek için kullanılır [43].



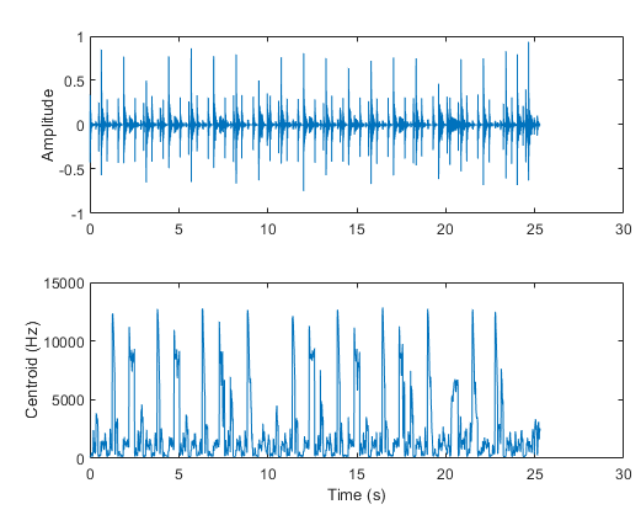
Şekil 2.15: Dikey eksen frekansları (0'dan 10kHz'e kadar) gösterir ve yatay eksen klibin zamanını gösterir. Tüm eylemin spektrumun altında gerçekleştiği görüldüğünde, frekans ekseni logaritmik bir eksene çevrilir [43].

Kurulum: Aşağıda belirtilen kodlar Ek 1.7'de mevcuttur.

2.2.5. Ses sinyalinden özellik çıkarma

Her ses sinyali birçok özellikten oluşur. Ancak, çözmeye çalıştığımız problemle ilgili özellikleri çıkarmalıyız. Analiz için kullanmak üzere öznitelik çıkarma işlemine öznitelik çıkarma denir. Birkaç özellik aşağıda ayrıntılı olarak incelenmiştir.

Fourier Dönüşümü kullanılarak zamana dayalı sinyalin frekans alanına dönüştürülmesiyle elde edilen temel frekans, frekans bileşenleri, spektral merkez, spektral akı, spektral yoğunluk, spektral yuvarlanma gibi spektral öznitelikler (frekans temelli öznitelikler) [43].



Şekil 2.16: Ses sinyalinden özellik çıkarma [43].

Şekil 2.16' deki Spektral centroid ayrıca konuşmayı sesli veya sessiz olarak sınıflandırmak için yaygın olarak kullanılır. Örneğin, ağırlık merkezi, sesli olmayan konuşma bölgelerinde atlar.

2.2.5.1. Spektral Merkez

Spektral centroid, bir spektrumun enerjisinin hangi frekansta merkezlendiğini veya başka bir deyişle bir sesin "kütle merkezinin" nerede olduğunu gösterir. Bu ağırlıklı ortalama gibidir: [43].

$$fc = \frac{\sum_k S(k) f(k)}{\sum_k S(k)} \qquad (1)$$

S(k), k frekansındaki spektral büyüklüktür,

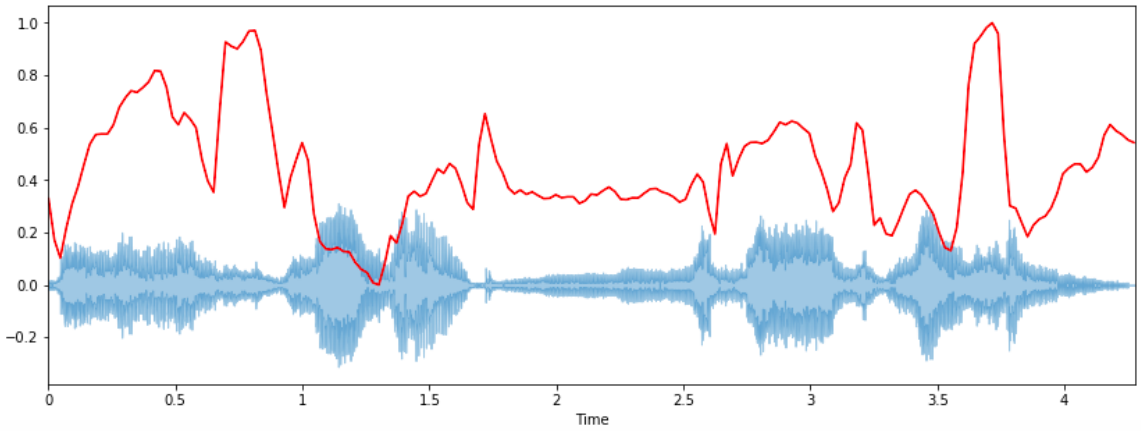
f(k) k kutusundaki frekanstır

(k) band edges, (bins)' de spektral ağırlık merkezini hesaplamak için [43].

librosa. feature. spectral_centroid, bir sinyaldeki her çerçeve için spektral merkezi hesaplar:

Kurulum: Aşağıda belirtilen kodlar Ek 1.8'de mevcuttur.

Spectral_centroid, örneğinizde bulunan çerçeve sayısına eşit sütunlara sahip bir dizi döndürür (Şekil 2.17).

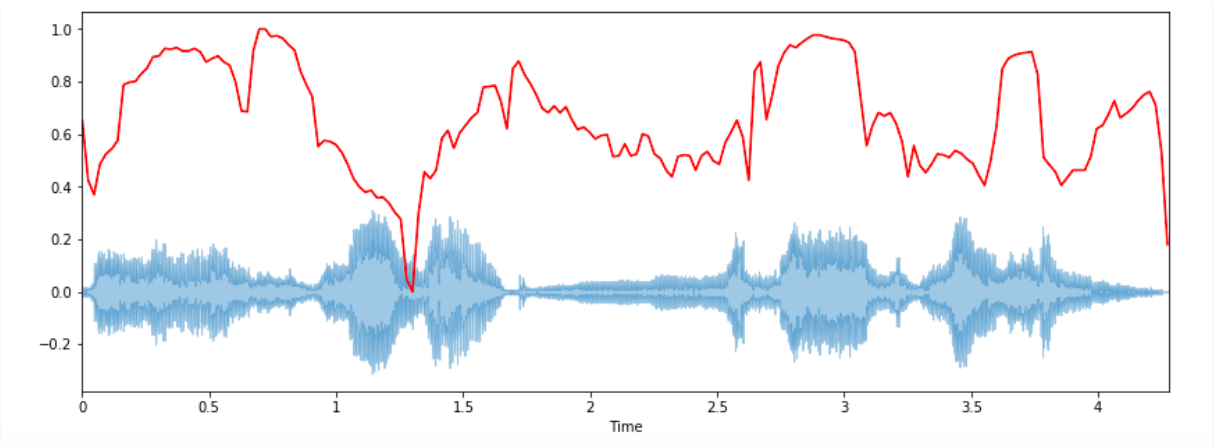


Şekil 2.17: Başlangıçta spektral merkezde bir artış var [43].

2.2.5.2. Spektral Devrilme (Spectral Rolloff)

Sinyalin şeklinin bir ölçüsüdür. Yüksek frekansların 0'a düştüğü frekansı temsil eder. Bunu elde etmek için, gücünün %85'inin daha düşük frekanslarda olduğu güç spektrumundaki kutuların oranını hesaplamamız gerekir (Şekil 2.18).

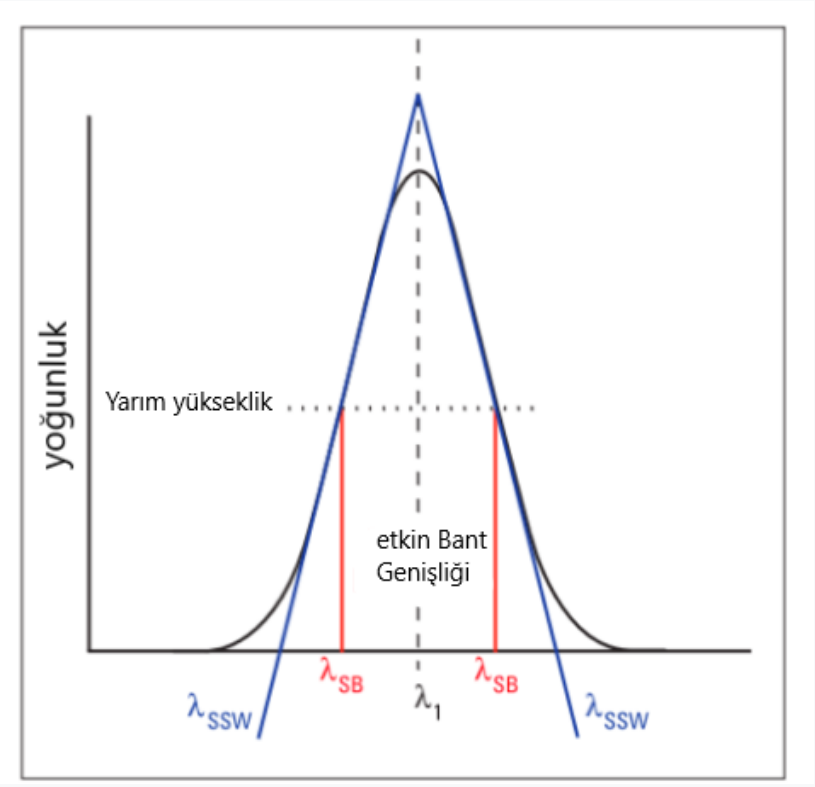
Kurulum: Aşağıda belirtilen kodlar Ek 1.9'de mevcuttur.



Şekil 2.18: librosa. feature. spectral_rolloff, bir sinyaldeki her çerçeve için rolloff frekansını hesaplar [43].

2.2.5.3. Spektral Bant Genişliği (Spectral Bandwidth)

Spektral bant genişliği, tepe maksimumun yarısında (veya yarı maksimumda [FWHM] tam genişlikte) ışık bandının genişliği olarak tanımlanır ve dalga boyu ekseninde iki dikey kırmızı çizgi ve λSB ile temsil edilir[43].

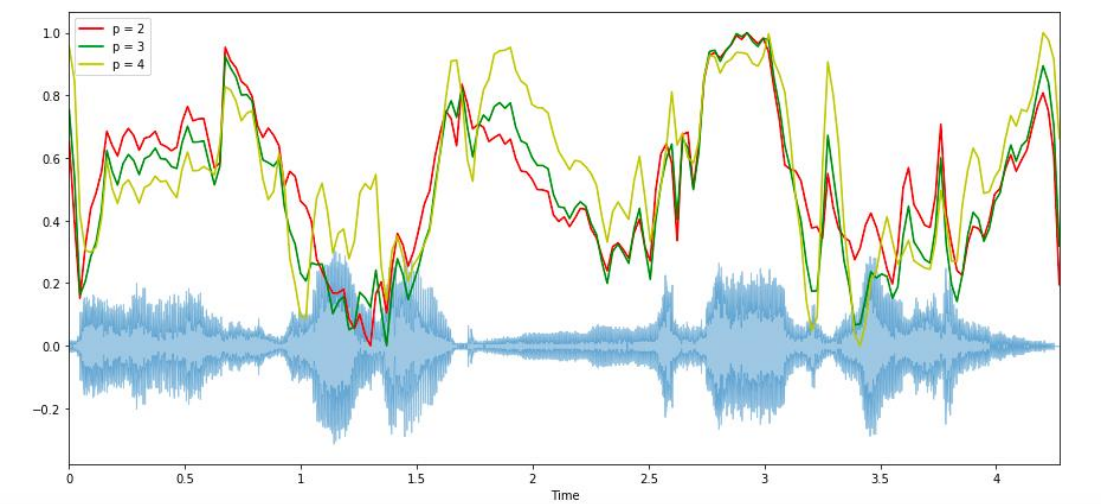


Şekil 2.19: Spektral Bant Genişliği (Spectral Bandwidth) [43].

Monokromatörden çıkan dalga boylarının Gauss yoğunluk dağılımıdır. Spektral bant genişliği kırmızı sınırlar ve λSB ile tanımlanır. Spektral yarık genişliği, mavi sınırlar ve λSSW ile gösterilir (Şekil 2.19).

Kurulum: Aşağıda belirtilen kodlar Ek 1.10'de mevcuttur.

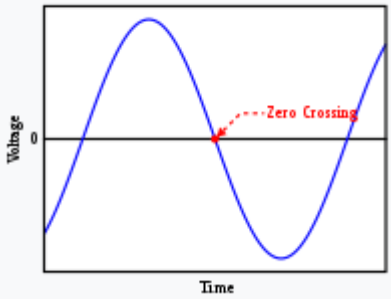
librosa.feature.spectral_bandwidth sipariş-p spektral bant genişliğini hesaplar (Şekil 2.20).



Şekil 2.20: librosa.feature.spectral_bandwidth sipariş-p spektral bant genişliğini hesaplar [43].

2.2.5.4. Sıfır Geçiş Oranı (Zero-Crossing Rate)

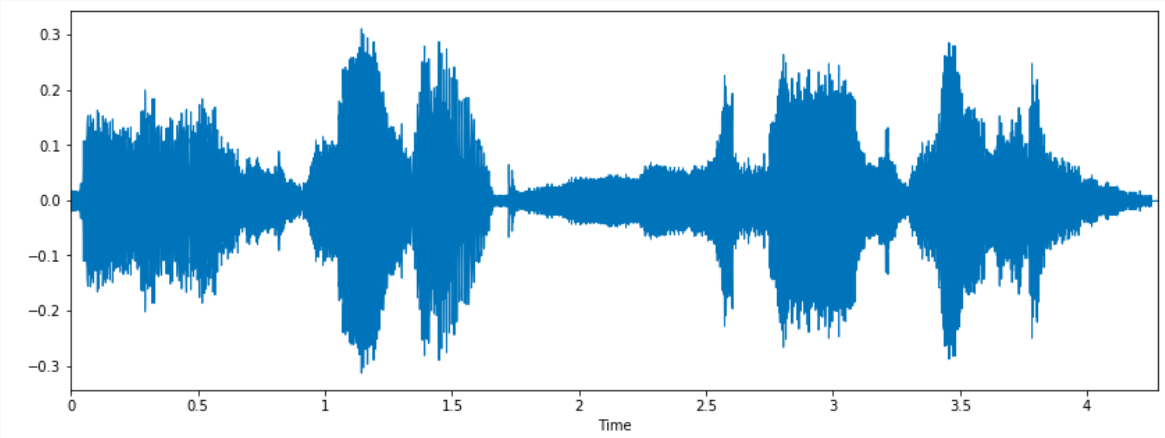
Bir sinyalin düzgünlüğünü ölçmenin en basit yolu, o sinyalin bir segmenti içindeki sıfır geçiş sayısını hesaplamaktır. Bir ses sinyali yavaşça salınım yapar, bir nokta matematiksel fonksiyon değişiklikleri - örneğin, 100 Hz'lik bir sinyal saniyede sıfırı 100'ü geçer - oysa sessiz bir frikatif saniyede 3000 sıfır geçişe sahip olabilir. Zaman içindeki gerilimi temsil eden bir dalga biçiminin çizgi grafiğindeki sıfır geçiş (Şekil 2.21).



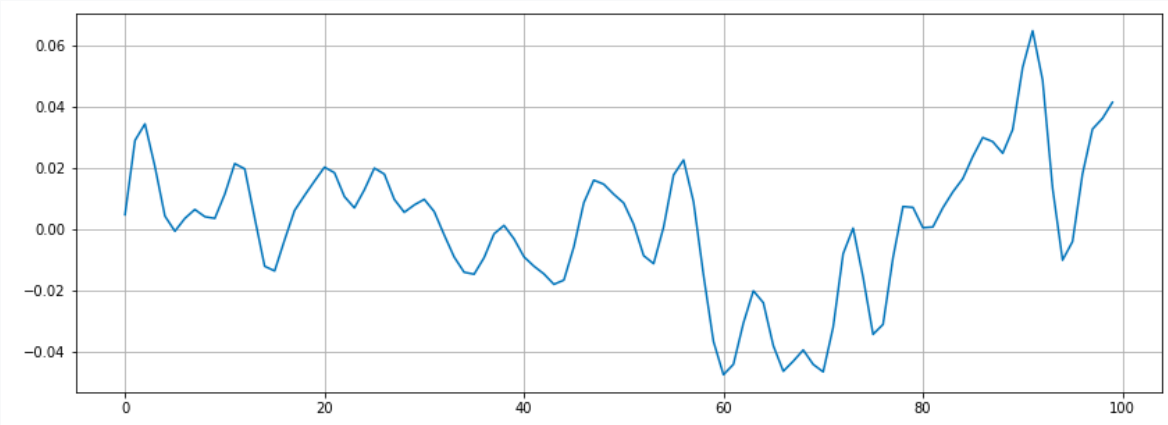
Şekil 2.21: Sıfır geçiş oranını [43].

Metal ve rock'takiler gibi yüksek perküsyonlu sesler için genellikle daha yüksek değerlere sahiptir. Aşağıdaki kod ile sıfır geçiş oranı hesaplaması ve görselleştirilmesi yapılmıştır.

Kurulum: Aşağıda belirtilen kodlar Ek 1.11'de mevcuttur.



Şekil 2.22: sıfır geçiş oranının hesaplaması



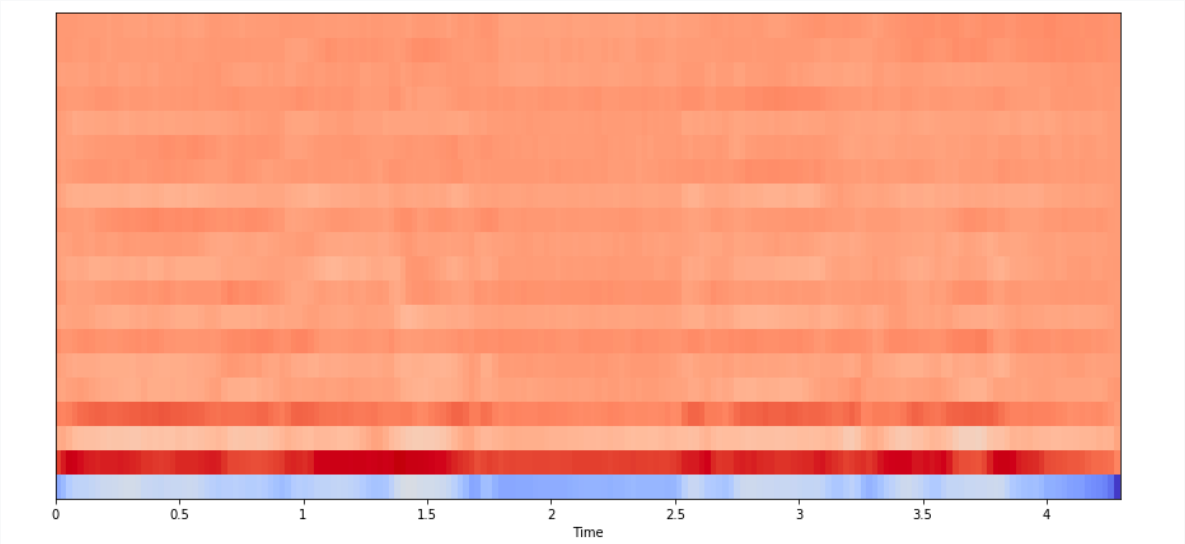
Şekil 2.23: sıfır geçiş oranının daha detaylı yakınlaştırması [43].

Tabloya göre 16 sıfır geçiş var. Bu geçişler Librosa ile doğrulanmaktadır.

2.2.5.5. Mel-Frekans Cepstral Katsayıları (MFCC'ler)

Bir sinyalin Mel frekans kepstral katsayıları (MFCC'ler), bir spektral zarfın genel şeklini kısaca açıklayan küçük bir özellikler dizisidir (genellikle yaklaşık 10-20). İnsan sesinin özelliklerini modellemektedir (Şekil 2.24).

Kurulum: Aşağıda belirtilen kodlar Ek 1.12'de mevcuttur.

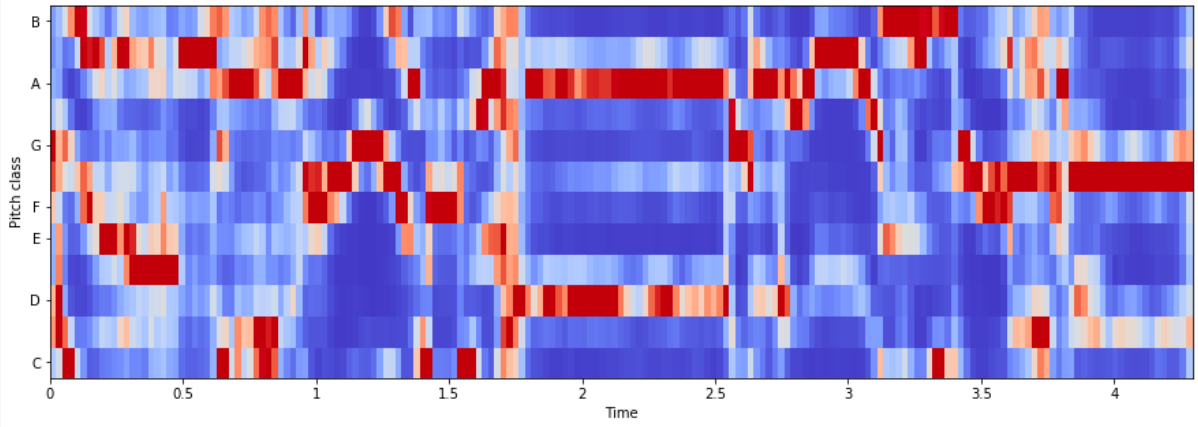


Şekil 2.24: Mel-Frekans Cepstral Katsayıları (MFCC'ler) [43].

2.2.5.6. Renk özelliği (Chroma feature)

Bir Rank özelliği, tipik olarak, sinyalde her perde sınıfının ({C, C#, D, D#, E, … B}) ne kadar enerji bulunduğunu gösteren 12 elemanlı bir özellik vektörüdür. Müzik parçaları arasındaki benzerlik ölçüsünü tanımlamanın sağlam bir yolunu sağlar (Şekil 2.25).

librosa. feature. chroma_stft, Chroma özelliklerinin hesaplanması için kullanılır [43].

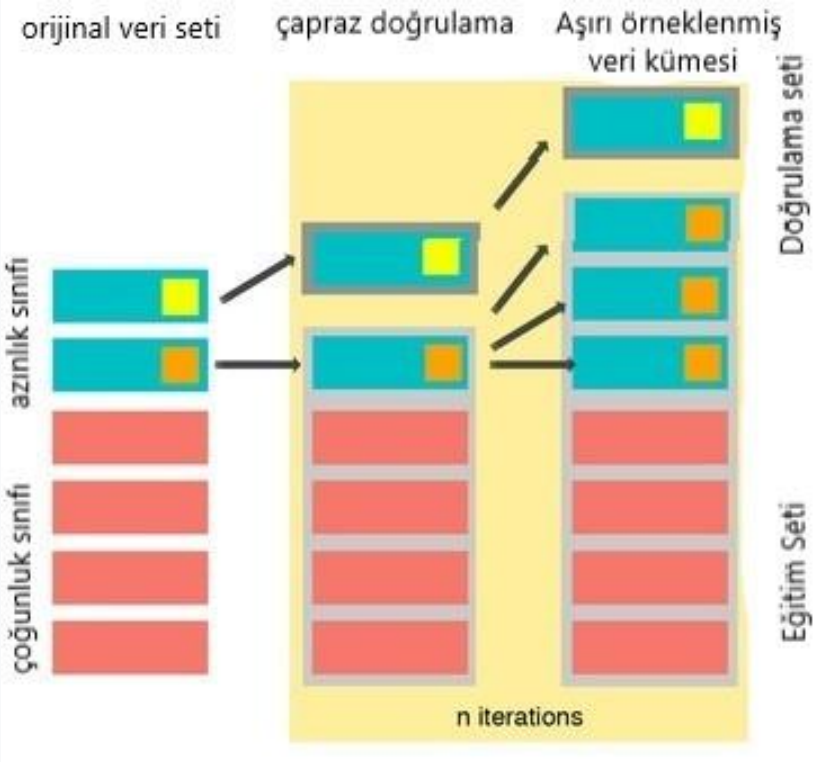


Şekil 2.25: Renk özelliği [43].

Kodlar sayesinde ses verileriyle nasıl oynalınacağı ve python kullanarak önemli özellikleri nasıl çıkarılacağı belirlenmiştir. Sonraki kısımda ise bu özellikler kullanılarak müzik türü sınıflandırması için bir YSA modeli oluşturulacaktır [43].

2.3. Çapraz Doğrulama

Çapraz Doğrulama, modelin görünmeyen veriler üzerindeki etkinliğinden ve doğruluğundan emin olunmasına yardımcı olan bir örnekleme tekniğidir. Mevcut girdi veri kümesinin alt kümeleri üzerinde diğer birkaç Makine öğrenimi modelini eğiterek ve bunları veri kümesinin alt kümesinde değerlendirerek Makine Öğrenimi modellerini değerlendirmek için kullanılan bir yöntemdir. Çapraz Doğrulamanın temel işlevselliğini gösteren belirli adımlar mevcuttur: İlk adım, temizlenmiş veri setini eşit büyüklükte K bölümlerine bölmektir (Şekil 2.26).



Şekil 2.26: Çapraz doğrulama [45].

• Sonrasında Fold-1'i bir test gurubu, diğer K-1'i eğitim olarak ele alınır.

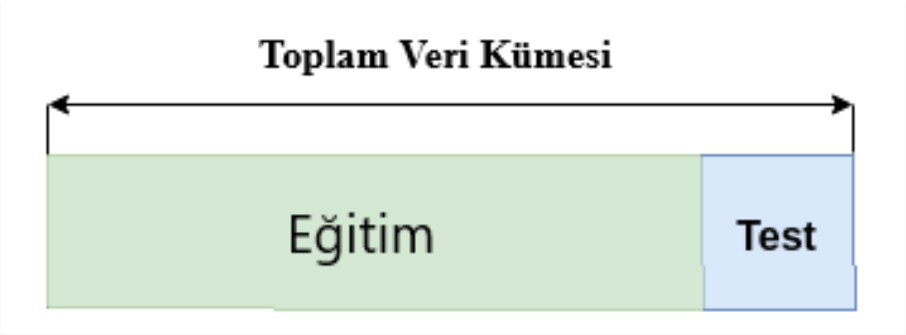
• Bu işlem K sayısı kadar tekrarlanır.

• Son adım, tüm iterasyonların puanlarının ortalaması alınması ile sonuçlanır.

Çapraz Doğrulama Türleri

2.3.1. Holdout Yöntemi

Bu teknik, tahminleri almak için eğitim veri setinin bir kısmını eğitime kalan diğer kısmı ise test etmek için kulanılan bir yöntemdir (Şekil 2.27).



Şekil 2.27: Holdout Yöntemi [45].

Artıları:

• Bu Yöntem, verilerden tamamen bağımsızdır.

• Bu Yöntemin yalnızca bir kez çalıştırılması gerekir, bu nedenle hesaplama maliyetleri daha düşüktür.

Eksileri:

• Performans, verilerin daha küçük boyutu göz önüne alındığında daha yüksek varyansa tabidir.

2.3.2. K-Katlı Çapraz Doğrulama

Veriye Dayalı bir Dünyada, modeli eğitmek için hiçbir zaman yeterli veri yoktur, bunun üzerine bunun bir kısmını doğrulama için kaldırmak daha büyük bir yetersiz uydurma sorunu yaratır ve veri kümemizdeki önemli kalıpları ve eğilimleri kaybetme riskiyle karşı karşıya kalınır. Dönüş Bias'ı artırır. İdeal olarak, modeli eğitmek için bol miktarda veri sağlayan ve doğrulama kümeleri için bol miktarda veri bırakan bir yönteme ihtiyaç duyulur.

K-Fold çapraz doğrulamada, veriler k alt kümeye bölünür veya k kez tekrarlanan bir holdout yöntemi olarak alabiliriz, öyle ki her seferinde k alt kümeden biri doğrulama kümesi olarak kullanılır ve diğer k-1 alt kümeler eğitim kümesidir. Modelimizin toplam verimliliğini elde etmek için tüm k denemede hatanın ortalaması alınır. Her veri noktasının tam olarak bir kez doğrulama setinde olacağı ve k-1 zaman eğitim setinde olacağı görülür. Bu, verilerin çoğunu uydurma için kullanır ve yanlılığı azaltmamıza yardımcı olur. Aynı zamanda verilerin çoğu doğrulama kümesinde de kullanıldığından varyansı azaltır [45].

Artıları

• Bu, hesaplama gücü sorununun üstesinden gelinmesine yardımcı olacaktır.

• Veride bir aykırı değer varsa, modeller çok fazla etkilenmeyebilir.

• Değişkenlik sorununun üstesinden gelmemize yardımcı olur.

Eksileri

• Dengesiz veri kümeleri modelimizi etkileyecektir.

2.3.3. Tabakalı K-Katlama Çapraz Doğrulama

K-Fold Çapraz Doğrulama tekniği, Dengesiz (Imbalanced) bir Veri seti için beklendiği gibi çalışmayacaktır. Dengesiz bir veri setimiz olduğunda, K Katlama çapraz doğrulama tekniğinde küçük bir değişiklik gerekliydi, öyle ki her kat, tam olarak her bir çıktı sınıfının yaklaşık olarak aynı katmanlarını içerir. K-Katlama Çapraz Doğrulamada bir katman kullanmanın bu varyasyonu, Katmanlı K-Katlama Çapraz Doğrulama olarak bilinir [45].

Artıları

• Hiper parametre ayarı kullanarak farklı modelleri iyileştirebilir.

• Modelleri karşılaştırmamıza yardımcı olur.

• Hem Önyargıyı hem de Varyansı azaltmaya yardımcı olur.

• 4. Bırakma Çapraz Doğrulama

Bu yaklaşımda, toplam n veri noktasından eğitim verisinin dışında p veri noktası bırakılır, ardından modeli eğitmek için n-p örnekleri ve doğrulama seti olarak p noktaları kullanılır. Bu, tüm kombinasyonlar için tekrarlanır ve ardından hatanın ortalaması alınır.

• Sıfır rastgeleliğe sahiptir.

• Önyargı daha düşük olur.

Eksileri

• Bu yöntem kapsamlıdır ve hesaplama açısından olanaksızdır.

• Ayrıca Okuyun: Makine Öğreniminde Kariyer;

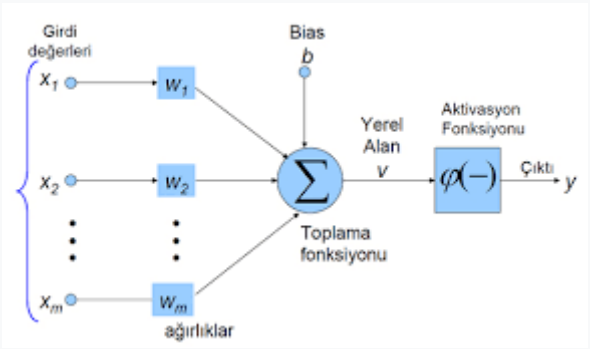
Bu yazıda, Veri Bilimi Projesi Yaşam Döngüsünde bir Makine Öğrenimi Modelinin Doğrulanmasının önemi, doğrulama ve çapraz doğrulamanın ne olduğu ve farklı Çapraz Doğrulama teknikleri belirlendi. Bu tekniklerin avantaj ve dezavantajları gözlemlendi [44].

2.4. Derin Öğrenme

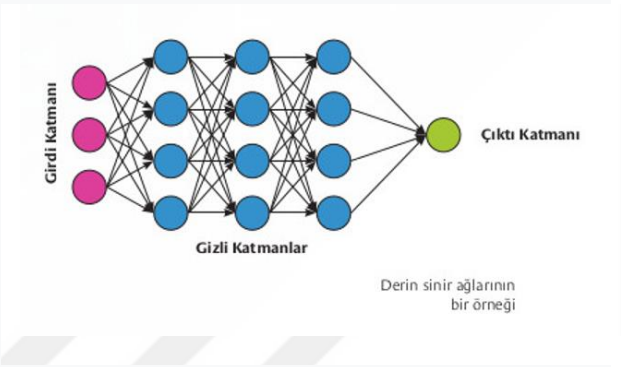
Derin öğrenme, bir makinenin insan vücudundaki nöronları simüle ederek kendi kendine öğrenmesini sağlayan teoriler ve algoritmalar bulmakla ilgilenen yeni bir araştırma alanıdır. Yapay zeka ile ilgilenen bilim dallarından biridir. Makine öğrenimi bilimlerinin bir dalı olan derin öğrenme araştırmalarının çoğu, doğrusal ve doğrusal olmayan değişkenler kullanarak büyük bir veri kümesini analiz ederek yüksek derecede soyutlamalar ortaya çıkarmaya yönelik yöntemler bulmaya odaklanır.

Bu derin öğrenmedeki keşifler, yüz tanıma, konuşma tanıma, bilgisayarla görme ve doğal dil işleme dahil olmak üzere birçok alanda hızlı ve etkili ilerleme olduğunu kanıtladı.

Makine, diğerlerinin yanı sıra, metin ve sürekli verilerle yaygın olarak kullanılan Tekrarlayan Ağlar (RNN'ler) ve görsel lobdaki biyolojik süreçlerden ilham alan Evrişimli Sinir Ağları (CNN'ler) dahil olmak üzere derin öğrenme ağlarının çeşitli tasarımlarını kullanarak büyük verilerden öğrenir.

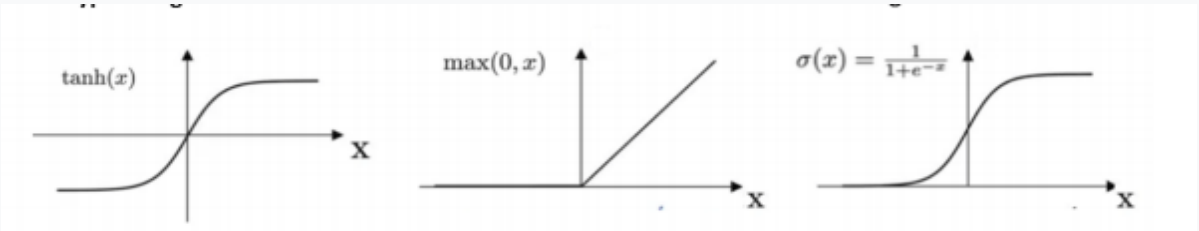


Şekil 2.28: Yapay Nöron [46].



Şekil 2.29: Yapay Sinir Ağı[46].

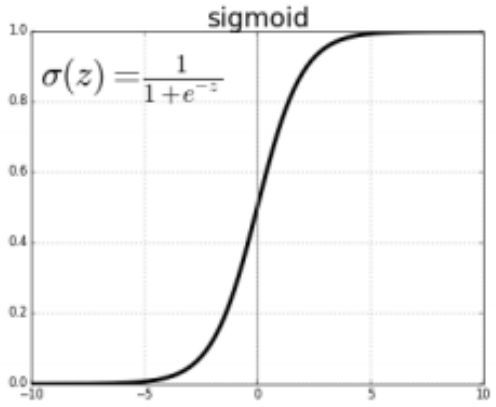
Yapay veya algısal nöronların birden fazla girişi (x) vardır ve ölçeklenir. Ağırlık (w), anlamlılığı ifade etmek için girdi değerini artırmak veya azaltmak için kullanılır. Sensör çıkış değeri, giriş ve aktivasyon değerine bağlıdır. Aktivasyon fonksiyonu, aktivasyonu dönüştürmek için kullanılan doğrusal olmayan bir fonksiyondur. Birim seviyesi bir çıkış sinyaline dönüştürülür. Aktivasyon işlevlerinden bazıları şunlardır: Sigmoid, Tanh, ReLu ve Softmax. Günümüzde ReLu en çok kullanılan aktivasyon fonksiyonudur ve genellikle Softmax kullanılmaktadır. Son katman, çıktı vektörünü olasılık vektörü olarak alınır (Şekil 2.30).



Şekil 2.30: Aktivasyon Fonksiyonu [48].

Aktivasyon fonksiyonu'da, bir düğümün Aktivasyon Kısmı, girdiye veya uyaran kümesine bağlı olarak bir düğümün dönüş sinyalini tanımlar. Her nöron için giriş, enerji veren tek değerli fonksiyon ile hesaplanır ve daha sonra sonuç çıkış olarak gönderilir. Bu çıkışa bağlı olarak, enerji verme fonksiyonu bu nöronun bağlantısının tetiklenip tetiklenmeyeceğine karar verir. Rectified Linear Units (ReLU) basitlikleri nedeniyle en çok kullanılan aktivasyon alt programıdır. ReLU'nun diğer aktivasyon prosedürüne kıyasla eğitimi daha hızlı hale getirdiği kanıtlanmıştır. ReLU aktivasyon fonksiyonu, çıkış sinyali cypher'dan küçükse cypher'ı geri kazanır ve giriş cypher'dan büyükse girişin aynı değerini geri getirir [54]. Sigmoid sosyal olayın doğrusal olmayan doğası, sinir ağları için benzersizdir [49]. Sigmoid kısmı, girilen değeri bir

skalaya (0, birlik) dönüştürür, burada lise değerleri asimptotik bir yoldan 1'e ve çok düşük değerler asimptotik olarak 0'a eğilimlidir. Onun denklemi aşağıdaki şekildedir (şekil 2.31).



Şekil 2.31: Görüldüğü gibi tipik "S" şekli nedeniyle bu şekilde adlandırılır [49].

Sigmoid özellikleri, Yavaş yakınsama, Sıfır merkezli değildir, 0 ile 1 arasında sınırlandırılmıştır ve Son seviyede iyi harekete geçer.

Yapay sinir ağları, eğitim verilerine ve her uygulamaya uygun olmalıdır. Bu nedenle en iyi sonuçları elde etmek için parametreler değiştirilmelidir.

2.4.1. Kayıp Fonksiyonu (Loss)

Kayıp fonksiyonu, hedefin gerçek ve tahmin edilen değeri arasındaki mesafeyi ölçer. Tahminlerin ve beklentilerin eğitim aşamasına tam olarak uyması için kaybın sıfırlanması hedeflenmektedir. Ölçülen değer üzerinden modelin istenilen değere ne kadar yaklaştığı sonucuna ulaşılır. En yaygın kullanılan fonksiyon ise (Mean Squared Error) ortalama kare hatadır [47].

2.4.2. Optimizer

Optimize edici, kayıp fonksiyonun değişimini en aza indirilmesine ve ızgara ağırlığı ve sapma değerlerinin ayarlanmasına yardımcı olan bir optimizasyon algoritmasıdır. Stokastik Gradient Descent, Adam, Adamax ve RMSprop gibi birçok farklı tür vardır [47].

2.4.3. Devir (Epochs)

(epok, devir), tüm veri setinin sinir ağı üzerinden ileri geri geçtiği zamandır. Modeli eğitmek için ağa bir kez, çağ sayısının daha büyük olması gerekmektedir. Bunun nedeni, zaman arttıkça ağırlıkların sayısı arttıkça ağın değişmesi, eğrinin uygunsuzdan daha iyiye değişmesi ve hatta eğri uydurmaya ulaşmasıdır [47].

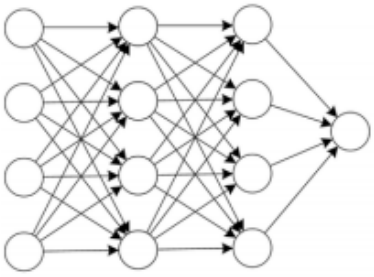
2.4.4. Küme Boyutu (Batch Size)

Küme boyutu, makine öğreniminde kullanılan bir terimdir ve bir yinelemede kullanılan eğitim örneklerinin sayısını ifade eder. Küme boyutu üç seçenekten biri olabilir:

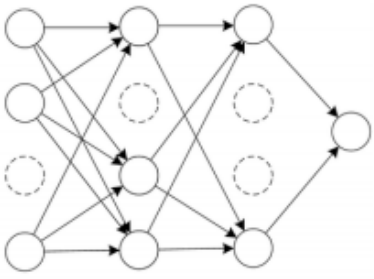
- Küme modu (batch mode): parti boyutunun toplam veri kümesine eşit olduğu ve böylece yineleme ve dönem değerlerini eşdeğer hale getirdiği yer.
- Mini-küme modu: küme boyutunun birden büyük ancak toplam veri kümesi boyutundan küçük olduğu yer. Genellikle, toplam veri kümesi boyutuna bölünebilen bir sayı.
- stokastik mod: parti boyutunun bire eşit olduğu mod. Bu nedenle gradyan ve sinir ağı parametreleri her örnekten sonra güncellenir [47].

2.4.5. Dropout

Dropout, eğitim verilerinde karmaşık eş-adaptasyonları önleyerek yapay sinir ağlarında fazla uydurmayı azaltmak için bir düzenleme tekniğidir. Sinir ağları ile model ortalamasını gerçekleştirmenin verimli bir yoludur. Dropout, sinir ağındaki birimlerin (nöronların) atılmasını ifade etmektedir. Bu nöronlar, eğitim aşamasında belirli bir olasılıkla rastgele atılmaktadır. Sinir ağını diğer nöronların farklı rastgele alt kümeleri ile karşılaştırılabilecek daha güçlü işlevleri öğrenmeye zorlamaktadır [47]. Çok sayıda parametreye sahip bu teorik hesap, eğitim bilgilerine kolayca sığabilir. Dropout'tur, Şekil (32, 33)'te gösterildiği gibi bir nöral ağ örgüsünde bütünün bırakılması anlamına gelir.



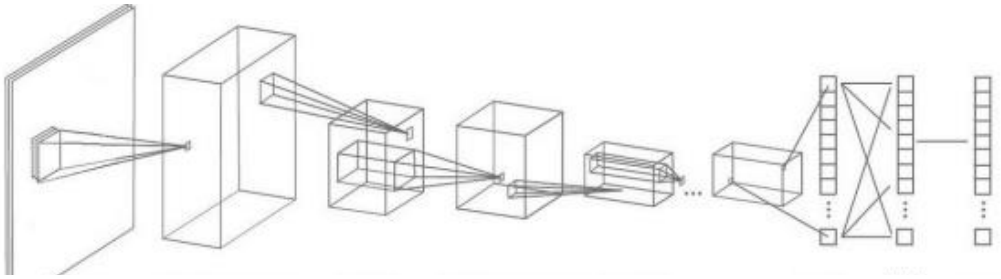
Şekil 2.32: Standart sinir ağı [49].



Şekil 2.33: Dropout yaptıktan sonra ağı [49].

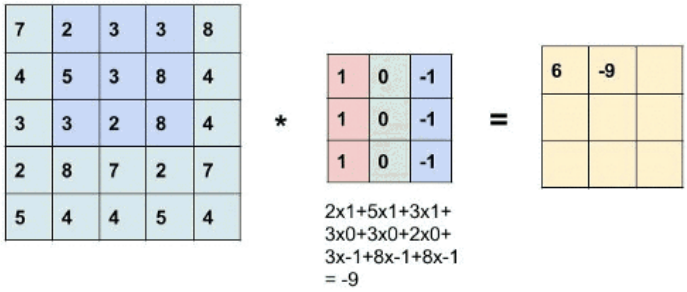
2.5. Evrişimli Sinir Ağıları

Evrişimli Sinir Ağı (CNN), görüntü tanıma ve sınıflandırma alanlarında çok etkili olduğu kanıtlanmış bir sinir ağı türüdür. Ses bir spektrograma dönüştürüldüğü için CNN'nin etkisi normal sinir ağından daha iyidir. CNN, ilgili filtrelerin uygulanması ile aşağıdaki şekillerde görüntüdeki uzamsal korelasyonu başarılı bir şekilde yakalayabilmektedir. Bu mimari, ilgili parametre sayısının azalması ve yeniden kullanılabilirlik ağırlığının azalması nedeni ile görüntüye daha iyi uyar. CNN'nin rolü, görüntüyü daha kolay hale getirmektir (Şekil 2.34).



Şekil 2.34: Evrişimli Sinir Ağı Yapısı [48].

Tüm sinir ağları gibi, bir sinyal tabanına, gizli bir düzleme ve bir çıkış katmanına sahiptirler, ancak bu tür ağların iki tür gizli katmanı vardır: evrişim katmanları ve havuzlama katmanları [49]. Bu gizli seviye, isim penceresinin küçük iki boyutlu Eş Anlamlıları / Hipernimlerinde (Tahmini Sıklığa Göre Sıralanmıştır) yerel kalıpları öğrenmekle ilgilidir. Çizgiler, kenarlar ve driblet rengi gibi görüntülerdeki görsel özellikleri tespit etmeye yarayan katmanlardır. Bu katmanlar, belirli bir noktadaki görüntünün bir özelliğini öğrenmeyi sağlar ve onu kendi görüntüsünde herhangi bir yerde tanıyabilir [50]. Evrişim katmanları, önceki katmanda öğrenilenlere göre farklı öğeleri daha karmaşık öğrenebilmektedir. Bir örnek, bir katmanın paradigmada görünen farklı kişisel kredi limitlerini öğrenmesi ve sonraki katmanın, önceki katmanda öğrenilen farklı zorlamalardan oluşan bir fonksiyon aralığının öğelerini öğrenebilmesi olabilir. Bunu yapmak için zihinsel yetenek, bu ağların giderek daha karmaşık hale gelen görsel kavramları verimli bir şekilde öğrenmesini sağlar [52]. Evrişimli sinir ağları, katman eksenleri, genişlik, boy ve kanal üzerinde çalışır. Yorumda, açıklama görüntüsü gri tonlamalıysa kanal tek olabilir veya RGB renkli görüntü için her renk için bir tane (kırmızı, yeşil ve mavi) olmak üzere 3 olabilir. Ağın içinde derinlik genellikle 3'ten yüksektir. Tüm giriş nöronları bu evrişim katmanının tüm nöronlarına bağlı olmayacaktır. Görüntünün pikselini depolayan giriş nöronlarının küçük lokalize alanı aracılığıyla yapılır. Sonuç, Orijinal model görüntüsüne Çekirdek filtresinin uygulanmasının sonucudur. Sonuç, her slaytta orijinal görüntüye uygulanan çekirdek madden ürünü ve çıkıntısıdır. Çekirdek boyutuna bağlı olarak çıktı boyutu değişir. Her Kartezyen ürün iyon değeri, ürünün sonucudur ve ilk Orijinal model görüntüsüne filtreli orijinal paradigmanın özetidir [53]. Son ürün seçeneği 4x4 boyutuna sahiptir çünkü 6x6 giriş görüntüsünde 3x3 filtreyi 4x4 filtre takılabilir. Matris, sol üstteki tüm sinir hücresinden geçmeye başlar ve sağa doğru hareket eder. Bir parkur bitirildiğinde, aşağıda gösterilen alıştırmadaki gibi devam edilir (Şekil 2.35).

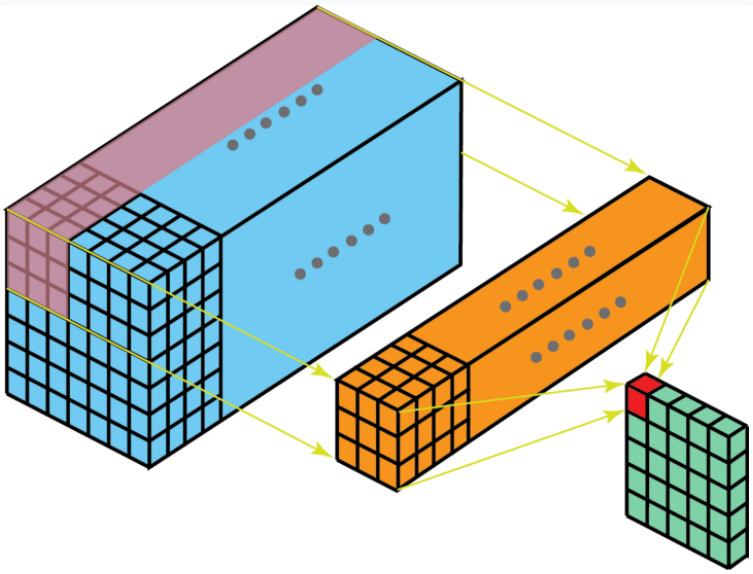


Şekil 2.35: farklı filtreler uygulayarak farklı özellikleri tespit etmek mümkündür [52].

CNN'lerin katman tipleri aşağıdaki bölümlerde açıklanmıştır.

2.5.1. Evrişimsel Katmanlar (Convolutional Layers)

Evrişim katmanı, 2B evrişim uygulayan ve dikdörtgen şeklinde açıklığa sahip bir katmandır. Kenarlar, gerilimler, gölge düşüşleri ve diğerleri gibi görüntülerdeki işlevleri veya görsel işlevleri algılayabilmektedir. Bu umut verici bir özelliktir. Çünkü, fotoğrafta belirli bir faktörde bir özellik öğrendildiğinde daha sonra onu herhangi bir bölümünde tanıyabilmektedir. Bir başka kritik işlev, evrişimli katmanların mekansal ilişkileri koruyarak örüntülerin mekansal hiyerarşilerini inceleyebilmesidir (Şekil 2.36).

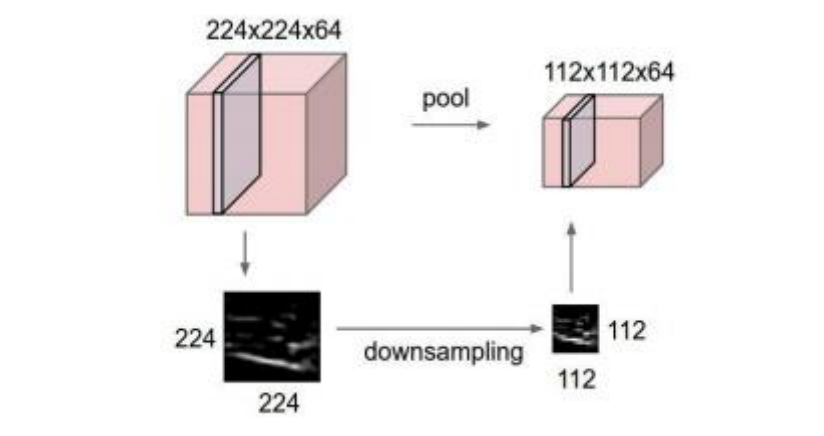


Şekil 2.36: 2D Evrişim Katmanı [48].

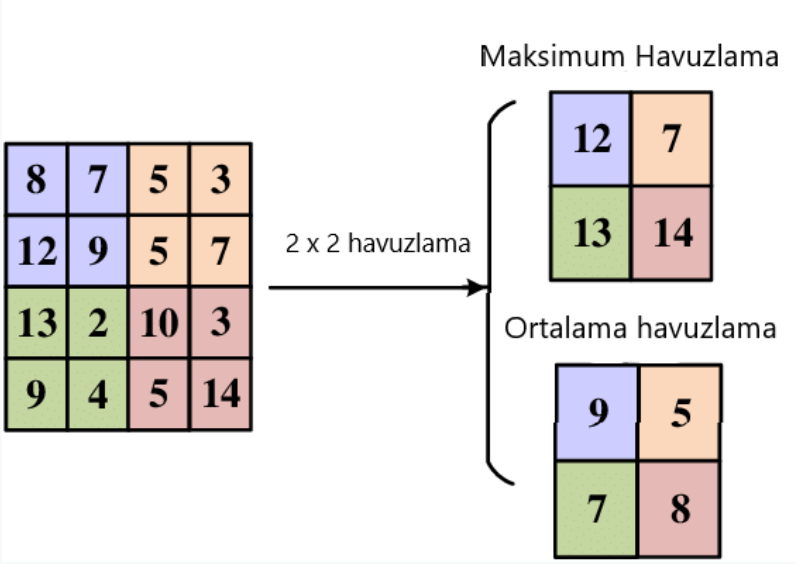
2.5.2. Havuzlama Katmanı (Pooling layer)

Pooling katmanı, Convolved Feature uzamsal boyutunu küçültmekten sorumludur. Bu, isim verisinin Eşanlamlılarını / Hipernimlerini (Tahmini Frekansa Göre Sıralanmıştır) işlemek için gereken hesaplama gücünü boyutsallık azaltma yoluyla azaltmak için yararlıdır.

Ayrıca, rotasyonel ve konumsal değişmez olan baskın özellik makalesinin çıkarılması için yararlıdır, bu, mankenin etkin bir şekilde hazırlanma sürecini sürdürür. İki popüler tür vardır; Madde kısmından en yüksek değeri döndüren kolay yerleşimli havuzlama ve bölümün tüm değerlerinin ortalamasını döndüren ortalama havuzlama (Şekil 2.37). Örneğin, gruplandırma 2x2 boyutundaysa, çıktıda girdi verisinin boyutunun dörtte birini elde edilir. Şekil 38 'da bu dönüşüm tekniğine rağmen uzamsal ilişkinin korunduğu görülmektedir.



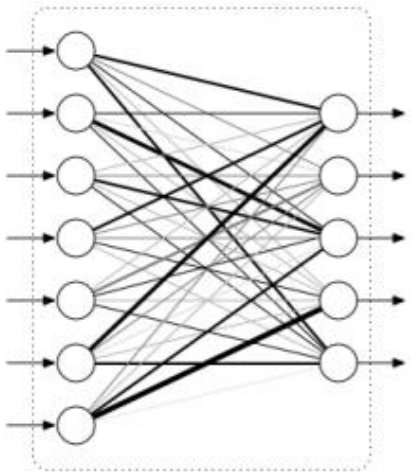
ŞEKİL 2.37: Pooling layer [48].



Şekil 2.38: havuzlama katmanı operasyon yaklaşımları [48].

2.5.3. Tam Bağlantılı Katman (Fully Connected Layer)

Fully Connected katmanda birkaç kez evrişimli katmandan ve pooling katmanından geçen ve matris halinde olan görselimiz düz bir vektör haline getirilir. Bu özellik, giriş için hayati önem taşıyan düz kargo/vektör bilgilerini içermektedir. Ağ eğitildiğinde, bu özellik vektörü ayrıca sınıflandırma, regresyon veya SVM gibi başka bir tür çıkış sinyaline çevrilmek için başka bir ağa giriş için kullanılmaktadır. FC katmanından önceki bir burulma yatağı, görüntüdeki yerel özelliklerle ilgili bilgileri içermektedir. Kenarlar, noktalar, şekiller vb. girdiler her evrişimsel katman, yerel bir özelliği temsil eden birkaç filtre içerir. FC katmanı, en önemli evrişim katmanlarının hepsinden toplanan karmaşık bitki ve seçici bilgileri içermektedir. Girdi resmimizi sinir ağları ile eğitebileceğimiz kıvama getirdikten sonra geriye sadece klasik sinir ağlarındaki çalışma mantığı kalıyor. Yine katmanlardaki nodelarda (düğüm) özellikler tutuluyor ve weight (ağırlık) ve bias değiştirilerek öğrenme sürecine giriliyor. Mimari olarak da adlandırılan sinirsel elektronik ağ yapıları, çoklu katmanlara, farklı katman türlerine, aktivasyonların matematiksel işlevine, boyutuna sahiptir. En iyi sonuçlara ve daha az işlem ve parametreye sahip olmak için hesaplamalı parasal değerden tasarruf etmek için çok önemli bir faktördür [52].



Şekil 2.39: Tam Bağlantılı Katman [52].

Şekil 2.39 ‘de darboğaz MobileNetV2 için önerilen bir tür evrişimli bloktur.
Temel olarak bu tarz bir yapıya sahip CNN algoritması üzerinde sürekli geliştirmeler yapılıyor.
CNN mimarilerinin en yaygın türü Very Deep Convolutional (VGG), Resent, Densenet, Squeezenet, MobileNetV2, LeNet, GoogLeNet, AlexNet ve ZFNet'tir [47].

2.6. DERİN ÖĞRENME TEKNİKLERİ

2.6.1. Transfer Öğrenimi

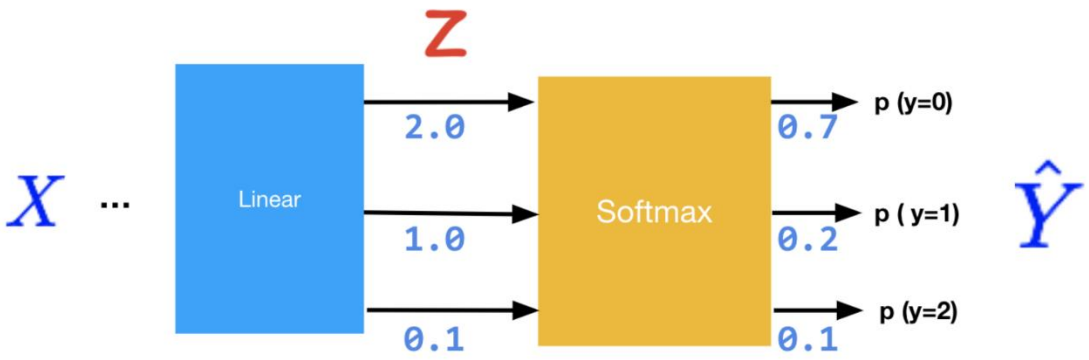
Makine öğrenmesi (Machine Learning) alanında uzman kişiler tarafından oluşturulan ve önceden büyük bir veri kümesi (Dataset) üzerinde eğitilmiş bir modelin yeniden kullanılmasına dayanan bir teknik. Transfer öğrenme (Transfer learning), makine öğrenmesi yöntemlerinin bir problem çözülürken elde edilen bilgileri depolaması ve başka bir problemle karşılaşıldığında bu bilgiyi kullanmasıdır. Transfer learning yaparken, önceden eğitilmiş modelin sadece son katmanı yeni modele uygun şekilde değiştirilir [47].

2.6.2. Veri Büyütme

Veri büyütme, farklı yönlere sahip yeni "veriler" yaratmanın bir yoludur. Bunun avantajları iki yönlüdür, birincisi sınırlı verilerden "daha fazla veri" üretme yeteneği ve ikincisi geçersiz kılmanın önlenmestir. Sınırlı verilerle çalışmanın kendi sorunları vardır, veri kümelerinin kullanımı ancak toplama yöntemleri mevcut veri kümesini iyileştirdiği takdirde olumlu sonuçlar verebilir [47].

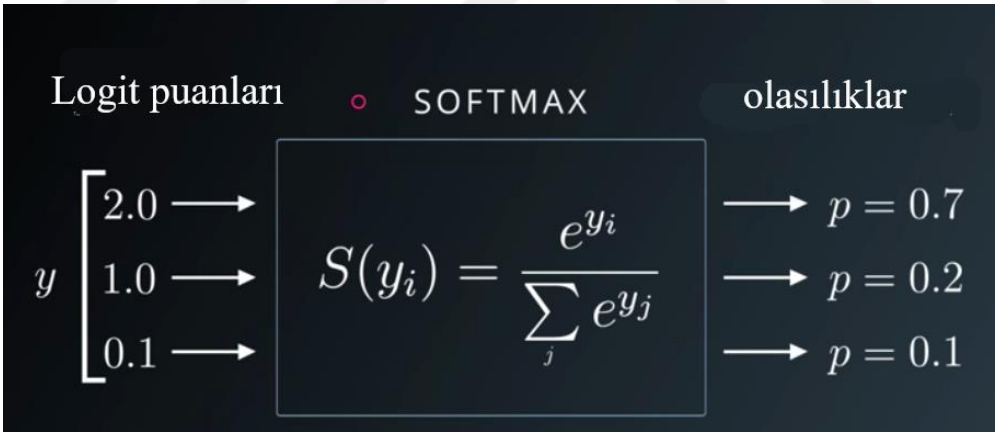
2.6.3. Softmax Sınıflandırıcı

Softmax, bir sayı vektörünü bir olasılık vektörüne dönüştüren matematiksel bir fonksiyondur; burada her bir değerin olasılığı, vektördeki her bir değerin göreli ölçeğiyle orantılıdır. Softmax fonksiyonunun uygulamalı makine öğreniminde en yaygın kullanımı, bir sinir ağı modelinde aktivasyon fonksiyonu olarak kullanılmasıdır. Spesifik olarak, ağ, sınıflandırma görevindeki her sınıf için bir tane olmak üzere N değerleri verecek şekilde yapılandırılır ve çıktıları normalleştirmek için softmax işlevi kullanılır, bunları ağırlıklı toplam değerlerden toplamı bir olan olasılıklara dönüştürür. softmax fonksiyonunun çıktısındaki her değer, her sınıf için üyelik olasılığı olarak yorumlanır. ImageNet gibi veri kümeleri için, genellikle Evrişimli Sinir Ağlarının 5. sıra doğruluğuna bakılır (burada yer gerçeği etiketinin, belirli bir giriş görüntüsü için bir ağ tarafından döndürülen en iyi 5 tahmin edilen etikette olup olmadığını kontrol ederiz). (1) gerçek sınıf etiketinin ilk 5 tahminde olup olmadığını ve (2) tahmin edilen etiketle ilişkili olasılığı görmek güzel bir özelliktir [55].



Şekil 2.40: Softmax Sınıflandırma Özellikleri [56].

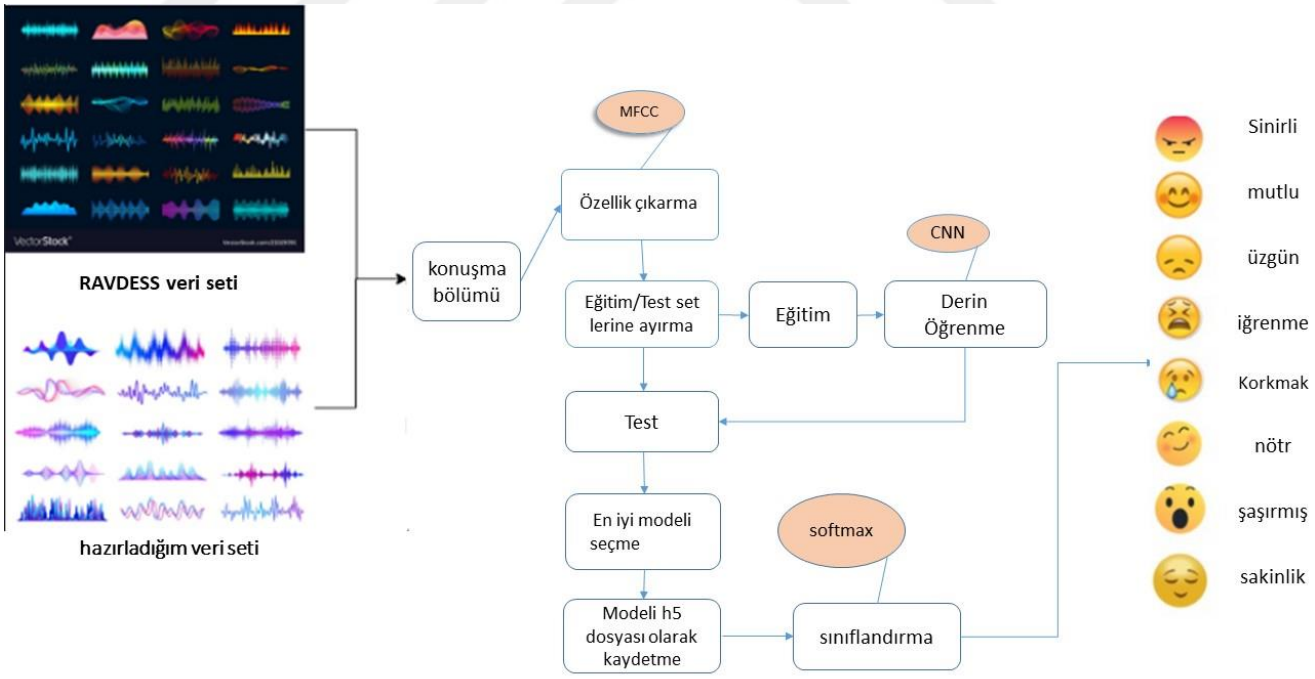
Softmax, Çoğu zaman çoklu Sigmoid olarak da bilinen bu fonksiyon, çok sınıflı hedef değişkeni içeren sınıflandırma problemleri (multi-class classification) için uygun bir aktivasyon fonksiyonudur. Softmax, çıktı olarak her sınıfa ait olasılık sonucu döndürür. Softmax sınıflandırıcı, Lojistik Regresyonun ikili formunun bir genellemesidir. Menteşe kaybında veya kare menteşe kaybında olduğu gibi, eşleme fonksiyonu f, bir girdi X veri seti alacak ve bunları, veri X ve ağırlık matrisi W'nin basit (doğrusal) bir nokta çarpımı aracılığıyla çıktı sınıfı etiketlerine eşleyecek şekilde tanımlanır [56].



Şekil 2.41: Softmax Özellikleri [56].

3. MALZEME VE YÖNTEM

Konuşulan dil aracılığıyla insan iletişimi, Bilgi alışverişinin temelidir ve ilk insan yerleşimlerinden bu yana toplumun ana yönüdür. Aynı şekilde duygular da, bugün bildiğimiz ve ilk doğal iletişim stratejisi sayılabilecek konuşma dilinden önceki ilkel bir içgüdüye kadar uzanır. Proje, konuşmacının konuşurken ortaya çıkardığı duyguların tespiti ile ilgilidir. Örnek olarak, korku, öfke veya sevinç durumunda üretilen konuşma, daha yüksek ve daha geniş bir perde aralığı ile yüksek ve hızlı hale gelirken, üzüntü veya yorgunluk gibi duygular yavaş ve düşük perdeli konuşma üretir. Ses kalıbı ve konuşma kalıbı analizi yoluyla insan duygularının tespiti, insan-makine etkileşimlerine daha iyi yardımcı olmak gibi birçok uygulamaya sahiptir. Özellikle, derin sinir ağlarına (CNN'ler) dayalı konuşmalar tarafından ortaya çıkarılan bir duygu sınıflandırma modeli, Mel Frekans Cepstral Katsayısı (MFCC) gibi akustik özelliklere dayalı MLP Sınıflandırması sunuyoruz. Model sekiz farklı duyguyu (nötr, sakin, mutlu, üzgün, kızgın, korkulu, iğrenme, şaşkınlık) sınıflandırmak üzere eğitilmiştir (şekil 3.1).



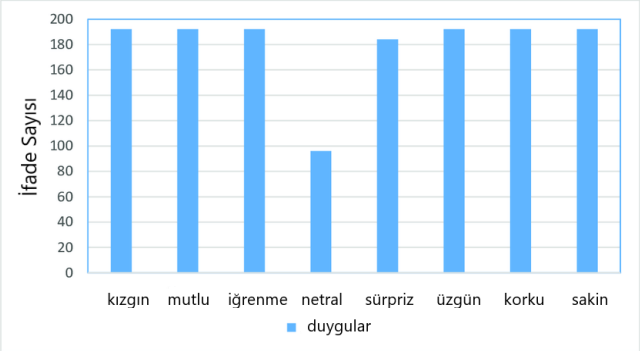
Şekil 3.1: Derin öğrenme sistemini kullanarak Konuşmadaki duyguları tanımak için akış şeması tasarımı.

3.1. Veri kümeleri

Derin nöral mimarilerin denetimli bir şekilde eğitilmesiyle ilgili temel sorunlardan biri, verimli öğrenme için gereken hesaplama çabası ve etiketlenmiş veri miktarıdır. Birincisi, donanımdaki gelişmeler ve genel amaçlı GPU hesaplama ile evrensel bir temelde ele alınırken, ikincisi alana çok bağlıdır. Ne yazık ki, çevresel kayıtların halka açık veri kümeleri hala çok sınırlıdır. Manuel açıklamanın yüksek maliyeti göz önüne alındığında, bu oldukça anlaşılabilir bir durumdur. Durum, yeni kayıt koleksiyonlarının sunulmasıyla kademeli olarak iyileşse de, bu alanda yeni veri yoğun yaklaşımların geliştirilmesinin önündeki en büyük engellerden biri olmaya devam etmektedir. Bu özellikle önemlidir, çünkü denetimli derin modellerin performansı, öğrenme için mevcut veri kümesinin boyutundan güçlü bir şekilde etkilenir. Bu nedenle, orijinal araştırma problemi bu bağlamda genişletilmelidir.

Evrişimli sinir ağları, sınırlı miktarda eğitim verisi ile çevresel ses sınıflandırması için etkin bir şekilde kullanılabilir. Modellerin değerlendirilmesi için halka açık en iyi veri setlerinden biri olan RAVDESS seçildi (Şekil 3.2).

RAVDESS veri seti, bu projenin kapsamını, bir ses dosyasından başlayarak konuşmacının duygularını tahmin etmek için bir sınıflandırıcı oluşturmaktır. Bu görev için RAVDESS veri setinden 4948 kısa (5 saniye) örnek kullanıldı. Tahmin etmeye çalıştığımız sınıflar şunlardır: (01 = nötr, 02 = sakin, 03 = mutlu, 04 = üzgün, 05 = kızgın, 06 = korkmuş, 07 = iğrenme, 08 = şaşırmış). RAVDESS'in yapısı ilerleyen sayfalarda açıklanmıştır. Kuzey Amerika İngilizcesinde dinamik, çok modlu bir yüz ve ses ifadeleri seti olan RAVDESS 7356 dosya içermekte olup farklı 200 dosya ise bu tez çalışması için sonradan eklenmiştir. Veriler karşılaştırılabilir çapraz doğrulama için 10 kat halinde düzenlenmiştir (10-fold Cross-Validation).



Şekil 3.2: RAVDESS veri seti [41].

Örnekler;

- Yalnızca ses dosyaları;
- Video + ses dosyaları: Projenin ana dizininde bulabileceğiniz Mp4ToWav.py betiğini kullanarak her dosyadan sesi çıkarıldı.

(RAVDESS) CC BY-NA-SC 4.0 altında lisanslanmıştır. 7356 RAVDESS dosyalarının her birinin benzersiz bir dosya adı vardır. Dosya adı 7 parçalı bir sayısal tanımlayıcıdan oluşur (örneğin, 02-01-06-01-02-01-12.mp4). Bu tanımlayıcılar uyarıcı özelliklerini tanımlar: [42].

Dosya adı tanımlayıcıları

• Modalite (01 = tam AV, 02 = yalnızca video, 03 = yalnızca ses).

• Vokal kanal (01 = konuşma, 02 = şarkı).

• Duygu (01 = tarafsız, 02 = sakin, 03 = mutlu, 04 = üzgün, 05 = kızgın, 06 = korkulu, 07 = iğrenme, 08 = şaşırmış).

• Duygusal yoğunluk (01 = normal, 02 = güçlü). NOT: "Nötr" duygu için güçlü bir yoğunluk yoktur.

• Açıklama (01 = “Kapının yanında çocuklar konuşuyor”, 02 = “Kapının yanında köpekler oturuyor”).

• Tekrar (01 = 1. tekrar, 02 = 2. tekrar).

• Aktör (01'den 24'e kadar. Tek numaralı oyuncular erkek; çift numaralı oyuncular kadın).

Dosya adı örneği: 02-01-06-01-02-01-12.mp4

• Yalnızca video (02)

• Konuşma (01)

• Korkunç (06)

• Normal yoğunluk (01)

• “Köpekler” ifadesi (02)

• 1. Tekrar (01)

• 12. Oyuncu (12)

• Kadın, aktör kimlik numarası çift olduğu için.

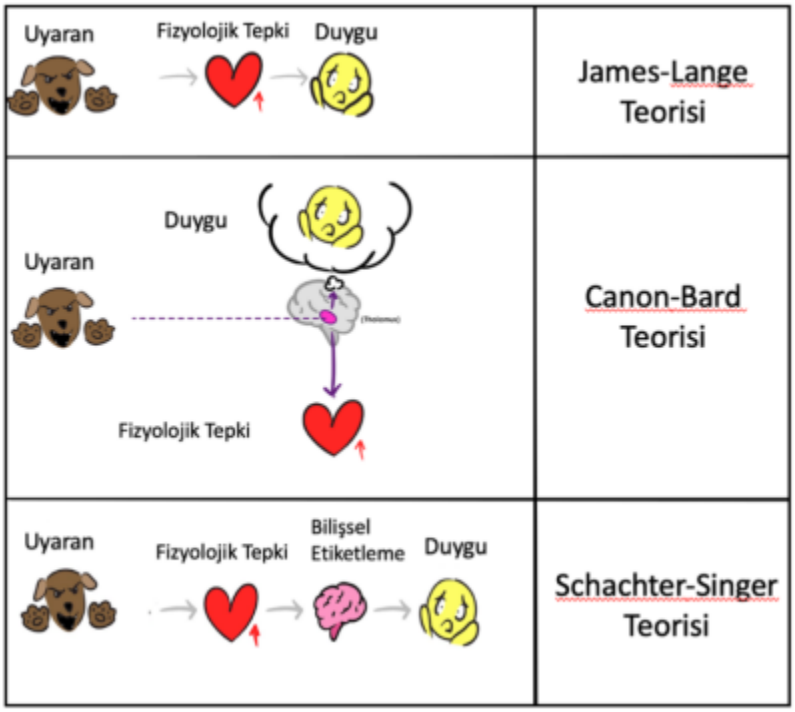
2.2. Duygular

Duygu, beynin gerçekleştirmesi gereken en karmaşık şeylerden biridir ve duygunun en önemli itici güçlerinden biri de sestir. Ve bu kadar önemli olmasının nedeni, bilişsel radarımızın altında çalışmasıdır. Araştırmalar uzun süredir majör akorlarda köklenen seslerin olumlu duygular üretme eğiliminde olduğunu ve minör akorlarda köklenen seslerin olumsuz duygular ürettiğini göstermiştir. Öte yandan, günlük hayatta duyduğumuz sesler genellikle dinleyiciye duygusal tepkiyi etkileyen bir şeyi temsil eder. Modern çağda, insanlar sürekli olarak çeşitli reklamlar, uyarılar ve bildirimler ile çevrilidir ve bu seslerin çoğu, arzu edilen bir duygusal tepkiyi üretmeyi amaçlayan akor nitelikleri ile kasıtlı olarak tasarlanmıştır. Her gün duyduğumuz yüzlerce tek tip sesin belirli bir anlamı veya amacı vardır ve her birinin dinleyiciye doğru mesajı iletmesi önemlidir. Bu proje, öznelerin seslere karşı duygusal tepkilerinin öncelikle seslerin akor kalitesi tarafından mı (başlangıçta ses tasarımcılarının amaçladığı gibi) belirlendiğini veya bu seslerin yaşam olaylarıyla ilişkilendirilmesinin seslerin akor niteliklerini mi geçersiz kıldığını belirlemek için tasarlanmıştır [57].

Schachter-Singer teorisi, duyguların esasen aynı mekanizma tarafından serebral kortekste uyarıldığını ve işaretlendiğini iddia eder [58].

James-Lange teorisi ile zıt gerçekleri savunan bu teoride duygunun uyaranın yoğunluğu tarafından belirlendiği ve bilişsel değerlendirmenin korteks tarafından yapıldığı söylendiği için ikinci faktör teorisi olarak da adlandırılır [58].

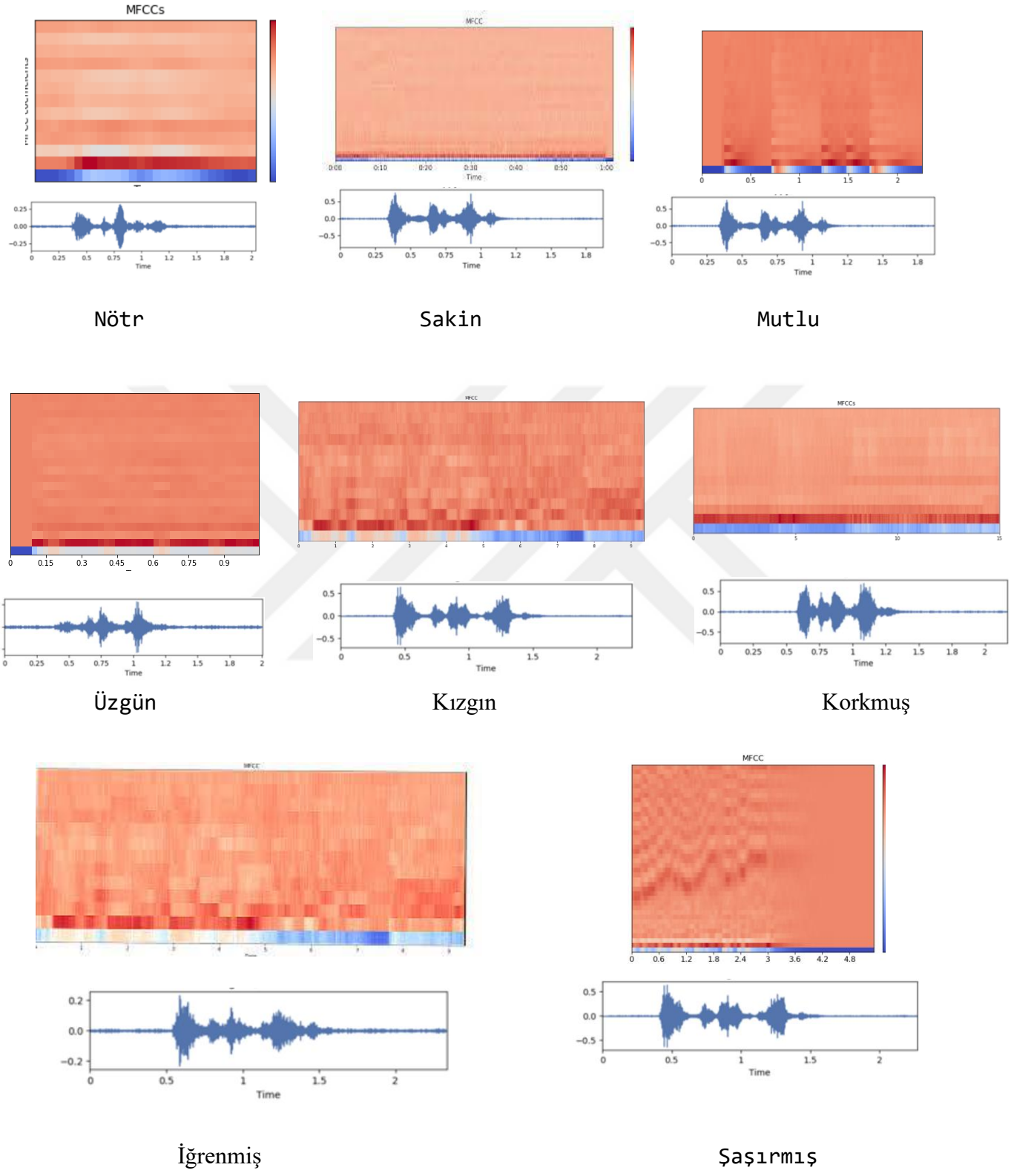
Benzer deneyimler sonucunda ortaya çıkan duygulara farklı tepki mekanizmaları tepki gösterse de etkinin kaynağı benzer olduğu için kafa karışıklığına neden olabilir. Bu, bilişsel süreçlerin duygulara dayalı olarak aktif olduğunu doğrulamaktadır (Şekil 3.3).



Şekil 3.3: Oluşumsal duygu teorileri [58].

Sesler, insanlar belirli akorları olumlu, diğerlerini olumsuz olarak deneyimlemek için kablolu olabilir. Ancak bir şarkıyı birden çok kez duymak etkisini değiştirir ve tanıdık bir sesin tanıdık olmayan bir sesten farklı bir duygusal etkisi vardır. Belirlenen duygu kalıpları 22 farklı duygu kategorisine karşılık gelmektedir: hayranlık, güzelliğin takdir edilmesi, eğlence, öfke, kaygı, huşu, beceriksizlik, can sıkıntısı, sakinlik, kafa karışıklığı, özlem, iğrenme, empatik acı, mest olma, heyecan, korku, dehşet, tiksinme, ilgi, neşe, nostalji, rahatlama.

Bu projede 8 farklı duygu kategorisi kullanılmıştır, aşağıda gösterilen ses -dalga grafikleri ile görselleştirilmiştir. Belirtilen grafiklerde kullanılan ses dalgaları hazır olarak alınıp sisteme girdi olarak verilmiştir. Her ses kaydının dalga görüntüsü derin öğrenmenin işleyebilmesi için MFCC spektrograma çevrilmiştir. RAVDESS grubu 8 farklı duygu için “Hello, how are you” cümlesi dalga ve spektrogram görüntüleri yer almaktadır. Bu projede 8 farklı duygu kategorisi kullanılmıştır, aşağıda gösterilen ses -dalga grafikleri ile görselleştirilmiştir. Belirtilen grafiklerde kullanılan ses dalgaları hazır olarak alınıp sisteme girdi olarak verilmiştir. Kodlar 8 temel duygu (nötr, sakin, mutlu, üzgün, kızgın, korkulu, iğrenme, sürpriz) kullanılarak oluşturulmuştur (şekil 3.4).

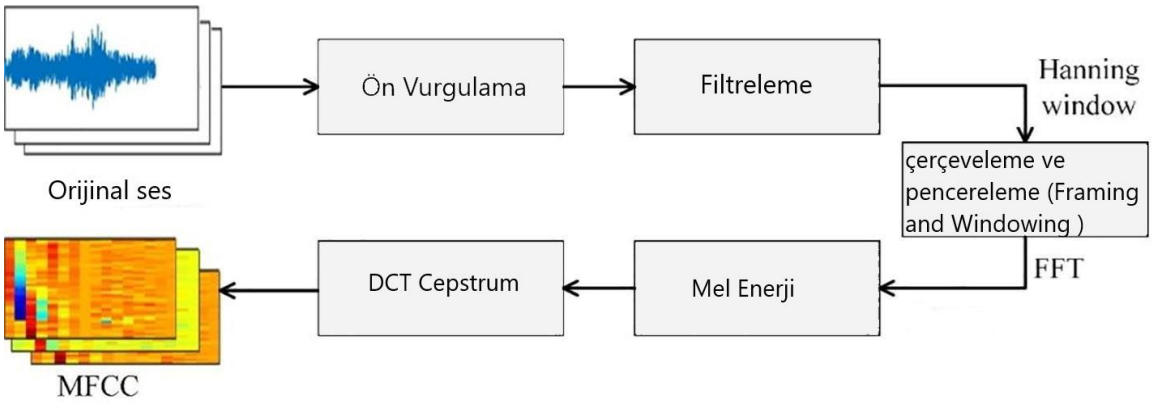


Sekil 3.4: duyguda dalga ve spektrogram görüntüleri.

Duygu tanımanın önemi, kullanıcı deneyiminin iyileştirilmesi ve Sesli Kullanıcı Arayüzlerinin (VUI'ler) katılımıyla popüler hale gelmektedir. Konuşmaya dayalı duygu tanıma sistemleri geliştirmenin pratik uygulama faydaları vardır. Bununla birlikte, bu faydalar, sistem pratik uygulamalarda kullanıldığında, konuşmaya dayalı duygu tanıma performansını bozan gerçek dünya arka plan gürültüsü tarafından bir şekilde reddedilir. Konuşma Duygusu Tanıma (SER), konuşma sinyali analizi alanındaki en zorlu görevlerden biridir, konuşma sinyallerinden duygu çıkarmaya çalışan bir araştırma alanı problemidir.

3.3. Mel Frekans kepstral katsayıları (MFCC)

MFCC, otomatik konuşmacı ve konuşma tanımada yaygın olarak kullanılan bir özelliktir. Mel-frekans ölçeği olarak ifade edilen enerji spektrumunun kısa vadeli gerçek log-kosinüs dönüşümünün sonucudur [60]. Şekil 3.5, MFCC çıkarma işlemini göstermektedir. İlk olarak, orijinal ses bir ön-vurgu filtresi ve bir geçiş bandı filtresi ile ön işleme tabi tutulmuştur. Ardından, önceden işlenmiş sinyal çerçevelere bölündü ve bir pencere eklendi. Her çerçeve üzerinde hızlı Fourier dönüşümü (FFT) işlemi yapıldı ve spektrumun karesi alındı. Daha sonra sonuç, Mel-enerjisini elde etmek için bir dizi Mel-filtresiyle süzüldü. Mel enerjisinin logaritmasını kullandık ve ardından ayrık kosinüs dönüşümü (DCT) gerçekleştirdik. Son olarak, MFCC'yi türettik.



Şekil 3.5: MFCC çıkarma işlemi [61].

3.4. Evrişimli Sinir Ağı Özelikleri

3.4.1. Hiperparametreler

Hiperparametreler, makine / derin öğrenme modelinizi oluştururken çevirebileceğiniz düğmelerdir. Hiperparametreler, eğitime başlamadan önce önceden belirlenmiş bir değerle manuel olarak ayarlanan tüm eğitim değişkenleridir. Öğrenme Oranı, Bırakma Oranı ve model tasarımı hiperparametreler olarak kabul edilir. Ağırlıklar olarak adlandırılan verilerden öğrenilen değişkenler olan eğitim sürecinden elde edilen parametreler. Bu ağırlıklar model parametreleri olarak bilinir. Şekil 3.6, seyreltme (Dropout) oranının %0 ila %100 arasında olduğunu ve öğrenme Hızı (Learning Rate) 10-6 ila 10-1 arasında olduğunu göstermektedir.



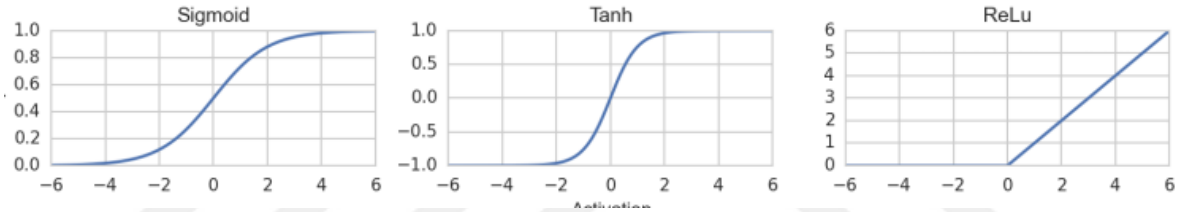
Şekil 3.6: Hiperparametreler - "düğmeler" veya "kadranlar" metaforu [61].

Model mimarisini tanımladıktan sonra bir sinir ağının hiper parametrelerini ayarlamak. Model mimarisi içinde, belirli bir katmandaki düğüm sayısı ve Seyreltme oranı için değişkenler dahil edilecektir. Ayrıca, optimize edicinin öğrenme oranı da dahil edilecektir. Model oluşturulduktan sonra hiperparametre ayarlayıcısına geri döndürülecektir. Ayarlayıcı daha sonra sinir ağını eğitim verilerimize uyduracak, değerlendirecek ve puanı döndürecektir. Denemeler tamamlandı, hiperparametre ayarlayıcı hangi hiperparametrelerin en iyi doğruluğu sağladığını gösterecek.

3.4.2. Aktivasyon Fonksiyonu

Aktivasyon fonksiyonları modele non-linearite katıyor. Gizli katmanlarda (Hidden layer'da) y = f(x,w) şeklindeki lineer fonksiyonumuzda matris çarpımı yapılıp nöronların ağırlığı hesaplandıktan sonra çıktı doğrusal olmayan (non-linear) bir değere dönüştürülür.

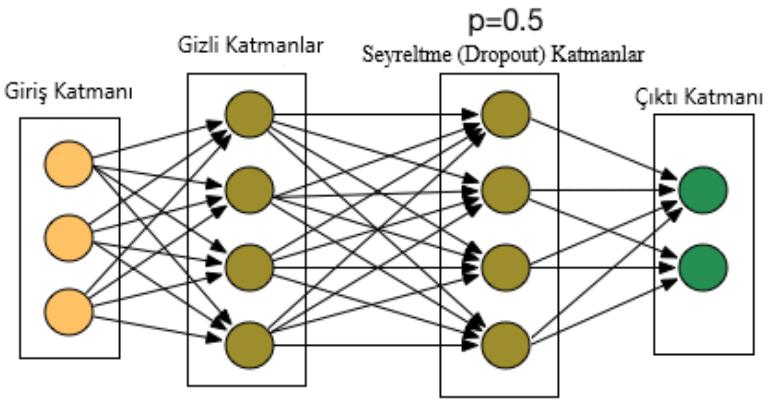
Aktivasyon fonksiyonları çok katmanlı yapay sinir ağlarında doğrusal olmayan (non-linear) dönüşüm işlemleri için kullanılmaktadır. Gizli katmanlarda (Hidden layer'larda) geri türev alınabilmesi (gradient decent hesaplanabilmesi) için (öğrenmede fark geri türevle alınıyor) gizli katmanların (hidden layer) çıktısı bazı aktivasyon fonksiyonları ile normalize edilmektedir. Bu aktivasyon fonksiyonlarından bazıları Şekil 3.7'de fonksiyon grafikleri verilen sigmoid, tanch, ReLu, PreLu vb'dir. İçlerinde en kullanışlısı ReLu'dur. ReLu'da sigmoid'e göre parametreler daha hızlı bir şekilde öğrenilmektedir. PReLu ise, ReLU'nun kaçırdığı negatif değerleri yakalamaktadır; eğer bizim için negatif değerler önemliyse PReLu tercih edilmelidir.



Şekil 3.7: En bilinen aktivasyon fonksiyonlarının grafikleri [61].

3.4.3. Seyreltme (Dropout) Değeri ve Seyreltme Uygulanacak Katmanlar

Tam bağlı (Fulyy connected) katmanlarda belli eşik değerin altındaki düğümlerin seyreltilmesinin başarımı arttırdığı gözlenmiştir. Yani zayıf bilgilerin unutulması öğrenimi arttırdığı gözlenmiştir [61].



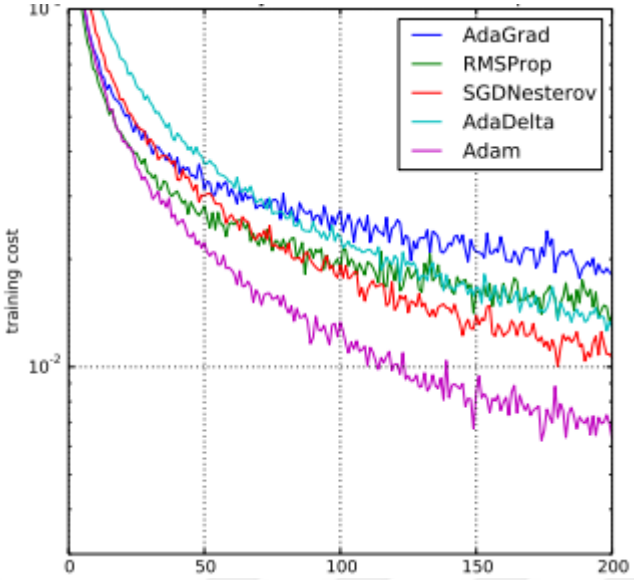
Şekil 3.8: Örnek dropout uygulaması

Seyreltme (Dropout) değerinin bazı özellikleri aşağıdaki gibidir;

- Seyreltme (Dropout) değeri olarak genelde 0.5 kullanılmaktadır. Farklı kullanıldığı durumlar da yaygındır. Probleme ve verisetine göre değişiklik göstermektedir.
- Seyreltme (Dropout) için rastgele eleme yöntemi de kullanılabilir.
- Seyreltme (Dropout) değeri eşik değeri olarak kullanıldığında [0,1] aralığında bir değer olarak tanımlanmaktadır.
- Tüm katmanlarda aynı dropout değeri kullanılması zorunlu değildir; farklı seyreltme (Dropout) değerleri de kullanılabilir.

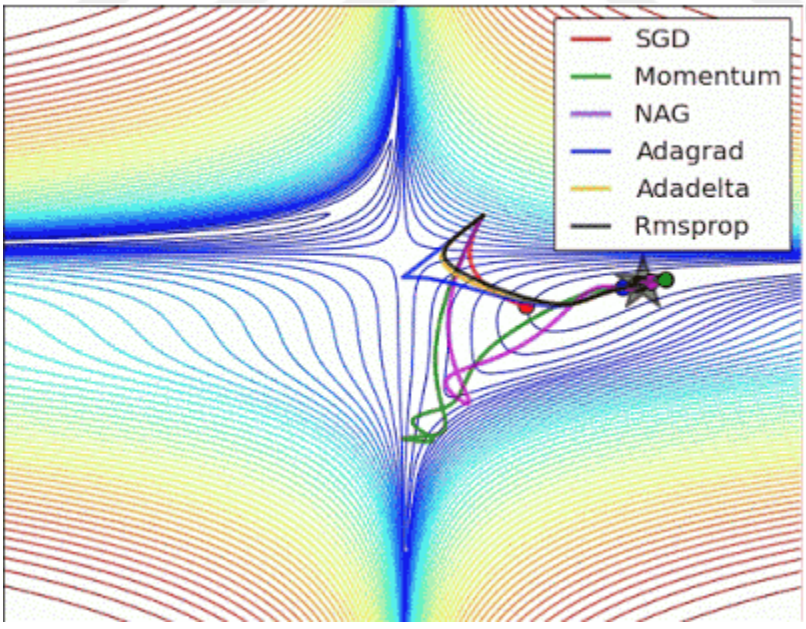
3.4.4. Optimizasyon Algoritmasının Seçimi

Derin öğrenme uygulamalarında öğrenme işleminin temelde bir optimizasyon problemi olduğu daha önce vurgulanmıştı. Doğrusal olmayan problemlerin çözümünde optimum değeri bulmak için optimizasyon yöntemleri kullanılmaktadır. Derin öğrenme uygulamaların yaygın olarak stochastic gradient descent, adagrad, adadelta, adam, adamax gibi optimizasyon algoritmaları kullanılmaktadır. Bu algoriatmalar arasında başarım ve hız bakımından farklılıklar bulunmaktadır [61].



Şekil 3.9: Belli başlı optimizasyon algoritmalarının MNIST veriseti üzerindeki çalışma zamanı grafiği

Belli başlı optimizasyon algoritmalarının MNIST veriseti üzerindeki çalışma zamanı grafiği Şekil 3.9'deki grafikte görülebilir.



Şekil 3.10: Optimizasyon algoritmalarının hız performans gösterimi (Resim Kaynak : Alec Radford) [61].

Optimizasyon algoritmalarının bazı özellikleri aşağıdaki gibidir;

- İki tür gradient hesaplama yöntemi var; sayısal (numeric) ve analitik (analytic). Sayısal gradient bize yaklaşık sonuç veriyor, yavaş çalışıyor bununla birlikte uygulaması kolay. Buna karşı analitik gradient hızlı, tam sonuç veriyor ve hataya eğilimli. Pratikte analitik gradient ile hesaplama yapılıyor sonuçlar sayısal gradient ile kontorol ediliyor.
- Derin öğrenme modellerinde genelde varsayalan olarak kullanılan optimizasyon algoritması stochastic gradient descent (SGD).
- Bununla birlikte şekil 3.10'da görüleceği üzere schoastic gradient descent diğer yöntemlere göre daha yavaş çalışıyor.
- Stochastic Gradient Descent (SGD) bazı problemlerde çok kötü sonuç verebilmektedir; özellikle resim tanıma problemlerinde.
- Adaptive algoritmalar öğrenme hızını (learning rate) kendisi öğrenmektedir ve dinamiktir.
- AdaGrad seyrek parametreler için büyük güncellemeler yaparken sık parametreler için daha küçük güncellemeler yapar. Bu nedenle NLP ve resim tanıma gibi seyrek veriler için daha uygundur.
- AdaGrad'da her parametrenin kendi öğrenme hızı vardır ve algoritmanın özelliklerine bağlı olarak öğrenme oranı giderek azalmaktadır. Bu nedenle öğreneme oranı giderek azalır ve zamanın bir noktasında sistem öğrenmeyi bırakır. Bu AdaGrad'ın en büyük dez avantajıdır.
- RMSprop ve benzeri olan AdaDelta, AdaGrad'ın bu sorununu çözerek bu hızlı düşüşü önler.
- Adam veya adaptif momentum AdaDelta'ya benzer bir algoritmadır. AdaDelta'dan farklı olarak parametrelerin her birinin öğrenme oranlarının yanısıra momentum değişikliklerini de önbellekte (cache) saklar; yani RMSprop ve momentumu birleştirir.
- Adaptif algoritmalar hız açısından SGD'den daha iyi performans göstermektedir. Bununla birlikte SGD momentum ile kullanıldığında, SGD'nin başarım açısından adaptif yöntemlere göre daha yüksek başarım gösterdiği bazı durumlarda vardır.

3.4.5. Katman Sayısı ve Gizli Katmanlardaki Nöron Sayısı

Derin öğrenme yöntemini diğer yapay sinir ağlarından ayıran özellikle de karmaşık problemlerde iyi sonuç vermesini sağlayan en önemli özelliği katman sayısı. Derinlik kavramı da buradan geliyor zaten. Katman sayısı arttıkça modelin daha iyi öğrendiği gözlenmiş.

Bildiğim kadarıyla bunun neden böyle olduğu “matematiksel” olarak henüz ispatlanmadı(literaür o kadar hızlı ilerliyor ki, bugün çıkmış olabilir:). Ama neden olduğuna dair öngörülerimiz var. Örneğin resim tanıma problemlerinde modelin başlarında elde edilen görüntülerin incelenmesi ile çok katmanlı yapay sinir ağlarında başlarda veriye ait genel hatların öğrenildiği daha sonraki katmalarda gittikçe verinin yavaş yavaş diğer özelliklerinin öğrenildiği gözlendi.

3.4.6. Evrişimli Sinir Ağı (CNN) Kernel Boyutu

Evrişimsel sinir ağlarında her bir katmanda matris üzerinde işlem yapan kerneller kullanılmaktadır. Bunlar basit Gabor filtreleri gibidir. Bu kernellerin boyutu öğrenme üzerinde hayli etkilidir. Çünkü kernel boyutu ile ne kadar genişlikte verinin birbirini etkileyeceğine karar verilmektedir. Genelde 3x3, 5x5, 7x7 kerneller kullanılmaktadır. Büyük boyutlu kernel kullanılması evrişim uygulandıktan sonra oluşacak resmin küçük olmasına neden olacaktır. Bu durumda bilgi kaybına yol açtığından genelde 3x3 gibi küçük boyutlu kernel kullanılmaktadır. Kenar bulma işleminde işlem yapılan pikselin sağına-soluna üstüne-altına bakılabilmesi için genelde tek sayılardan oluşan merkezi olabilecek filtreler kullanılır [61].

3.4.7. Evrişimli Sinir Ağı (CNN) “Pooling” yöntemi: max, min, mean

Evrişimsel sinir ağlarında kullanılan kernellerin çıktısı üzerinde “pooling” işlemi yapılmaktadır. “Pooling” işlemi kernelde bulunan veriler için filtreleme tekniğidir. Sıkça kullanılan pooling yöntemleri; kernel içindeki matris’de bulunanların en büyüğünü alma “max pooling”, en küçüğünü alma “min pooling” ve ortalamasını alma “mean pooling”dir. Bunlar arasında en sık tercih edileni ve en başarılı sonuç verdiği iddia edileni “max pooling”dir [61].

3.5. Kullanılan yazılım

İlk olarak, paython'da pip kullanarak bazı bağımlılıkların kurulması gerekir. Librosa, Numpy, Soundfile, Scikit-learn, PyAudio Paython sistemindeki temel bağımlıklıklardır.

Kurulum: Aşağıda bulunan bağımlılık kurulum kodları Ek 2.1’de bulunmaktadır.

Tüm işlem hattı aşağıda belirtilen şekildedir (farklı bir makine öğrenimi işlem hattıyla aynı özellikleri göstermektedir):

1. Veri Kümesi Hazırlama: İlk aşamada veri kümesi indirip çıkartmaya uygun hale getirilir.

2.Veri Kümesi Yükleme: Bu işlem, konuşma sinyalinden güç, perde ve ses yolu konfigürasyonu gibi farklı özelliklerin elde edilmesi gibi ses özelliklerinin çıkarılmasını içeren veri kümesini Python'da yüklemekle ilgilidir, bu aşamada librosa kitaplığını kullanılmaktadır.

3. Model Eğitimi: Veri seti hazırlayıp yüklendikten sonra, uygun bir sklearn modeli üzerinde eğitilir.

4. Modeli Test Etme Kısmı: Modelin ne kadar iyi çalıştığı ölçülür.

İlk olarak, üzerinde çalışmak için bir veri kümesine ihtiyaç duyulur. Bunun için RAVDESS veri kümesi kullanılır. RAVDESS indirilir, üzerine ekstra olarak ses eklenir ve başarılı sonuç elde edilir.

Konuşma dalga biçimini nispeten daha düşük bir veri hızında bir parametrik temsil biçimine değiştiren çıkarma özelliklerini işleyen işlev oluşturulur:

Kurulum: Aşağıdaki kodlar Ek 2.2'de mevcuttur.

Sınıflandırmaya yardımcı olmayan gereksiz bilgiler içerebilecek ham dalga formu yerine konuşma özellikleri olarak MFCC, Chroma ve Mel Frequency Cepstrum kullanılmaktadır.

Mevcut olan verileri yüklemek için aşağıda bulunan işlev yazılmalıdır:

int2emotion sözlüğü bu veri kümesindeki mevcut duyguları içerir, ancak AVAILABLE_EMOTIONS sınıflandırmak istenilen duygulardır. Bu kısımda istenilen duygulan eklenebilir ya da çıkartılabilir.

Kurulum: Aşağıdaki kodlar Ek 2.3'de mevcuttur.

Veri kümesi hakkında bazı bilgilerin günlüğe kaydedilmesini gösteren ekran göruntusu şekil 3.11' dedir.

```
# print some details
# number of samples in training data
print("[+] Number of training samples:", X_train.shape[0])
# number of samples in testing data
print("[+] Number of testing samples:", X_test.shape[0])
# number of features used
# this is a vector of features extracted
# using extract_features() function
print("[+] Number of features:", X_train.shape[1])
```

```
[+] Number of training samples: 504
[+] Number of testing samples: 168
[+] Number of features: 180
```

Şekil 3.11: veri kümesi hakkında bulgular

Kurulum: Aşağıdaki kodlar Ek 2.4'de mevcuttur.

Yani, sistem temelde 300 birim içeren bir katmana, 256, 500 yineleme toplu iş boyutuna uyarlanabilir bir öğrenme hızına sahip tam bağlantılı (yoğun) bir sinir ağıdır (bu optimal hiperparametreler değildir, bu nedenle ince ayar yapılabilir).

Şekil 3.12'de parametreler kullanılarak çalıştırılmıştır: Model yeni yüklenen veri seti ile eğitilmiştir;

```
# train the model
print("[*] Training the model...")
model.fit(X_train, y_train)
```

```
[*] Training the model...

MLPClassifier(alpha=0.01, batch_size=256, hidden_layer_sizes=(300,),
              learning_rate='adaptive', max_iter=500)
```

Şekil 3.12: # Eğitim verlerin yüklenmiş

4. BULGULAR

4.1. CNN Değerlendirme Sonuçları

Model, aralıklı doğrulama seti olarak kullanılan tek bir eğitim katı ile 10 katlı çapraz doğrulama rejiminde değerlendirilmiştir.

Tablo 4.1, kıvrımlar arasında ortalama doğrulukla birlikte, değerlendirilen modelin varyantları ve temel uygulamalar tarafından elde edilen sınıflandırma doğruluğunu sunar. Her durumda, evrişimli bir sinir ağına dayalı modeller, özellikle RAVDESS'in daha çeşitli kategorilerinden ve projede kullanılan derleme veri setinden ses olaylarını sınıflandırırken (temel doğruluk: 78,79 %) manuel olarak tasarlanmış özellikleri kullanan ilgili uygulamalardan daha iyi performans göstermiştir. Daha uzun zaman ölçeklerinde çalışan modeller, daha kısa segmentlerde çalışan varyantlara göre hafif iyileştirmeler sunmaktadır. Eğitim kısmı %70, test kısmı %30 olarak sonuç vermiştir ve bu tablo 4.1 gösterilmiştir. f1-score, recall, precision and macro ortalamanın sonucu aşağıdaki Tablo 4.1'da mevcuttur. Tablo 4.1'de ise %70 eğitim ve %30 test stratejisinden oluşan karışıklık matrisinin (confusion matrix) sonucu gösterilmiştir.

Tablo 4.1: RAVDESS'in veri seti eğitim-test doğruluk oranları

Model	Grup	Doğruluk	Ağırlıklı F-1 Skoru
Derin Öğrenme	(%70-Eğitim Seti) (%30-Test Seti)	78,79%	80 %

```
# predict 25% of data to measure how good we are
y_pred = model.predict(X_test)

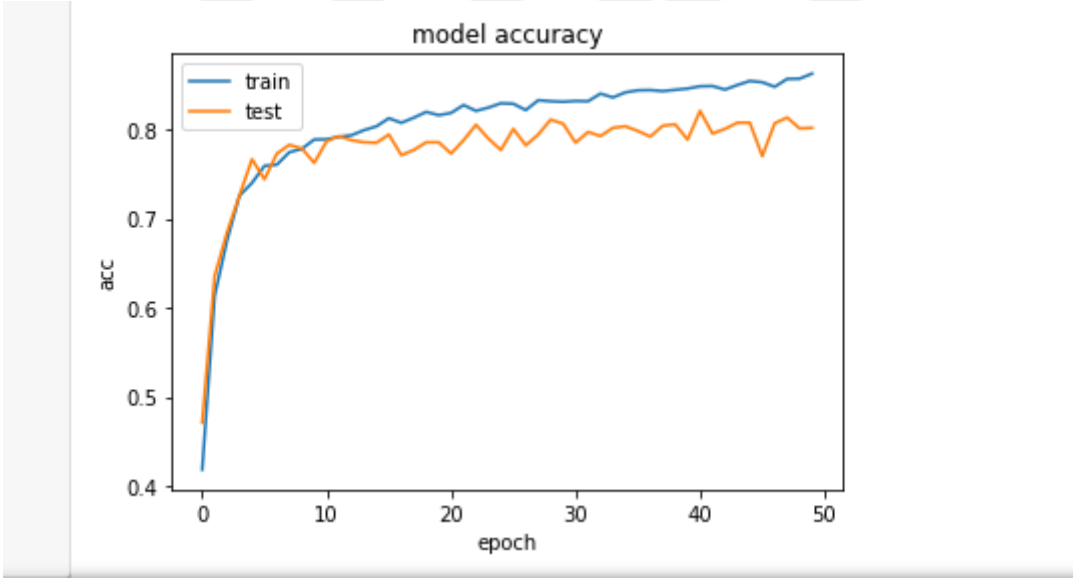
# calculate the accuracy
accuracy = accuracy_score(y_true=y_test, y_pred=y_pred)

print("Accuracy: {:.2f}%".format(accuracy*100))
```

```
Accuracy: 76.79%
```

Şekil 4.1: Sonuçların başarı oranı

Şekil 4.1, 4.2'de 8 duyguda %76,79 doğruluk elde edilmiştir. Kaydedilecek modelde ses ayarını test etmek için aşağıdaki kodun kullanılması uygun olur. 5 saniye konuştuktan sonraki kodun sonucu aşağıdaki gibidir.



Şekil 4.2: Eğitim kısmını (%70) ve deneysel kısmını (%30) gösteren grafik.

Şekil 4.3, 4.4 'da görüldüğü gibi python kodu çalıştığında program bizden minimum 5 saniye veya daha fazla konuşmamızı isteyecektir. Program sesi ilerletecek, ardından ses için bazı özellik çıkarımları ve sonuç olarak duygu tipinin sınıflandırılması uygulayacaktır.

```
if __name__ == "__main__":
    # load the saved model (after training)
    model = pickle.load(open("result/mlp_classifier.model", "rb"))
    print("Please talk")
    filename = "test.wav"
    # record the file (start talking)
    record_to_file(filename)
    # extract features and reshape it
    features = extract_feature(filename, mfcc=True, chroma=True, mel=True).reshape(1, -1)
    # predict
    result = model.predict(features)[0]
    # show the result !
    print("result:", result)
```

```
Please talk
result: happy
```

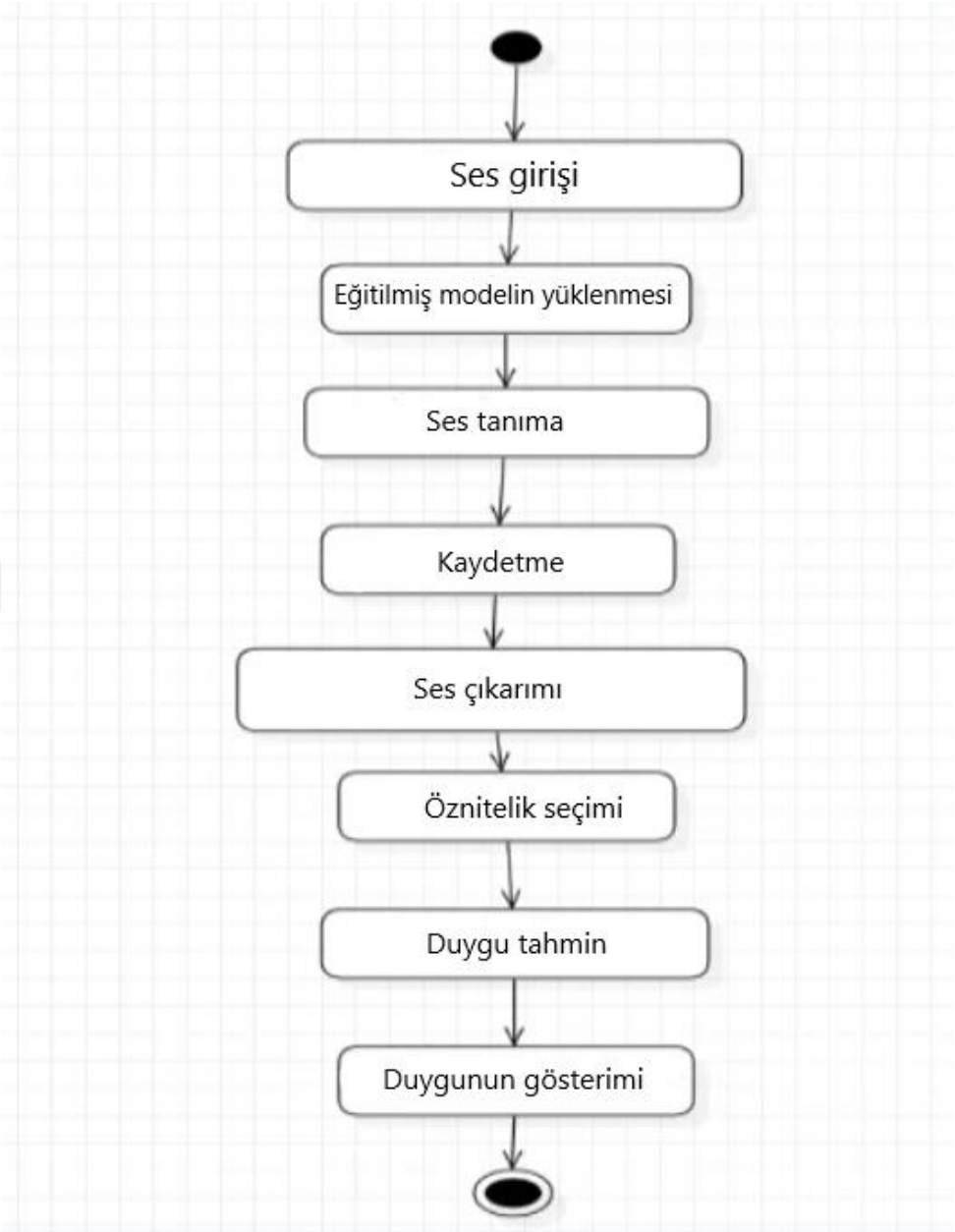
Şekil 4.3: 5 saniye konuştuktan sonraki kodun sonucu

```
if __name__ == "__main__":
    # load the saved model (after training)
    model = pickle.load(open("result/mlp_classifier.model", "rb"))
    print("Please talk")
    filename = "test.wav"
    # record the file (start talking)
    record_to_file(filename)
    # extract features and reshape it
    features = extract_feature(filename, mfcc=True, chroma=True, mel=True).reshape(1, -1)
    # predict
    result = model.predict(features)[0]
    # show the result !
    print("result:", result)
```

```
Please talk
result: angry
```

Şekil 4.4: 5 saniye konuştuktan sonraki kodun sonucu

Sistemin tüm sonuçları, şekil 4.5 'de gösterildiği gibi bazı farklı prosedürlerden geçtikten sonra ortaya çıkar. Konuşan kişinin sesi giriş sesi olacak ve sisteme yüklenecektir. ses, MFCC gibi bazı özellik çıkarımlarına sahip olacaktır. Sistem duyguyu tahmin edecek ve duygunun sonucu yazılı olarak gösterilecektir.



Şekil 4.5: Etkinlik Şeması ve Kullanım Durumu Şeması

Tablo 4.2 'de tablolaştırıldığı gibi, derin öğrenme, tüm performans ölçümlerinde geleneksel makine öğrenimi algoritmalarından daha iyi performans gösterir. Mevcut öğrenme probleminde derin öğrenmenin neden üstün olduğunu açıklayan iki ana neden belirledik. Birincisi, bir girdi olarak hem zaman alanlı kablosuz sinyale hem de CW dönüştürülmüş görüntüye sahip olmak, CNN+LSTM modeli için zengin bir öğrenme kaynağıdır, oysa ML algoritmaları, insan yargısının düzeyine duyarlı girdiler olarak çıkarılan öznitelikler ile eğitilir. özelliklerin yanı sıra

orijinal verilerden bariz bilgi kaybı seçimi. İkincisi, CNN'ler, doğru hiper-parametreler verildiğinde, hedef değerlerini yeniden yapılandırmaya yardımcı olan ham verilerde saklı olan en küçük bilgileri bile öğrenen kendi kendine öğrenen kişilerdir. ML algoritmaları, modele beslenecek ham verilerden anlamlı istatistiksel parametreleri (veya bir parametre kombinasyonunu) bulmak için insanlara bir şekilde bağımlıdır. Bununla birlikte, bu ML modelleri, uygulanan DL modelinin ne kadar iyi performans gösterebileceğini ölçmek için bir kriter olarak kullanılabilecek kabul edilebilir bir performans rapor etmektedir.

Tablo 4.2: f1-score, recall, precision and macro ortalamanın sonucunu

Duygular	precision	recall	f1-score	support
sakinlik	0.91	0.82	0.87	192
Mutlu	0.75	0.59	0.66	123
Üzgün	0.80	0.78	0.79	264
Sinirli	0.73	0.83	0.78	275
Korkmuş	0.81	0.88	0.84	252
İğrenmiş	0.81	0.81	0.81	241
Şaşırmış	0.75	0.84	0.79	197
Sakinlik	0.89	0.74	0.81	190

4.2. Confusion Matrix Sonuçları (Karışıklık matrisi)

Bu matrix üzerinden birbiri ile karışan duygular, yanlış pozitif veriler gözlenebilmektedir. Confusion matrixin Y etiketlenmiş duyguyu tahminini, X düzlemi ise modelin duygu göstermektedir. Profesyonel bireylerin %70'lik verisinin kullanıldığı eğitim seti confusion matrixi Tablo 4.3 'de gösterilmektedir. Matrixteki veriler her sütun ve satırın toplamı (1.0) sonucunu verecek şekilde dağılım göstermektedir. (1.0) modele o duygu için verilen verinin 100% 'ünü temsil etmektedir. Taplo da görüldüğü gibi en yüksek doğruluk oranı 'Sad' için elde edilmiştir. Daha sonra en güvenilir model etiketlemesi sırasıyla angry, happy, fearful, disgust, neutral, surprised ve calm için elde edilmiştir. 'Sad' duygusu için en yüksek yanlış pozitif 14% ile korkulu duygusuna aittir. Bu durum modele verilen bir 'Sad' verinin en çok korkulu duygusu ile karıştığını göstermektedir. Mutluluk için en yüksek yanlış pozitif 14% ile sinirlik duygusundadır. En yüksek yanlış yanlış pozitif iğrenme için 10% ile öfke, şaşkınlık için 16%

iğrenme, korku için 17% ile hüzün, iğrenme için 10% ile öfke ve öfke için 0.09 ile mutluluk duygusunda bulunmuştur. Tablo 4.3 ’de gösterilmektedir.

Tablo 4.3: Karışıklık matrisinin (Confusion matrix) sonucu.

Duygular	Nötr	Sakinlik	Mutlu	Üzgün	Sinirli	Korkmuş	İğrenmiş	Şaşırmış
Nötr	158	6	1	17	2	4	2	2
Sakinlik	6	73	12	20	0	3	8	1
Mutlu	2	11	207	8	14	11	8	3
Üzgün	0	3	8	228	11	14	9	2
Sinirli	2	1	6	4	223	4	10	2
Korkmuş	3	1	9	17	10	195	2	4
İğrenmiş	1	2	5	8	10	3	165	3
Şaşırmış	1	0	10	9	6	7	16	141

5. TARTIŞMA VE SONUÇ

Bu çalışmada, duygu durumlarının tanınmasında kullanılabilecek konuşma özellikleri tanımlanmıştır ve bu özellikleri kullanarak derin sinir ağları uygulanmıştır. DNN kullanılarak yapılan uygulamanın bir sonucu olarak, test doğruluğunun yaklaşık %78,79 olduğu doğrulanmıştır.

Gelecekte; tempo, perde ve eklenen diğer özelliklerle bir DNN uygulayarak doğruluğu arttırılacaktır. Ayrıca, bir evrişimli sinir ağı (CNN) kullanırken doğruluğun gelişip gelişmediğini kontrol etmek için sayısal veriler yerine Mel spektrogram görüntülerini kullanılacaktır. Görüntüleri ve sesi kullanarak bir sürücünün duygularını tanımak, AI platformunun çeşitli hizmetler sunmasını sağlayacaktır. Bir DNN modeli için büyük miktarda veri elde etmek gerekir. Bu nedenle, bir veri seti ekleyerek İngilizce ses verileriyle uygulanan derin öğrenme ağlarının modellerini de karşılaştırmak gerekmektedir. Veri miktarı küçükse, çeşitli konuşma sinyali modülasyon teknikleri kullanılarak büyütme gerekli olacaktır.

Elde edilen sonuçlara dayanarak, bazı duygu çiftlerinin birbiriyle kolayca karıştırılabileceği açıktır; nötr ve can sıkıntısı veya öfke ve neşe. Ses özellikleri, bu duygular arasındaki farkları ayırt etmek için yeterli değildi [62]. Deneylerinde, perde ve enerjinin boyutsal istatistiksel özelliklerini kullanarak 8 duyguyu %76.79 doğrulukla sınıflandırabilmişlerdir. Makaleye benzer bir genel doğrulukla, bir duygu için maksimum doğrulukları %100'ümüze kıyasla %81'di, ancak özellikleri her duyguyu minimum %64 doğrulukla sınıflandırabiliyordu. Oyuncuların yüz ifadelerini tespit etmek için pratik uygulamalar için uygun olmayan gerekli ekipmanları giymelerini gerektiren görsel yardım ile sonuçlar daha iyi (%89.1) yapılmıştır. Makalede [60], tekrarlayan sinir ağlarında boyutsal öznitelikler kullanılarak, maksimum %62.85 doğruluk elde edilmiştir [61], 384 boyutlu düşük seviyeli tanımlayıcıları kullandıkları yöntemlerinde, 5 duyguyu tespit etmede %42 ortalama kesinlik elde ettiler. Sınıflandırma için derin sinir ağları uyguladılar. Bu yaklaşımların her ikisi de sinir ağlarını kullandıkları için çok fazla hesaplama kaynağı gerektirir, ancak elde edilen doğruluk bizimkinden daha düşüktür. Chi-Chun Lee ve ark. [63], Bayesian Lojistik Regresyon modelini kullanarak 4 duyguyu sınıflandırmak için 40-60 global öznitelik kullandı ve bizimkinden önemli ölçüde daha düşük olan %56.5 doğruluk elde edildi. Doğrulukları düşük olmasına rağmen, kullandıkları veri tabanı, duygusal konuşmalar yerine doğal etkileşimlerden gelen konuşmaları içeriyordu ve bu

da yöntemlerini daha sağlam ve gerçek yaşam uygulamaları için uygulanabilir hale getirildi. Başka bir makalede [64], 40 Fourier Parametresi (FP) özelliği 6 duyguyu %75 doğrulukla sınıflandırabildi. Başlangıçta yöntem 1800 öznitelik çıkarmış ve bunlardan 40 FP özniteliğine normalize edilmiştir. Yöntemlerinden farklı olarak, herhangi bir normalizasyon olmadan ham MFCC öznitelikleri kullanıldı. Gizli Markov Modellerini (HMM) kullanarak, Schuller ve ark. [59], enerji aralığı gerçekleştirilmiş yerel öznitelikler ile %77,8 doğruluk ve 20 küresel öznitelik ile %86,8 doğruluk elde edebilmiştir. Bizimkine benzer şekilde, yerel özellikleri kullanırken kullandıkları yöntemler de korku ve iğrenme gibi belirli duyguları tespit etmede nispeten iyi performans gösterdi. Bununla birlikte, küresel özellikleri, duyguların çoğunu, maksimum %93,2 doğrulukla doğru bir şekilde algılayabiliyordu. Genel olarak yüksek doğrulukları, küresel özelliklerin yerel olanlara göre üstünlüğünü kanıtlamakta ve ayrıca HMM sınıflandırıcılarının duygu tespitindeki etkinliğini kanıtlamaktadır. Yaklaşımımız, özellik boyutunun düşük sayısı nedeniyle minimum hesaplama gücü gerektirir. Ayrıca, ayıklanan ham MFCC özellikleri, herhangi bir başka filtreleme veya işleme olmaksızın doğrudan modeli eğitmek için kullanıldı. Ayrıca, LMT modellerinin eğitimi, diğer birçok sınıflandırıcıdan daha az zaman harcar, özellikle de sinir ağlarının eğitiminden çok daha az zaman harcar [65]. Bununla birlikte, öznitelik boyutunun küçük boyutu, modelin doğruluğunu bir dereceye kadar tehlikeye atmış ve tüm duyguları iyi oranlarda sınıflandırmak için mücadele etmiştir. Sonuç olarak yöntemimiz, öfke, mutlu ve üzgün gibi ses bilgilerine daha bağımlı olan duyguların hızlı bir şekilde tespit edilmesini gerektirebilecek uygulamalar için iyi bir doğruluk düzeyi ile uygundur.

Bu çalışmamada, konuşma duygu tanıma sisteminin bazı uygulamalarından bazıları: Çağrı merkezinde görüşme, bir çağrı görevlisinin hizmet kalitesini iyileştirmeye yardımcı olan müşterilerle çağrı görevlilerinin davranışsal çalışmasını analiz etmek için kullanılabilir, Telefon görüşmelerinin duygu analizi suçlular arasında suç soruşturma departmanına yardımcı olacaktır, robotik evcil hayvanlar ve insansı ortaklarla sohbet, insanlar gibi duyguları anlayabilir ve ifade edebilirlerse daha gerçekçi ve eğlenceli olurdu.

Bu çalışmada, otomatik İngilizce duygusal konuşma tanıma görevine Derin öğrenme uygulaması ele alınmıştır. Parametreler doğru seçildiğinde, elde edilen sonuçlar karşılaştırılabilir ve aslında seçilen diğer sınıflandırıcıların sonuçlarından daha da iyidir.

Gelecekteki çalışmalar, bu deneyleri veritabanındaki diğer dillere genişletmeyi ve şu anda geliştirilmekte olan Türkçe ve Arapça'yı kullanmayı içermektedir.

KAYNAKLAR

[1] B. W. Schuller, "Speech emotion recognition: Two decades in a nutshell benchmarks and ongoing trends", Commun. ACM, vol. 61, pp. 90-99, 2018.

[2] M. S. Hossain and G. Muhammad, "Emotion recognition using deep learning approach from audio–visual emotional big data", Inf. Fusion, vol. 49, pp. 69-78, Sep. 2019.

[3] M. Chen, P. Zhou and G. Fortino, "Emotion communication system", IEEE Access, vol. 5, pp. 326-337, 2016.

[4] N. D. Lane and P. Georgiev, "Can deep learning revolutionize mobile sensing?", Proc. ACM 16th Int. Workshop Mobile Comput. Syst. Appl., pp. 117-122, 2015.

[5] J. G. Rázuri, D. Sundgren, R. Rahmani, A. Moran, I. Bonet and A. Larsson, "Speech emotion recognition in emotional feedbackfor human-robot interaction", Int. J. Adv. Res. Artif. Intell., vol. 4, no. 2, pp. 20-27, 2015.

[6] D. Le and E. M. Provost, "Emotion recognition from spontaneous speech using hidden MARKOV models with deep belief networks", Proc. IEEE Workshop Autom. Speech Recognit. Understand., pp. 216-221, Dec. 2013.

[7] A. B. Nassif, I. Shahin, I. Attili, M. Azzeh and K. Shaalan, "Speech recognition using deep neural networks: A systematic review", IEEE Access, vol. 7, pp. 19143-19165, 2019.

[8] S. Lalitha, A. Madhavan, B. Bhushan and S. Saketh, "Speech emotion recognition", Proc. Int. Conf. Adv. Electron. Comput. Commun. (ICAECC), pp. 1-4, Oct. 2014.

[9] K. R. Scherer, "What are emotions? And how can they be measured?", Social Sci. Inf., vol. 44, no. 4, pp. 695-729, 2005.

[10] T. Balomenos, A. Raouzaiou, S. Ioannou, A. Drosopoulos, K. Karpouzis and S. Kollias, "Emotion analysis in man-machine interaction systems", Proc. Int. Workshop Mach. Learn. Multimodal Interact., pp. 318-328, 2004.

[11] R. Cowie, E. Douglas-Cowie, N. Tsapatsoulis, G. Votsis, S. Kollias, W. Fellenz, et al., "Emotion recognition in human-computer interaction", IEEE Signal Process. Mag., vol. 18, no. 1, pp. 32-80, Jan. 2001.

[12] O. Kwon, K. Chan, J. Hao and T. Lee, "Emotion recognition by speech signal", Proc. EUROSPEECH, pp. 125-128, 2003.

[13] R. W. Picard, "Affective computing", 1995.

[14] S. G. Koolagudi and K. S. Rao, "Emotion recognition from speech: A review", Int. J. speech Technol., vol. 15, no. 2, pp. 99-117, 2012.

[15] M. El Ayadi, M. S. Kamel and F. Karray, "Survey on speech emotion recognition: Features classification schemes and databases", Pattern Recognit., vol. 44, no. 3, pp. 572-587, 2011.

[16] A. D. Dileep and C. C. Sekhar, "GMM-based intermediate matching kernel for classification of varying length patterns of long duration speech using support vector machines", IEEE Trans. neural Netw. Learn. Syst., vol. 25, no. 8, pp. 1421-1432, Aug. 2014.

[17] L. Deng and D. Yu, "Deep learning: Methods and applications", Found. Trends Signal Process., vol. 7, no. 3, pp. 197-387, Jun. 2014.

[18] J. Schmidhuber, "Deep learning in neural networks: An overview", Neural Netw., vol. 61, pp. 85-117, Jan. 2015.

[19] Chanel, G., Kronegg, J., Grandjean, D., Pun, T.: Emotion Assessment: Arousal Evaluation Using EEG's and Peripheral Physiological Signals. In: Gunsel, B., Jain, A.K., Tekalp,

A.M., Sankur, B. (eds.) MRCS 2006. LNCS, vol. 4105, pp. 530–537. Springer, Heidelberg (2006)

[20] Picard, R.W., Klein, J.: Toward computers that recognize and respond to user emotion: Theoretical and practical implications. Interacting with Computers 14(2), 141–169 (2002)

[21] Takahashi, K.: Remarks on emotion recognition from bio-potential signals. In: 2nd International Conference on Autonomous Robots and Agents, pp. 186–191 (2004)

[22] Choppin, A.: Eeg-based human interface for disabled individuals: Emotion expression with neural networks. Masters thesis, Tokyo Institute of Technology, Yokohama, Japan (2000)

[23] Partala, T., Jokiniemi, M., Surakka, V.: Pupillary responses to emotionally provocative stimuli. In: ETRA 2000: Proceedings of the 2000 Symposium on Eye Tracking Research & Applications, pp. 123–129. ACM Press, New York (2000)

[24] W. Minker, J. Pittermann, A. Pittermann, P.-M. Strauß, and D. Buhler, ¨ "Challenges in speech-based human–computer interfaces," International Journal of Speech Technology, vol. 10, no. 2-3, pp. 109–119, 2007.

[25] S. E. Bou-Ghazale and J. H. Hansen, "A comparative study of traditional and newly proposed features for recognition of speech under stress," IEEE Transactions on speech and audio processing, vol. 8, no. 4, pp. 429–442, 2000.

[26] C. M. Lee, S. Yildirim, M. Bulut, A. Kazemzadeh, C. Busso, Z. Deng, S. Lee, and S. Narayanan, "Emotion recognition based on phoneme classes," in Eighth International Conference on Spoken Language Processing, 2004.

[27] B. Schuller, G. Rigoll, and M. Lang, "Hidden markov model-based speech emotion recognition," in Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03). 2003 IEEE International Conference on, vol. 2. IEEE, 2003, pp. II–1.

[28] J. Lee and I. Tashev, "High-level feature representation using recurrent neural network for speech emotion recognition," 2015.

[29] M. Wieman and A. Sun, "Analyzing vocal patterns to determine emotion," 2014.

[30] R. Chen, Y. Zhou, and Y. Qian, "Emotion recognition using support vector machine and deep neural network," in National Conference on Man-Machine Speech Communication. Springer, 2017, pp. 122–131.

[31] D. Neiberg, K. Elenius, and K. Laskowski, “Emotion recognition in spontaneous speech using gmms,” in Ninth International Conference on Spoken Language Processing, 2006.

[32] A. S. Utane and S. Nalbalwar, “Emotion recognition through speech using gaussian mixture model and support vector machine,” emotion, vol. 2, no. 8, 2013.

[33] N. Landwehr, M. Hall, and E. Frank, “Logistic model trees,” Machine learning, vol. 59, no. 1-2, pp. 161–205, 2005.

[34] C. Busso, Z. Deng, S. Yildirim, M. Bulut, C. M. Lee, A. Kazemzadeh, S. Lee, U. Neumann, and S. Narayanan, “Analysis of emotion recognition using facial expressions, speech and multimodal information,” in Proceedings of the 6th international conference on Multimodal interfaces. ACM, 2004, pp. 205–211.

[35] Audio Deep Learning Made Simple (Part 1): State-of-the-Art Techniques https://towardsdatascience.com/audio-deep-learning-made-simple-part-1-state-of-the-art-techniques-da1d3dff2504

[36]https://commons.wikimedia.org/wiki/File:FFT-Time-Frequency-View.png#/media/File:FFT-Time-Frequency-View.png

[37]https://commons.wikimedia.org/wiki/File:Signal_Sampling.png#/media/File:Signal_Sampling.png

[38] http://www.cs.toronto.edu/~gpenn/csc401/soundASR.pdf

[39] First published Thu Jul 14, 2005; substantive revision Fri Apr 10, 2020

[40] New method enables high quality speech separation Authors to present research at SIGGRAPH 2018

[41] ENVIRONMENTAL SOUND CLASSIFICATION WITH CONVOLUTIONAL NEURAL NETWORKS Karol J. Piczak Institute of Electronic Systems Warsaw University of Technology

[42] https://github.com/marcogdepinto/emotion-classification-from-audio-files

[43] A brief introduction to audio data processing and genre classification using Neural Networks and python. By Nagesh Singh Chauhan, Data Science Enthusiast.

[44] https://deepai.org/machine-learning-glossary-and-terms/feature-extraction

[45] A Gentle Introduction to k-fold Cross-Validation by Jason Brownlee on May 23, 2018 in Statistics

[46] Available: https://www.datacamp.com/community/tutorials/neural-networkmodels-r

[47] [Online] Available: https://lilly021.com/convolutional-neural-networks/

[48] [Online] Available: http://cs231n.github.io/convolutional-networks/

[49] Bengio, Y.; Courville, A.; Vincent, P. (2013). "Representation Learning: A Review and New Perspectives". IEEE Transactions on Pattern Analysis and Machine Intelligence. 35 (8): 1798–1828. arXiv:1206.5538. doi:10.1109/tpami.2013.50. PMID 23787338. S2CID 393948.

[50] Schmidhuber, J. (2015). "Deep Learning in Neural Networks: An Overview". Neural Networks. 61: 85–117. arXiv:1404.7828. doi:10.1016/j.neunet.2014.09.003. PMID 25462637. S2CID 11715509.

[51] Bengio, Yoshua; LeCun, Yann; Hinton, Geoffrey (2015). "Deep Learning". Nature. 521 (7553): 436–444. Bibcode:2015Natur.521..436L. doi:10.1038/nature14539. PMID 26017442. S2CID 3074096.

[52] Hu, J.; Niu, H.; Carrasco, J.; Lennox, B.; Arvin, F. (2020). "Voronoi-Based Multi-Robot Autonomous Exploration in Unknown Environments via Deep Reinforcement Learning". IEEE Transactions on Vehicular Technology. 69 (12): 14413–14423. doi:10.1109/TVT.2020.3034800. S2CID 228989788. Archived from the original on 2020-11-16. Retrieved 2021-05-04.

[53] Krizhevsky, Alex; Sutskever, Ilya; Hinton, Geoffry (2012). "ImageNet Classification with Deep Convolutional Neural Networks" (PDF). NIPS 2012: Neural Information Processing Systems, Lake Tahoe, Nevada. Archived (PDF) from the original on 2017-01-10. Retrieved 2017-05-24.

[54] "Google's AlphaGo AI wins three-match series against the world's best Go player". TechCrunch. 25 May 2017. Archived from the original on 17 June 2018. Retrieved 17 June 2018.

[55] Softmax Activation Function with Python by Jason Brownlee on October 19, 2020 in Deep Learning.

[56](https://medium.datadriveninvestor.com/notes-on-deep-learning-softmax-classifier-971b3df27466)

[57]https://plato.stanford.edu/entries/emotion/

[58](Doror, 2017; Dutton ve Aaron, 1974; Schachter & Singer, 1962)

[59] B. Schuller, G. Rigoll, and M. Lang, “Hidden markov model-based speech emotion recognition,” in Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP’03). 2003 IEEE International Conference on, vol. 2. IEEE, 2003, pp. II-1.

[60] J. Lee and I. Tashev, “High-level feature representation using recurrent neural network for speech emotion recognition,” 2015.

[61] R. Chen, Y. Zhou, and Y. Qian, “Emotion recognition using support vector machine and deep neural network,” in National Conference on Man-Machine Speech Communication. Springer, 2017, pp. 122–131.

[62] C. Busso, Z. Deng, S. Yildirim, M. Bulut, C. M. Lee, A. Kazemzadeh, S. Lee, U. Neumann, and S. Narayanan, “Analysis of emotion recognition using facial expressions, speech and multimodal information,” in Proceedings of the 6th international conference on Multimodal interfaces. ACM, 2004, pp. 205–211.

[63] C.-C. Lee, E. Mower, C. Busso, S. Lee, and S. Narayanan, “Emotion recognition using a hierarchical binary decision tree approach,” Speech Communication, vol. 53, no. 9-10, pp. 1162–1171, 2011.

[64] K. Wang, N. An, B. N. Li, Y. Zhang, and L. Li, “Speech emotion recognition using fourier parameters,” IEEE Transactions on Affective Computing, vol. 6, no. 1, pp. 69–75, 2015.

[65] T.-S. Lim, W.-Y. Loh, and Y.-S. Shih, “A comparison of prediction accuracy, complexity, and training time of thirty-three old and new classification algorithms,” Machine learning, vol. 40, no. 3, pp. 203–228, 2000.

[66] English speech sound improvement system based on deep learning from signal processing to semantic recognition Yucheng Yang1 · Yibo Yue1 Received: 1 March 2020 / Accepted: 6 July 2020

EKLER

Ek 1.1: pip librosa kaydedilmesi

```
pip install librosa
or
conda install -c conda-forge librosa
```

Ek 1.2: IPython.display.

```
import librosa
audio_data = '/../../gruesome.wav'
x , sr = librosa.load(audio_data)
print(type(x), type(sr))#<class 'numpy.ndarray'> <class
'int'>print(x.shape, sr)#(94316,) 22050
```

Ek 1.3: Örnekleme hızı, Hz veya kHz cinsinden ölçülen

```
librosa.load(audio_data, sr=44100)
librosa.load(audio_path, sr=None)
```

Ek 1.4: Ses Çalma

```
import IPython.display as ipd
ipd.Audio(audio_data)
```

Ek 1.5: Sesi Görselleştirme

```
%matplotlib inline
import matplotlib.pyplot as plt
import librosa.display
plt.figure(figsize=(14, 5))
librosa.display.waveplot(x, sr=sr)
```

Ek 1.6: Spektrogram gösteremek kod

```
X = librosa.stft(x)
Xdb = librosa.amplitude_to_db(abs(X))
plt.figure(figsize=(14, 5))
librosa.display.specshow(Xdb, sr=sr, x_axis='time', y_axis='hz')
plt.colorbar()
```

Ek 1.7: Bir Ses Sinyali Oluşturun

```
librosa.display.specshow(Xdb, sr=sr, x_axis='time', y_axis='log')
plt.colorbar()
import numpy as np
sr = 22050 # sample rate
T = 5.0    # seconds
t = np.linspace(0, T, int(T*sr), endpoint=False) # time variable
x = 0.5*np.sin(2*np.pi*220*t)# pure sine wave at 220 Hz
#Playing the audio
ipd.Audio(x, rate=sr) # load a NumPy array
#Saving the audio
librosa.output.write_wav('tone_220.wav', x, sr)
```

Ek 1.8: librosa. feature. spectral_centroid

```
1.8
import sklearn
spectral_centroids = librosa.feature.spectral_centroid(x, sr=sr)[0]
spectral_centroids.shape
(775,)
# Computing the time variable for visualization
plt.figure(figsize=(12, 4))frames = range(len(spectral_centroids))
t = librosa.frames_to_time(frames)
# Normalising the spectral centroid for visualisation
def normalize(x, axis=0):
    return sklearn.preprocessing.minmax_scale(x, axis=axis)
#Plotting the Spectral Centroid along the waveform
librosa.display.waveplot(x, sr=sr, alpha=0.4)
plt.plot(t, normalize(spectral_centroids), color='b')
```

Ek 1.9: Spektral Devrilme (Rolloff)

```
spectral_rolloff = librosa.feature.spectral_rolloff(x+0.01, sr=sr)[0]
plt.figure(figsize=(12, 4))librosa.display.waveplot(x, sr=sr, alpha=0.4)
plt.plot(t, normalize(spectral_rolloff), color='r')
```

Ek 1.10: Spektral Bant Genişliği (Bandwidth)

```
spectral_bandwidth_2 = librosa.feature.spectral_bandwidth(x+0.01, sr=sr)[0]
spectral_bandwidth_3 = librosa.feature.spectral_bandwidth(x+0.01, sr=sr,
p=3)[0]
spectral_bandwidth_4 = librosa.feature.spectral_bandwidth(x+0.01, sr=sr,
p=4)[0]
plt.figure(figsize=(15, 9))librosa.display.waveplot(x, sr=sr, alpha=0.4)
plt.plot(t, normalize(spectral_bandwidth_2), color='r')
plt.plot(t, normalize(spectral_bandwidth_3), color='g')
plt.plot(t, normalize(spectral_bandwidth_4), color='y')
plt.legend(('p = 2', 'p = 3', 'p = 4'))
```

Ek 1.11: Sıfır Geçiş Oranı

```
x, sr = librosa.load('/../../gruesome.wav')
#Plot the signal:
plt.figure(figsize=(14, 5))
librosa.display.waveplot(x, sr=sr)
# Zooming in
n0 = 9000
n1 = 9100
plt.figure(figsize=(14, 5))
plt.plot(x[n0:n1])
plt.grid()
n0 = 9000
n1 = 9100
plt.figure(figsize=(14, 5))
plt.plot(x[n0:n1])
plt.grid()
zero_crossings = librosa.zero_crossings(x[n0:n1], pad=False)
print(sum(zero_crossings))#16
```

Ek 1.12: MFCC

```
mfccs = librosa.feature.mfcc(x, sr=fs)
print(mfccs.shape)
(20, 97)
#Displaying  the MFCCs:
plt.figure(figsize=(15, 7))
librosa.display.specshow(mfccs, sr=sr, x_axis='time')
chromagram = librosa.feature.chroma_stft(x, sr=sr, hop_length=hop_length)
plt.figure(figsize=(15, 5))
librosa.display.specshow(chromagram, x_axis='time', y_axis='chroma',
hop_length=hop_length, cmap='coolwarm')
```

Ek 2.1: paython'da pip kullanarak bazı bağımlılıkların kurulması

```
pip3 install librosa==0.6.3 numpy soundfile==0.9.0 sklearn
pyaudio==0.2.11
import soundfile # to read audio file
import numpy as np
import librosa # to extract speech features
import glob
import os
import pickle # to save model after training
from sklearn.model_selection import train_test_split # for splitting
training and testing
```

```
from sklearn.neural_network import MLPClassifier # multi-layer perceptron
model
from sklearn.metrics import accuracy_score # to measure how good we are
```

Ek 2.2: çıkarma özelliklerini işleyen işlev oluşturmak

```
def extract_feature(file_name, **kwargs):
    """
    Extract feature from audio file `file_name`
        Features supported:
            - MFCC (mfcc)
            - Chroma (chroma)
            - MEL Spectrogram Frequency (mel)
            - Contrast (contrast)
            - Tonnetz (tonnetz)
        e.g:
        `features = extract_feature(path, mel=True, mfcc=True)`
    """
    mfcc = kwargs.get("mfcc")
    chroma = kwargs.get("chroma")
    mel = kwargs.get("mel")
    contrast = kwargs.get("contrast")
    tonnetz = kwargs.get("tonnetz")
    with soundfile.SoundFile(file_name) as sound_file:
        X = sound_file.read(dtype="float32")
        sample_rate = sound_file.samplerate
        if chroma or contrast:
            stft = np.abs(librosa.stft(X))
        result = np.array([])
        if mfcc:
            mfccs = np.mean(librosa.feature.mfcc(y=X, sr=sample_rate,
n_mfcc=40).T, axis=0)
            result = np.hstack((result, mfccs))
        if chroma:
```

```
            chroma = np.mean(librosa.feature.chroma_stft(S=stft,
sr=sample_rate).T,axis=0)
            result = np.hstack((result, chroma))
        if mel:
            mel = np.mean(librosa.feature.melspectrogram(X,
sr=sample_rate).T,axis=0)
            result = np.hstack((result, mel))
        if contrast:
            contrast =
np.mean(librosa.feature.spectral_contrast(S=stft,
sr=sample_rate).T,axis=0)
            result = np.hstack((result, contrast))
        if tonnetz:
            tonnetz =
np.mean(librosa.feature.tonnetz(y=librosa.effects.harmonic(X),
sr=sample_rate).T,axis=0)
            result = np.hstack((result, tonnetz))
    return result
```

Ek 2.3: Veri kümesi hakkında bazı bilgilerin günlüğe kaydedilmesi

```
# all emotions on RAVDESS dataset

int2emotion = {
    "01": "neutral",
    "02": "calm",
    "03": "happy",
    "04": "sad",
    "05": "angry",
    "06": "fearful",
    "07": "disgust",
    "08": "surprised"
}

AVAILABLE_EMOTIONS = {
"neutral",
```

```
"calm",
"happy",
"sad",
"angry",
"fearful",
"disgust",
"surprised"
}
def load_data(test_size=0.2):
    X, y = [], []
    for file in glob.glob("data/Actor_*/*.wav"):
        # get the base name of the audio file
        basename = os.path.basename(file)
        # get the emotion label
        emotion = int2emotion[basename.split("-")[2]]
        # we allow only AVAILABLE_EMOTIONS we set
        if emotion not in AVAILABLE_EMOTIONS:
            continue
        # extract speech features
        features = extract_feature(file, mfcc=True, chroma=True,
mel=True)
        # add to data
        X.append(features)
        y.append(emotion)
    # split the data to training and testing and return it
    return train_test_split(np.array(X), y, test_size=test_size,
random_state=7)



# load RAVDESS dataset, 70% training 30% testing
X_train, X_test, y_train, y_test = load_data(test_size=0.30)


# print some details
# number of samples in training data
print("[+] Number of training samples:", X_train.shape[0])
```

```
# number of samples in testing data
print("[+] Number of testing samples:", X_test.shape[0])
# number of features used
# this is a vector of features extracted
# using extract_features() function
print("[+] Number of features:", X_train.shape[1])
```

Ek 2.4: Mümkün olan en iyi hiperparametreleri elde etmek için MLPClassifier üzerinde bir grid araması yaptıktan sonra, şimdiye kadar mevcut olan parametreler belirlenlenip işlev üzerinde kullanılmıştır: train model VE Accuracy Kodu

```
# best model, determined by a grid search
model_params = {
    'alpha': 0.01,
    'batch_size': 256,
    'epsilon': 1e-08,
    'hidden_layer_sizes': (300,),
    'learning_rate': 'adaptive',
    'max_iter': 500,
}
# initialize Multi Layer Perceptron classifier
# with best parameters ( so far )
model = MLPClassifier(**model_params)
# train the model
print("[*] Training the model...")
model.fit(X_train, y_train)

# predict 25% of data to measure how good we are
y_pred = model.predict(X_test)

# calculate the accuracy
accuracy = accuracy_score(y_true=y_test, y_pred=y_pred)

print("Accuracy: {:.2f}%".format(accuracy*100))
```

Ek 2.5: Sistemin sonuç vermesi birkaç saniye sürmektedir. Ardından doğruluk puanı hesaplanır ve ne kadar iyi olduğunu ölçmek için yazılır: Modelin kayıt kodu

```
# now we save the model
# make result directory if doesn't exist yet
if not os.path.isdir("result"):
    os.mkdir("result")

pickle.dump(model, open("result/mlp_classifier.model", "wb"))
if __name__ == "__main__":
    # load the saved model (after training)
    model = pickle.load(open("result/mlp_classifier.model", "rb"))
    print("Please talk")
    filename = "test.wav"
    # record the file (start talking)
    record_to_file(filename)
    # extract features and reshape it
    features = extract_feature(filename, mfcc=True, chroma=True,
mel=True).reshape(1, -1)
    # predict
    result = model.predict(features)[0]
    # show the result !
    print("result:", result)
```