T.C.
YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

MÜZİK TÜRÜ SINIFLANDIRMA

Yunus ATAHAN

YÜKSEK LİSANS TEZİ
Bilgisayar Mühendisliği Anabilim Dalı
Bilgisayar Mühendisliği Programı

Danışman
Prof. Dr. Nizamettin AYDIN

Eş-Danışman
Öğr. Gör. Dr. Ahmet ELBİR

Şubat, 2022

T.C.
YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

MÜZİK TÜRÜ SINIFLANDIRMA

Yunus ATAHAN tarafından hazırlanan tez çalışması 02.02.2022 tarihinde aşağıdaki jüri tarafından Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı Bilgisayar Mühendisliği Programı **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Prof. Dr. Nizamettin AYDIN
Yıldız Teknik Üniversitesi
Danışman

Öğr. Gör. Dr. Ahmet ELBİR
Yıldız Teknik Üniversitesi
Eş-Danışman

Jüri Üyeleri

Prof. Dr. Nizamettin AYDIN, Danışman
Yıldız Teknik Üniversitesi _____

Doç. Dr. Mehmet Sıddık AKTAŞ, Üye
Yıldız Teknik Üniversitesi _____

Prof. Dr. Murat SARAÇLAR, Üye
Boğaziçi Üniversitesi _____

Danışmanım Prof. Dr. Nizamettin AYDIN sorumluluğunda tarafımca hazırlanan Müzik Türü Sınıflandırma başlıklı çalışmada veri toplama ve veri kullanımında gerekli yasal izinleri aldığımı, diğer kaynaklardan aldığım bilgileri ana metin ve referanslarda eksiksiz gösterdiğimi, araştırma verilerine ve sonuçlarına ilişkin çarpıtma ve/veya sahtecilik yapmadığımı, çalışmam süresince bilimsel araştırma ve etik ilkelerine uygun davrandığımı beyan ederim. Beyanımın aksinin ispatı halinde her türlü yasal sonucu kabul ederim.

Yunus ATAHAN

İmza

Aileme ithafen

TEŞEKKÜR

Yüksek lisans eğitimim boyunca öğrettikleri değerli bilgiler ve gösterdikleri yardımseverlik ve anlayıştan dolayı danışmanım Prof. Dr. Nizamettin AYDIN hocama teşekkür ederim. Ne zaman bir sorum olsa cevaplayan ve tezin hem uygulama hem de yazım aşamasında çok büyük emek ve destekleri olan eşdanışmanım Öğr. Gör. Dr. Ahmet ELBİR hocama teşekkür ederim.

Bütün bu süreç boyunca beni destekleyen anneme, babama ve kardeşlerime gösterdikleri sabır ve anlayış için teşekkür ederim.

Yunus ATAHAN

İÇİNDEKİLER

SİMGE LİSTESİ

fc	Spektral Merkez
$S(k) \rightarrow$ k	Frekans Kutusunun Genliği
$f(k) \rightarrow$ k	Frekans Kutusu
$\sum$	Toplam Sembolü
$\prod$	Çarpım Sembolü
$\sqrt{n}$	Karekök(n)

KISALTMA LİSTESİ

AE	Autoencoder
CNN	Convolutional Neural Network
KNN	K Nearest Neighbour
LDA	Linear Discriminant Analysis
LPC	Linear Predictive Coding
MFCC	Mel Frequency Cepstral Coefficients
MIDI	Musical Instrument Digital Interface
MIR	Music Information Retrieval
MLP	Multi Layer Perceptron
MSE	Mean Squared Error
NB	Naive Bayes
PCA	Principal Component Analysis
STFT	Short Time Fourier Transform
SVM	Support Vector Machine
ZCR	Zero Crossing Rates

ŞEKİL LİSTESİ

TABLO LİSTESİ

ÖZET

Müzik Türü Sınıflandırma

Yunus ATAHAN

Bilgisayar Mühendisliği Anabilim Dalı
Yüksek Lisans Tezi

Danışman: Prof. Dr. Nizamettin AYDIN
Eş-Danışman: Öğr. Gör. Dr. Ahmet ELBİR

Müzik internetin gelişmesi ile, erişimi ve paylaşımının kolaylığı nedeniyle eskisinden çok daha fazla insana ulaşmaya başlamıştır. Bunun sonucunda müzik platformları ortaya çıkmış ve insanlara dünyanın dört bir yanından müzikleri ulaştırmayı sağlamışlardır. Müzik platformlarının amaçlarından birisi de kullanıcıları platformlarında olabildiğince uzun süre tutabilmektir. Bunun için müzik öneri sistemleri kullanılarak kullanıcıların dinlediğinde sevebileceği müzikler kullanıcıya ulaştırılır. Müzik öneri sistemlerinde uygulanan bir yol, müzik türüne göre öneri yapılmasıdır. Kullanıcının dinlediği müziklerin türleri incelenir ve aynı tür yada uygun olabilecek türler kullanıcıya sunulur. Bu aşamada otomatik müzik türü sınıflandırma algoritmalarının kullanımı son yıllarda hızlı bir artmıştır. Müzik türlerinin insanlar tarafından çok iyi bir başarı ile sınıflandırılamaması, çok uzun zaman alması, ve kişiden kişiye müzik kavramının değişebilmesi gibi sebepler otomatik tür sınıflandırma algoritmalarının geliştirilmesine yol açmıştır. Bu çalışmada müziğin akustik özellikleri kullanıldı.

Müziğin hem zaman hem de frekans uzayında ayırt edici nitelikleri vardır. Bu nitelikler farklı müzik türlerinde farklı ve aynı müzik türlerinde benzer özellik gösterdiğinden dolayı müzik türü sınıflandırma amacıyla kullanılmıştır.

Bu tezde müziğin ses özelliklerinin kullanımı ile elde edilen sınıflandırma sonuçları, otomatik kodlayıcılar ile özellik çıkarımı yapılarak elde edilen sınıflandırma sonuçları ve bir boyutlu yapay derin öğrenme ağı ile yapılan sınıflandırma sonuçları karşılaştırılmıştır. Bu karşılaştırmaların yapılabilmesi için daha önce bu alanda yapılan

çalışmalarda sıkça kullanılan bir veri seti seçilmiştir. Veri seti özellik çıkarımına uygun ve yapay sinir ağlarına girdi olabilecek şekilde düzenlenmiştir. Müziklerden ses özellik çıkarımı zaman ve frekans bölgesi özellikleri ile yapılmıştır. Otomatik kodlayıcılar için, müziğin kısa zamanda Fourier dönüşümü alındıktan sonra elde edilen veriler resim verisi şeklinde temsil edilmiş ve otomatik kodlayıcı evrişimli sinir ağı olarak oluşturulmuştur. Tek boyutlu derin öğrenme ağı için ise müzik belirli uzunluklara bölünmüş ve ham ses verisi ağa girdi olarak verilmiştir.

Yapılan çalışmalar sonucunda akustik özellik çıkarımı kullanılarak yapılan sınıflandırmanın sonuçlarının otomatik kodlayıcıdan gelen sonuçlardan çok daha iyi olduğu gözlenmiştir. Tek boyutlu derin öğrenme ağı ise her iki yöntemden daha iyi sonuç vermiştir. Yapılan bu çalışma her ne kadar otomatik kodlayıcılardan elde edilen sonuçlar istenilen kadar iyi olmasa da, tek boyutlu ağdan elde edilen sonuçlar özellik çıkarımının yapay zeka ile de yapılabileceğini göstermiştir. Gelecekteki çalışmalarda farklı yapay sinir ağları ve derin öğrenme yöntemleri denenerek özellik çıkarımının nasıl sonuçlar vereceği gözlemlenebilir.

Anahtar Kelimeler: Müzik türü sınıflandırma, GTZAN veri seti, sayısal işaret işleme, derin öğrenme, yapay sinir ağları, otomatik kodlayıcı, Müzik öneri sistemleri

ABSTRACT

Music Genre Classification

Yunus ATAHAN

Department of Computer Engineering
Master of Science Thesis

Supervisor: Prof. Dr. Nizamettin AYDIN
Co-supervisor: Lec. Dr. Ahmet ELBİR

With the evolution of the Internet, because of its ease of access and share, music has reached more people than before. As a result of this, music platforms were created so more people from all around the World were able to access to different musics. One of the goals of the music platforms is to keep users in their platform as long as possible. For this purpose music recommendation systems are used to provide musics that users would love. One way for music recommendation system is using the music genres. Musics that user listens are examined and recommendation systems try to recommend the same or similar genre of musics to user. As a result of this, usage of automatic music genre classification algorithms increased in recent years. Low accuracy and long time requirement of manual music genre classification and interpersonal variability of music information and knowledge are main reasons for developing automatic music genre classification algorithms. In this study we used the approach which is based on acoustical features.

Music contains some features in time and frequency domains. Features are used for music genre classification since those features have different characteristics with different genre and same characteristics with the same genres.

In this thesis classification results obtained by the features that are gathered using acoustic features, classification results of features that are obtained by using auto-encoders and classification result of one dimensional neural network are compared. To make this comparison, a data-set used frequently in this field of research is used. The data-set is arranged in a way that acoustic features can be extracted and

it can be input to the neural network. Acoustic feature extraction has been done using time and frequency domain features. For auto-encoders, short time Fourier transform of the music signal is represented as images, and auto-encoder was built as a convolutional neural network. For one dimensional deep learning network musics were arranged to have the same length and raw audio data is used as an input.

As a result of the studies, classification accuracy obtained by using acoustic features is much better than the one obtained by the auto-encoder. One dimensional deep learning network on the other hand performed better than both acoustic features and auto-encoder in terms of accuracy. This study shows that although the results from auto-encoders are not as good as expected, results observed from one dimensional network show that feature extraction from music can be done using artificial intelligence methods rather than digital signal processing methods and acoustic features. In future studies different neural networks and deep learning methods can be applied and results can be examined.

Keywords: Music genre classification, GTZAN data-set, digital signal processing, deep learning, artificial neural networks, auto-encoder, music recommendation systems

1
GİRİŞ

1.1 Literatür Özeti

Müzik geçmişte olduğu gibi günümüzde de insanların hayatının bir parçası olmaya devam etmektedir. Eskiden belirli yörelere ait olan müziklerin başka yöredeki insanlara ulaşması kolay değildi. İnternetin gelişmesi ve hızlanması insanların ürettikleri eserleri kolayca dünyanın öbür ucundaki insanlara ulaştırmasını sağlamaya başladı. Bu hızlı gelişen müzik dağılımının bir sonucu olarak, dünyada büyük boyutlu müzik veri tabanları oluşmaya başladı. Başlarda kasetlerde saklanan bu veriler daha sonra CD'ler, DVD'ler, harici bellekler ve günümüzde de büyük veri merkezlerinde sabit disklerde saklanmaya başlanmıştır. Bu oluşan verinin büyüklüğü beraberinde müzik yayın platformlarının kurulmasını sağlamıştır. Müzik platformlarının amacı bu büyük boyutlu müzik veritabanında bulunan müzikleri düzenleyip, bunları kullanıcıya ulaştırarak hizmet vermektir. Bu platformlar bunu yaparken bazı zorluklarla karşı karşıya kalırlar. Müzik verileri içerisinde yalnızca ham verinin olması, şarkıcının, söz yazarının veya müziğin türü gibi üst verinin (meta-data) her zaman bulunmaması bu zorluklardan bazılarıdır. Müzik platformları kullanıcıya hizmet sunarken aynı zamanda kullanıcının platformda uzun süre geçirmesini hedeflemektedir. Bunun başlıca yöntemlerinden bir tanesi kullanıcının sistemde daha uzun zaman geçirmesini sağlamaktır. Bu öneri sisteminde kullanılan yöntemlerden bir tanesi de müzik türüne göre öneri yapmaktır. Ancak bunun için veri setinde bulunan müziklerin türlerinin bilinmesi gerekir. Çoğu uygulamada ve özellikle çok eski zamanlarda üretilen müzik kayıtlarında müzik türü bilgisinin bulunmaması önemli bir problem oluşturmaktadır. Bu problemleri çözmek için, ilk olarak insanlar kendileri birer birer türü bulunmayan müzikleri dinlemiş ve tür çıkarımı yapmışlardır. Kullanılan bu yöntem, süreç bakımından uzun sürmesi, doğruluk oranının istenildiği kadar olmaması ve kişilerin müzik bilgisi birikiminin aynı seviyede olmaması sebebiyle çok etkili olamamıştır.

Bahsedilen bu sorunlara çözüm olarak Müzik Bilgisi Edinme (Music Information retrieval - MIR) adlı bir bilim dalı ortaya çıkmıştır [1]. Bu bilim dalının amacı müziği

inceleyerek, müziğin ayırt edici öz niteliklerini çıkarmak ve bu nitelikler ile müzik hakkında çalışmalar yapmaktır. Bu çalışma alanlarından biri ise türü bilinmeyen bir müziğin türünün tahmin edilmesidir. Bunun için ilk yapılan çalışmalarda müzikten elde edilen belirli akustik özellikler [2] kullanılmış ve istatiksel örüntü tanıma algoritmaları eğitilerek özelliklerin performansı ve performansa verdiği katkı incelenmiştir. Gerçek zamanlı olmayan testlerde %61 sınıflandırma doğruluk oranına, gerçek zamanlı testlerde %44 sınıflandırma doğruluk oranına ulaşmışlardır. Yapılan bu çalışmada başarı oranı düşük olsa da literatüre kazandırılan veri seti daha sonraki çalışmlara deney veri seti olarak katkıda bulunmuştur. Daha sonraki çalışmalarda farklı özellikler ve farklı algoritmalar ile bu işlem tekrarlanmıştır. Örneğin [3] çalışmalarında MIDI (Musical Instrument Digital Interface) ses dosyalarından özellik çıkarımı yaparak bu özellikleri farklı makine öğrenmesi algoritmalarına vererek bu algoritmaları eğitmiştir. Bu çalışma sonucunda elde edilen veriler, basit müziksel özelliklerin müzik türü için önemli veriler içerdiğini ortaya koymuştur.

2003 yılında yapılan başka bir çalışmada ritim spektrumu, doğrusal öngörücü kodlama (Linear Predictive Coding (LPC)), sıfır geçiş oranı (Zero Crossing Rates (ZCR)), spektrum gücü (spectral power) ve mel frekans kepstral katsayıları (Mel Frequeny Cepstral Coefficients (MFCC)) özellikleri müzikten elde edilmiştir. Daha sonra bu özellikler destek vektör makinesi (Support Vector Machine (SVM)) algoritmasına girdi olarak verilip SVM eğitilmiş ve bu algoritma ile sınıflandırma yapılmıştır [4]. Bu çalışmada en iyi sonucu %6.86 hata oranı ile SVM vermiştir.

Haggblade ve arkadaşları yaptıkları çalışmada farklı makine öğrenmesi algoritmalarını karşılaştırmış ve yapay sinir ağları ve SVM algoritmasının en iyi sonuçları verdiğini belirtmişlerdir. Yapay sinir ağlarını kullanarak yaptıkları sınıflandırmanın doğruluk oranı %96 olmuştur. Bu sonucun önceki çalışmaları incelediklerinde bekledikleri bir sonuç olduğunu söylemişlerdir [5].

Yapılan diğer bir çalışmada müzik türü sınıflandırma çalışmalarında en sık kullanılan iki veri seti olan GTZAN ve ISMIR veri setleri beraber kullanılmıştır. Çalışmada çoklu doğrusal yöntemler ile günümüzde bilinen ve kullanılan diğer yöntemleri karşılaştırmışlardır. Bu veri setlerindeki müziklerden, çokdoğrusal altuzay analiz (multilinear subspace analysis) tekniklerini kullanarak, müziklerin kortikal temsillerini özellik olarak çıkarmışlardır [6].

2018 yılında yapılan çalışmada [7], müzikten elde elde edilen spektrogramları kullanarak zaman-frekans analizi ile özellik çıkarımı yapmışlardır. Bu çalışmada spektrogramların elde edilmesi için farklı pencere türleri, pencere boyutları ve örtüşme oranı kullanılmıştır. Elde edilen özellikler destek vektör makinesi algoritmasının

farklı kernelleri ve rastgele orman algoritması ile türlere göre sınıflandırma yapmak için kullanılmıştır. Bu çalışmada en iyi sonucu %71.3 ile pencere tipi olarak Parzen Window ve Polynomial Kernel kullanan SVM algoritması vermiştir.

2018 yılında yaptıkları diğer bir çalışmada sayısal işaret işleme yöntmeleri ile müzikten özellik çıkarımı yapılmış ve bu özellikler makine öğrenmesi yöntemleri kullanılarak sınıflandırılmıştır [8]. Aynı zamanda spektrogram görüntüsü evrişimli sinir ağlarına girdi olarak verilip bu ağlar eğitilmiş ve elde edilen sonuçlar karşılaştırılmıştır. Müzik özellikleri ile yapılan denemede SVM algoritması %72.39 ile çalışmanın en yüksek sınıflandırma doğruluk oranını vermiştir. Bu sonuç pencere tipi Barthann ve pencere boyutu 4096 olduğu zaman elde edilmiştir. CNN ile yapılan sınıflandırmada ise en yüksek doğruluk oranını %66 ile STFT özelliği vermiştir.

2020 yılında yapılan çalışmada ise MusicRecNet adında yeni bir evrişimli sinir ağı modeli sunmuşlardır [9]. Bu model işaret işleme ile elde edilen Mel Spektrogram özelliklerini girdi olarak alan ve sınıflandırma yapan bir modeldir. Bu modelin sondan bir önceki katmanı özellik vektörü olarak kullanılmış ve bu özellikler makine öğrenmesi algoritmalarına eğitim verisi olarak verilmiştir. Hem MusicRecNet modelinden hemde makine öğrenmesi modelinden gelen özellikler kullanılarak performans değerlendirmesi yapılmıştır. MusicRecNet tek başına kullanıldığında %81.8 sınıflandırma doğruluğu sağlarken SVM ile kullanıldığında bu oran %97.6'ya kadar çıkmıştır. Yapılan değerlendirmeler sonucunda elde edilen başarının önceki çalışmalardan çok daha yüksek olduğu görülmüştür.

T. Feng yaptığı çalışmada derin öğrenme modeli ile müzik türü sınıflandırması yapmıştır [10]. Bir diğer çalışmada IY. Jeong ve K. Lee derin öğrenmeyi kullanarak özellik çıkarımı yapmışlardır [11]. 2022'de yayınlanacak geleneksel makine öğrenmesi yöntemleri ile derin öğrenmeyi karşılaştıran bir çalışma bulunmaktadır [12].

1 boyutlu CNN ile çalışan Taejun ve arkadaşları %91 sınıflandırma başarı oranına ulaşmışlardır [13]. 1 boyutlu CNN ile yapılan bir diğer çalışmada ise [14] %3.62'lik bir başarı yükselmesi sağlanmıştır. Bu çalışmada residual CNN kullanmışlar ve normal CNN yapılarının sonuçları ile karşılaştırmışlardır.

1.2 Tezin Amacı

Bu çalışmada müzik türü sınıflandırma için kullanılacak özelliklerin çıkarımı ve bu özelliklerin sınıflandırma başarısına olan etkisinin incelenmesi amaçlanmıştır. Bu kapsamda, bu alanda yapılan önceki çalışmalarda sıklıkla kullanılan veri seti olan

GTZAN veri seti seçilmiştir. Yapılan bu çalışma sonucunda sayısal işaret işleme ile elde edilen müziğin zaman ve frekans uzayındaki özellikleri yerine yapay sinir ağları ile oluşturulmuş bir otomatik kodlayıcı kullanılabilir mi sorusuna cevap bulunması beklenmektedir. Bu çalışma önceki yapılan çalışmaların bir karşılaştırılması ve denetimsiz öğrenme ile denetimli öğrenme arasındaki farkların incelenmesi amacıyla yapılmıştır. Aynı zamanda günümüzde çok hızlı bir şekilde artan müzik verisinin denetimsiz eğitim ile kümeleme yapıldığında nasıl bir sonuç verdiğinin cevabının bulunması da bu çalışmanın amaçları arasındadır. Ek olarak 1 boyutlu bir CNN modeli kullanılarak, ham ses verisinden bir özellik çıkarımı yapılmadan yapılan sınıflandırma sonuçları incelenmiştir.

1.3 Hipotez

Müzik türü sınıflandırma için yapılan önceki çalışmalarda müziğin akustik özellikleri, yapay sinir ağları, makine öğrenmesi yöntemleri kullanılmıştır. Ancak otomatik kodlayıcılar ile yapılan çalışmaların [10] azlığı nedeniyle bu alana da yoğunlaşılmıştır. Otomatik kodlayıcılardan elde edilen özelliklerin ne kadar başarı sağlayacağı incelenebilir. Bu sonuçlar ile önceki çalışmalarda elde edilen sonuçlar karşılaştırılarak özellik çıkarımı için hangi yöntemin kullanılmasının daha iyi sonuç verebileceği gözlemlenebilir. Önceki çalışmalarda uygulanan denetimli öğrenme yöntemlerinin kümeleme gibi denetimsiz öğrenme yöntemleri ile kullanılmasının başarıya nasıl etki edeceği de incelenebilir. Bir özellik çıkarımı yapılmadan ve makine öğrenmesi algoritmaları kullanılmadan, kurgulanacak 1 boyutlu CNN modeli ile sınıflandırma yapıldığında elde edilen sonucun, özellik çıkarımı yapıldığında elde edilen sonuçlara yakın olması beklenmektedir.

2
MÜZİK VE SAYISAL İŞARET İŞLEME

Sayısal işaret işleme, analog işaret işleme ile işaret işlemenin alt alanlarından biridir. Sayısal işaret işlemenin ilgilendiği alanlardan birisi de ses ve müziklerdir. Müzik sözlü veya enstrümantal seslerin, duyguları güzel bir biçimde ifade eden şekilde birleşmesi olarak ifade edilebilir. Müzikler sayısal bir işaret olarak tanımlanabilir, analog veya sayısal ortamda saklanabilirler. İşaret, fiziksel olarak miktar gösterimi ile temsil edilen bilgi değişkenidir. İşaret analog veya sayısal işlemler yoluyla işlenebilir, depolanabilir veya aktarılabilir. Akustik; katı sıvı ve gazlardaki mekanik dalgaları inceleyen fiziğin bir koludur. Müzik seslerden oluşur, ses ise dalga şeklinde yayılır. Bu ses dalgaları analog ölçüm cihazları ile önce analog olarak ölçülür ve daha sonra sayısal işaret haline dönüştürülür. Bu aşamada bazı teknikler kullanılır. Sayısallaştırma bu tekniklerden bir tanesidir. Sayısallaştırma sırasında örnekleme (sampling) yaparak sonsuz uzunluktaki analog işaret sonlu sayıda sayısal işarete dönüştürülür. Bu her bir örnek belirlenen sayıda bit kadar yer kaplayan sayı dizisine nicemleme (quantization) yoluyla dönüştürülür. Sayısal işaret işlemenin kullanım amaçlarından bir tanesi de müziğin ayırt edici özelliklerinin çıkarılmasıdır. Sayısal işaret işleme teknikleri ile müzikten elde edilen özelliklerin bazıları şunlardır. Sıfır geçiş oranı, spektral merkez, spektral kontrast, spektral bant genişliği, spektral yuvarlama, spektral düzlük, spektral enerji, mel frekans kepstral katsayıları, kroma frekansı, kroma kısa zamanlı Fourier dönüşümü, tonnetz ve dalgacık dönüşümü. İlerleyen bölümlerde bu çalışmada kullanılan yöntemler detaylı olarak açıklanmıştır.

2.1 Müziğin Zaman ve Frekans Alanındaki Özellikleri

Müzikten zaman ve frekans alanında farklı özellik çıkarımları yapılabilmektedir. Bu özellikler müziğin farklı alanlarını temsil eder. Müzik türü sınıflandırma işlemlerinde çıkarımı yapılan bu özelliklerin kullanımına sıkça rastlanmaktadır. Bu özelliklerin bir kısmı aşağıdaki gibidir.

2.1.1 Sıfır Geçiş Oranı - Zero Crossing Rate

Müzik işaretinde, işaretin sıfır noktasından kaç kez geçtiğinin sayılması ile elde edilir[15]. Bu yöntemde işaretin pozitiften negatife ve negatiften pozitife geçişi pencere uzunluğuna oranlanarak hesaplanır.

2.1.2 Spektral Merkez - Spectral Centroid

Spektral merkez frekans alanındaki bir özelliktir. Frekansların ağırlık merkezinin noktasının bulunduğu yerdir [16].

2.1.3 Spektral Kontrast - Spectral Contrast

Bu özellik sesteki güç değişimi bilgisini verir ve işaretteki tepe ve çukur noktalarının değerlerinin farkı alınarak bulunur [17].

$$log\{\frac{1}{\alpha N}\sum_{i=1}^{\alpha N} x_{k,i}\} - log\{\frac{1}{\alpha N}\sum_{i=1}^{\alpha N} x_{k,N-i+1}\} \tag{2.1}$$

2.1.4 Spektral Bant Genişliği - Spectral Bandwidth

Spektral bant genişliği, spektral merkez kullanılarak elde edilir. Frekansların spektral merkeze olan uzaklıklarının ağırlıklı ortalaması alınarak hesaplanır [18].

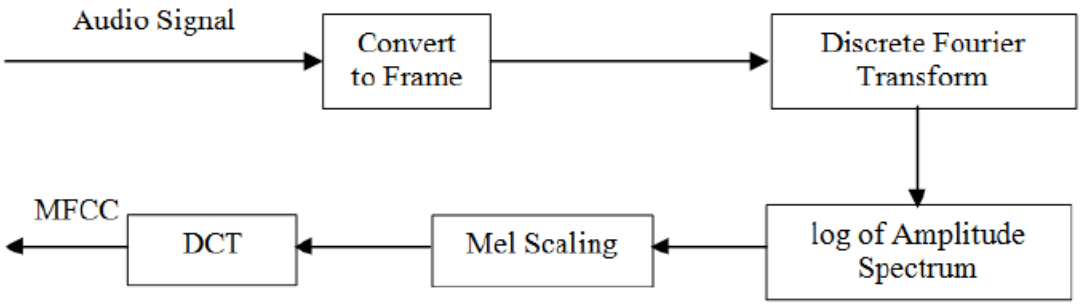
$$\sum_{k} S(k)(f(k) - f_c)^p)^{\frac{1}{p}} \tag{2.2}$$

$S(k) \rightarrow$ k. frekans kutusunun genliği
$f(k) \rightarrow$ k. frekans kutusu
$f_c \rightarrow$ Spektral Merkez

2.1.5 Spektral Yuvarlama - Spectral Rolloff

Spektral yuvarlama belirlenen toplam spektral enerji yüzdesini içinde bulunduran en düşük frekansa denir. Belirlenen toplam spektral enerji yüzdesi değeri genellikle %85'tir [19].

$$\sum_{n=1}^{R_i} M_i[N] = 0.85\ x\ \sum_{n=1}^{N} M_i[N] \tag{2.3}$$



Şekil 2.1 MFCC çıkarım işlem adımları [22]

2.1.6 Spektral Düzlük - Spectral Flatness

Spektral düzlük işaretin büyüklük spektrumunun geometrik ortalamasının aritmetik ortalamasına olan oranı olarak hesaplanır [20].

$$\frac{\sqrt[N]{\prod_k x(k)}}{\frac{1}{N}\sum_k x(k)} \tag{2.4}$$

2.1.7 Spektral Enerji - RMSE

İşaretlerde enerji, işaretin toplam büyüklüğüne eşdeğerdir. Ses işaretleri için bu değer bize sesin ne kadar yüksek olduğu bilgisini verir. Enerjilerin karelerinin ortalamasının karekökünü alarak işarete daha yakın bir sonuç elde edebiliriz.

Enerji formulü:

$$\sum_n |x(n)|^2 \tag{2.5}$$

Enerjinin Karelerinin Ortalamasının Kök Değeri Formülü:

$$\sqrt{\frac{1}{N}\sum_n |x(n)|^2} \tag{2.6}$$

2.1.8 Mel Frekans Kepstral Katsayıları - MFCC

Mel Frekans Kepstral Katsayıları (MFCC), kepstral katsayılarının insanın duyma sistemine uyarlanması ile elde edilir. Kepstral katsayılar doğrusalken insan 1Khz üzerindeki sesleri logaritmik ölçek ile duyar. MFCC'ler çerçeve baskılama (reject), pencereleme, hızlı Fourier dönüşümü, mel frekans sarma ve spektrum adımları ile elde edilebilir [21]. Şekil 2.1'de bu özelliklerin çıkarım adımlarını içeren akış şeması yer almaktadır.

2.1.9 Kroma Kısa Zamanlı Fourier Dönüşümü - Kroma STFT

Sesin kroma değeri müziği incelemek için kullanılan 12 farklı perde sınıfının ne kadar kullanıldığını gösteren bir yöntemdir. Kısa zamanlı Fourier dönüşümü (STFT) kullanılarak kroma özellikleri elde edilebilir [23, 24].

2.1.10 Tonnetz

Ton ağırlık merkezi özelliklerinin hesaplanması ile elde edilir. 12 düğüm ve 24 üçgenden oluşur [25].

2.1.11 Dalgacık Dönüşümü - Wavelet Transform

Fourier dönüşümünde pencerenin büyüklüğünün büyümesi ile zaman belirsizliğinin artması problemine çözüm olarak kullanılan bir yöntemdir. Bu sorun çoklu çözünürlük analizi yapılarak aşılabilir. Dalgacık dönüşümü, çoklu çözünürlük analizine bir örnektir. Bu yöntem sayesinde ses işareti içerisinde hangi frekansların olduğunun bulunmasının yanı sıra nerede bulundukları bilgisi de elde edilir [24, 26].

2.2 Makine Öğrenmesi Yöntemleri

Müzikten elde edilen ayırt edici özelliklerin anlamlı olabilmesi için, bu müzik özellikleri kullanılarak makine öğrenmesi algoritmaları eğitilir ve elde edilen modeller ile müzik türlerinin tahmini yapılır. Kullanılan başlıca makine öğrenmesi algoritmaları; K En Yakın Komşu, Naive Bayes, Karar Ağacı, Destek Vektör Makinesi, Rastgele Orman'dır.

2.2.1 K En Yakın Komşu - kNN

K en yakın komşu algoritması mesafe tabanlı denetimli öğrenme algoritmasıdır. Bu yöntemde model oluşturulmaz. Test edilecek veri, algoritmanın daha önce gruplandırdığı verilerden hangisine daha yakın ise o gruba dahil olur. Mesafe için Hamming, Manhattan ve Öklidyen gibi farklı uzaklık hesaplama teknikleri vardır. K en yakın komşu için genellikle Öklidyen (Euclidean) ve Manhattan uzaklıkları kullanılır [27].

2.2.2 Naive Bayes

Naive Bayes olasılıksal denetimli öğrenme algoritmasıdır. Önceden toplanan verileri kullanarak ön olasılık hesaplaması yapar ve test edilecek verilerde geleceğe yönelik

tahmin yapan bir model oluşturur ve bu model ile örnekler sınıflandırılır [28].

2.2.3 Karar Ağacı - Decision Tree

Karar ağaçları denetimli öğrenme algoritmalarıdır. Özelliklerin önem sırasına göre ağaç yapısında bir model oluşturarak yeni gelecek test verilerini bu modele göre sınıflandırır [29].

2.2.4 Destek Vektör Makinesi - Support Vector Machine

Destek Vektör Makinesi (SVM), model tabanlı denetimli öğrenme algoritmalarından biridir. SVM eğitim verilerini uzayda noktalar ile eşler. Daha sonra farklı iki kategori arasındaki sınır noktalarını dikkate alarak iki sınıfın birbirinden ayrılması için gerekli en uzak noktalardan geçen doğru veya eğri ile verileri iki farklı gruba ayırır. Test verisi bu gruplardan hangisine aitse o grubun etiketi ile sınıflandırma gerçekleşir [8, 30–32].

2.2.5 Rastgele Orman - Random Forest

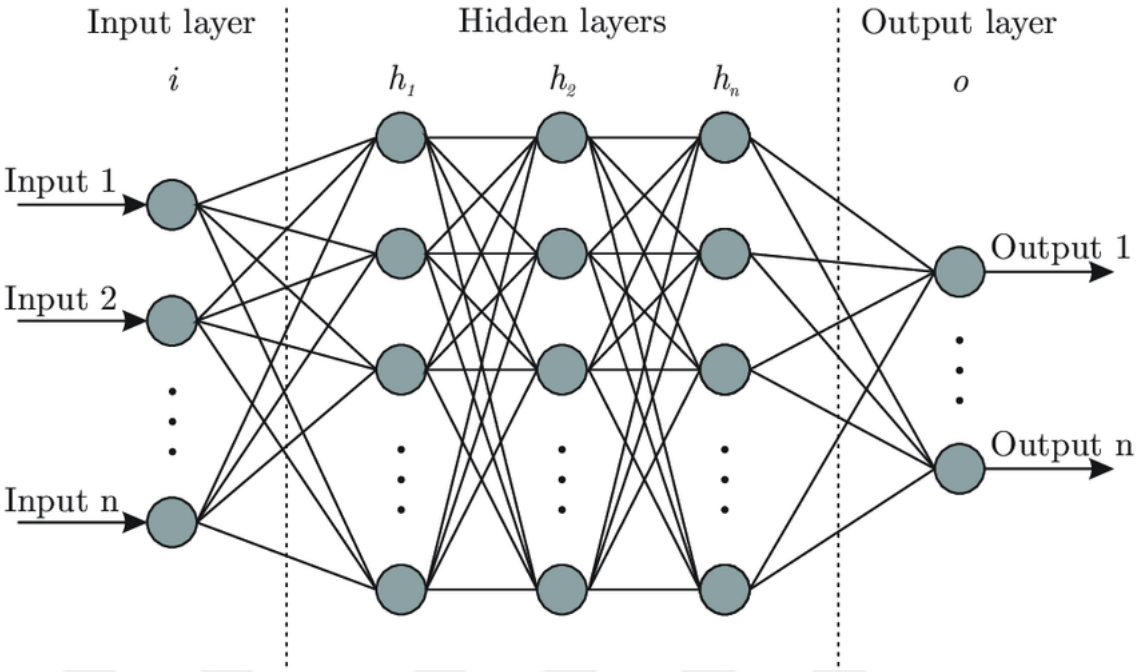
Rastgele Orman birden fazla karar ağacının birleşmesi ile oluşur. Karar ağaçlarında büyük problem olan aşırı öğrenme (over fitting) problemini çözmek için, veri setinden ve verilerin özelliklerden farklı seçimler yaparak birden çok ağaç eğitilir. Daha sonra her bir ağaçtan gelen sonuca göre, eğer çözülmesi gereken problem sınıflandırma ise en çok seçilen sınıf, eğer sayısal bir problem ise verilen sonuçlara uygulanan istatiksel yöntemler (ortalama, ortanca, standart sapma) ile sonuç belirlenir [7, 33].

2.3 Müzik Türü Sınıflandırma için Derin Öğrenme

Derin öğrenme yöntemleri, bilgisayarların donanımsal olarak hızla gelişmesi sonucu kullanımı günden güne artmış ve araştırmacılar önceden makine öğrenmesi gibi yöntemler ile çözüm buldukları problemlere derin öğrenme yöntemlerini uygulamaya başlamışlardır. Araştırmacıların buradaki amacı, derin öğrenme yöntemleri ile daha iyi bir sonuç alınıp alınamayacağını görmek ve özellik çıkarımı aşamasının otomatize edilip edilemeyeceğini test etmektir.

2.3.1 Derin Yapay Sinir Ağı

Yapay sinir ağları nöron adı verilen ve birbirleri ile bağlantılı bileşenlerden oluşan yapıya denir. Biyolojik sinir hücrelerinin zayıf bir benzetmesidir de denebilir. Her bir bağlantı sinir hücrelerinde olduğu gibi bir nörondan başka bir nörona işaret taşıma



Şekil 2.2 Yapay Sinir Ağları Genel Mimarisi

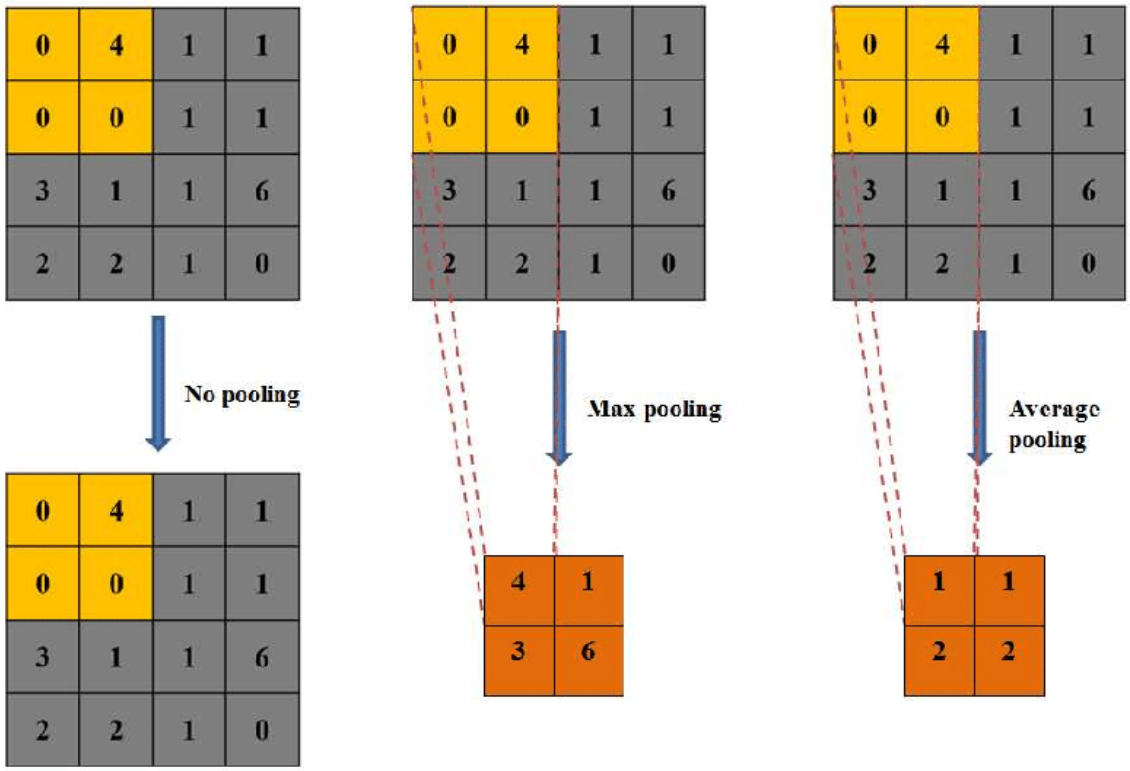
özelliğine sahiptir. Bu modelde eğitim her bir nöronun ağırlıklarının değiştirilerek istenilen sonuca ulaşılmaya çalışılması ile gerçekleşir. 1958'de ilk yapay sinir ağı olan algılayıcı (perceptron) bulunmuştur [34]. İlk çok katmanlı işlevsel yapay sinir ağı ise Ivakhnenko ve Lapa tarafından 1965 yılında yayınlanmıştır [35]. Yapay sinir ağlarının genel mimarisi Şekil 2.2'de gösterilmiştir. Derin yapay sinir ağları klasik yapay sinir ağlarından farklı olarak gizli katman bölümünde, fazla sayıda gizli katmana, evrişim (convolution), havuzlama (pooling) işlemlerine sahip olur.

2.3.2 Evrişimli Sinir Ağı (CNN)

Derin öğrenme ağlarının bir türü olan CNN, resim desenleri tanımlamada başarıları kanıtlanmış bir yapıdır. Müzik türü sınıflandırma alanında bu ağın kullanılabilmesi için seçilen özelliklerin bir resim şeklinde temsil edilmesi gereklidir. Bunun için en çok seçilen yöntem zaman frekans analizini görsel hale getiren spektrogramlar veya mel-spektrogramlardır [12]. Bu amaç için son araştırmalarda kullanılan bir diğer özellik ise MFCC'lerdir [8, 10].

CNN belirli katmanlardan oluşur. Bunlar evrişim, havuzlama (pooling), aşağı örnekleme (down-sampling), bırakma (dropout) ve aktivasyondur. Genel CNN mimarisi Şekil 2.4'te incelenebilir.

Evrişim katmanı gelen girdiye verilen ölçekte bir matris filtresi uygular. Verilen girdideki her bir element filtre boyutu kadar girdideki örtüşen elementler ile çarpılır



Şekil 2.3 Maksimum Örnekleme ve Ortalama Örnekleme Karşılaştırma [36]

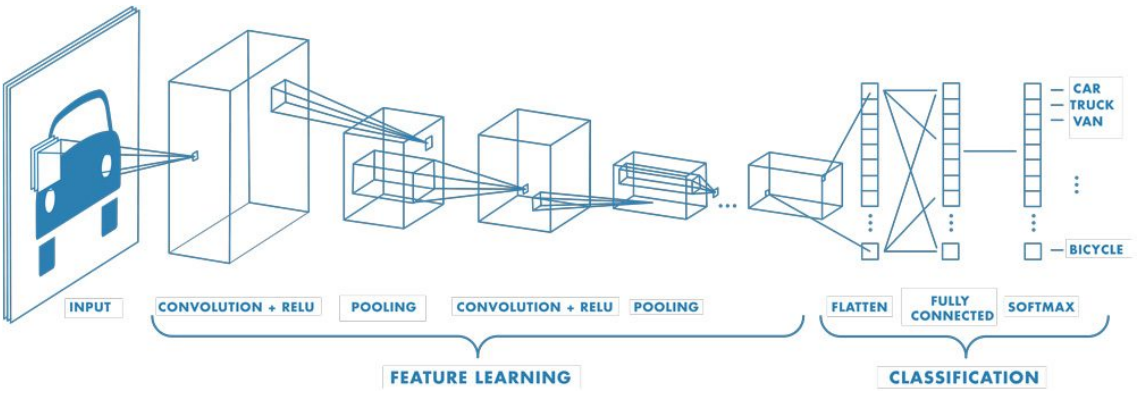
ve bu değerlerin toplamı evrişim değerini verir [12].

Aşağı örnekleme (Down Sampling) işaret işlemede bir işaret dizisi üzerinde belirli yöntemler ile verinin boyutunu küçültmek ve aynı zamanda veriyi bir şekilde orijinal veriye yaklaşık olarak kodlamaktır.

Havuzlama (Pooling) aşağı örnekleme yöntemlerinden birisidir. Verilen belirli bir pencere boyutu içerisinde kalan maksimum veya ortalama değeri tutarak işleme sürecinin ve depolama alanının azaltılmasını sağlar [12]. Şekil 2.3 ile bu iki yolun uygulanması sonucu elde edilen çıktılara örnek verilmiştir.

Bırakma (dropout) işlemi yapay sinir ağlarında sıkça görülen aşırı öğrenme (overfitting) problemine çözüm olarak kullanılan bir yöntemdir. Her bir öğrenme çevriminde rastgele nöronların ağırlıklarının değişmemesi sağlanarak eğer eğitime katkısı olmayan çekinik nöronlar varsa bu nöronlarında eğitime katılması sağlanır. Bırakma sinir ağlarının katmanlarına ön katman olarak eklenir ve o çevrimde ağırlıkları değişmeyecek olan nöronların bir önceki katman ile iletişimini keser [37].

Aktivasyon adımında doğrusal ya da doğrusal olmayan aktivasyon fonksiyonları kullanılabilir. Başarım açısından doğrusal olmayan problemlere etkili çözümler üreten bir sinir ağı oluşturmak için, modele doğrusal olmama ve türevlenebilme özelliği



Şekil 2.4 CNN genel mimarisi örneği [38].

kazandırılmalıdır [12].

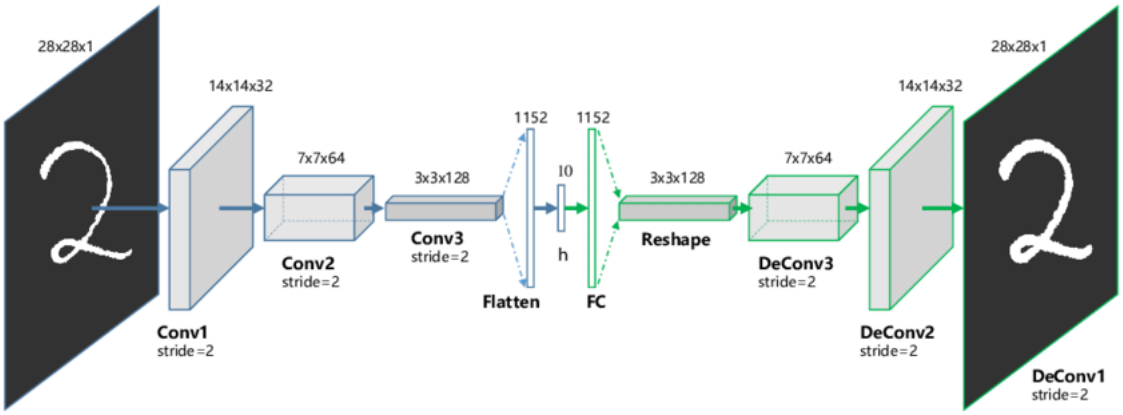
2.3.3 Otomatik Kodlayıcı (AE)

Otomatik kodlayıcı, yapay sinir ağlarının bir türüdür. Amacı, verilen girdideki gereksiz bilgilerden kurtularak girdiyi olabildiğince küçük bir yapı haline getirmek ve daha sonra bu yapıyı kullanarak baştaki girdiye en az hata ile geri dönüş yapabilmektir [39, 40].

İlk olarak 1985 yılında Rumelhart ve arkadaşları tarafından tanıtılmıştır [41]. Bu çalışmada araştırmacıların amacı denetimli öğrenme kullanmadan girdideki verileri eğitici olarak kullanarak, denetimsiz bir şekilde geri yayılım yapımı problemini çözmektir. Otomatik kodlayıcılar denetimli öğrenme sürecinde özellik çıkarımı aşamasını otomatize etmek amacıyla da kullanılan algoritmalardır. Bu amaç doğrultusunda temel bileşen analizi (Principal Component Analysis (PCA)) gibi yöntemler kullanılsa da otomatik kodlayıcılar bu yöntemlerin çıkardığı özelliklerden çok daha ayrıntılı ve mantıklı sonuçlar verebilme özelliğine sahiptir.

Yüz tanıma alanında yapılan bir çalışmada PCA ve AE karşılaştırılmış, elde edilen sonuçlara göre PCA basit ve düşük işlem maliyetine sahipken AE çok katmanlı yapısı ile modele doğrusal olmama kattığı için sonuçlarda gözle görülür bir iyileşme olmuştur. Diğer yandan, PCA'e göre çok daha fazla işlem maliyeti gerektirdiği tespit edilmiştir [42]. Ancak günümüzde bilgisayar mimarisinin ve donanımının gelişimi sayesinde bu yüksek işlem maliyeti problemi ortadan kalkmaya başlamıştır.

Müzik alanında yapılan çalışmalarda otomatik kodlayıcılar genellikle özellik çıkarımı, müzik oluşturma, gürültü giderme veya istenmeyen seslerin bastırılması gibi alanlarda kullanılmıştır [43, 44]. Zhao ve diğerleri yaptıkları çalışmada evrişimli gürültü giderme otomatik kodlayıcı (Convolutional Denoising Autoencoder) kullanarak sözün üzerine eklenmiş müziğin kaldırılmasına çalışmışlardır [45]. Yapılan başka bir



Şekil 2.5 Otomatik kodlayıcı ve CNN genel mimarisi örneği [47]

çalışmada müzik üretimi için otomatik kodlayıcıların bir çeşidi olan varyasyonel otomatik kodlayıcıyı kullanmışlardır [46].

Şekil 2.5'de otomatik kodlayıcının genel yapısı görülebilir. Otomatik kodlayıcılar 3 bölümden oluşur. Kodlayıcı, gizli temsil ve kod çözücü. Otomatik kodlayıcılar yeniden oluşturma hatasının en aza düşürülmesi üzerine eğitilirler. Girdi ilk olarak kodlayıcı bölüme verilir. Kodlayıcı bu girdiyi gizli temsil ya da kod denilen bölüme eşler. Daha sonra gizli temsil, kod çözücü bölüme girdi olarak verilir ve ilk baştaki girdinin çıktı olarak alınması hedeflenir.

3
DENEYSEL SONUÇLAR

Bu tez çalışmasında müzikten sayısal işaret işleme yöntemleri ile elde edilen özelliklerin sınıflandırma sonuçları, otomatik kodlayıcı eğitilerek elde edilen özelliklerin sınıflandırma sonuçları ve derin öğrenme modeli eğitilerek elde edilen sınıflandırma sonuçları karşılaştırılmıştır. Bunun için müzikler her yöntem için farklı ön işlemlerden geçirilmiştir. Bundan sonraki bölümde sırası ile kullanılan veri seti, yapılan ön işleme çalışmaları, sayısal işaret işleme yöntemleri, otomatik kodlayıcı, derin öğrenme, sınıflandırma ve kümeleme anlatılmıştır.

3.1 Veri Seti

Bu tez çalışmasında ilk olarak kullanılacak olan müzik veri seti GTZAN olarak belirlenmiştir. 2014 yılında yapılan bir çalışmada bu veri setinin bazı dezavantajlarından bahsedilse de günümüzde hala popüler ve çoğunlukla test amacıyla seçilen bir veri setidir. [48]. Önceki çalışmaların birçoğunda bu veri setinin kullanılması, sonuçların karşılaştırılabilmesi açısından faydalıdır. Kullanılacak veri seti 10 tür müzik barındırmaktadır. Bu türler blues, klasik, country, disko, hiphop, jazz, metal, pop, reggae, rock'tır. Her bir türden 30 saniyelik 100 farklı müzik vardır. Veri seti 2002 yılında yapılan bir çalışmada oluşturulmuştur [2].

3.2 Kullanılan Araçlar ve Programlar

Bu çalışmada python programlama dili kullanılmıştır. Sayısal işaret işleme yöntemleri ile özellik çıkarımı için librosa kütüphanesinden yararlanılmıştır. Farklı makine öğrenmesi algoritmalarını test edebilmek için scikit-learn kütüphanesi kullanılmıştır. Modellerin oluşturulması, eğitilmesi ve testi için ise tensorflow, keras ve google colab kullanılmıştır.

3.3 Ön İşleme

Veri seti incelendiğinde bazı müziklerin 30 saniyeden biraz fazla bazılarının da biraz az olduğu gözlemlenmiştir. Kullanılacak model ve yöntemlerde, aynı veri setini kullanan daha önceki çalışmalar incelendiğinde, CNN ve derin öğrenme yapıları ile yapılan çalışmalarda müziklerin 5 ya da 6 saniyelik parçalara bölündüğü ve işlemlerin bu parçalar üzerinde yapıldığı gözlemlenmiştir [9]. Bunun sebeplerinden bir tanesi yapılan bir araştırmada, 3 saniyelik bir müzik parçasının müzik türünü tanımlamak için yeterli olduğu, uzunluk arttıkça tanımlama oranının artmadığından bahsedilmesidir [2]. Bir diğer sebebi ise veri setindeki yetersiz örnek dezavantajını veriyi bölüp çoğaltarak az da olsa ortadan kaldırmaktır. Bu çalışmada sayısal işaret işleme yöntemlerinde müzikler parçalara ayrılmadan uygulanırken, otomatik kodlayıcı ve derin öğrenme yöntemleri için müzikler 5 saniyelik parçalara bölünmüştür.

3.3.1 Sayısal İşaret İşleme için Verinin Ön İşlenmesi

Sayısal işaret işleme için bir ön işleme gerek duyulmadı. Zaman ve frekans uzayına ait özellikler çıkarılmış ve bu özellikler ile bir özellik vektörü oluşturabilmek için ortalama, standart sapma, çarpıklık ve basıklık (skewness and kurtosis) ve ortanca değer gibi istatistiksel tanımlayıcılar kullanılmıştır.

3.3.2 Otomatik Kodlayıcı için Verinin Ön İşlenmesi

Bu tez çalışmasında otomatik kodlayıcılar CNN mimarisinde oluşturuldu. Alınan bir resmin aynısını üretmeyi amaçlayan bu modele girdi olarak 5 saniyelik bölümlere ayrılmış müziklerden elde edilen MFCC özellikleri resim haline getirilerek verilmiştir. Resimlerin otomatik kodlayıcıya girdi olabilmesi için öncelikle müziklerden MFCC özellikleri elde edildi. MFCC özellik çıkarımı için yapılan işlemler sırası ile şu şekildedir:

- İşaret kısa pencere aralıklarına bölünür
- İşaretin pencerelenmiş halinin Fourier Dönüşümü alınır
- İlk adımdan elde edilen spektrum güçleri mel ölçeğine eşlenir
- Her bir mel frekansı için güçlerin logaritması alınır
- Elde edilen mel logaritma güçleri listesinin ayrık kosinüs dönüşümü alınır
- Sonuçta elde edilen spektrumun genlikleri MFCC'lerdir

Elde edilen MFCC özelliklerinin resim olarak temsili Şekil 3.5'de verilmiştir. MFCC özellikleri çıkarılırken, mel spektrogram üretmek için pencere boyutu olarak 2048 ve atlama uzunluğu olarak da 512 kullanıldı. Önceki çalışmalardan edinilen bilgiler doğrultusunda ilk 13 MFCC özelliği seçildi [8, 49]. MFCC özelliklerinin son boyutu 13 x 259'dur. Burada 259 segment başına MFCC vektörlerinin sayısıdır. Daha sonra bu resim otomatik kodlayıcıya uygun olacak biçimde 128 x 128 boyutunda yeniden boyutlandırıldı. Rengi gri ölçeğe çevrildi. Temsil ettiği değerler float veri tipine dönüştürüldü ve normalize edildi. Normalize işlemi her bir değerin 255'e bölünmesi ile değerlerin 0-255 aralığından 0-1 aralığına dönüştürülmesidir. Normalize edilmesinin sebebi sinir ağları modellerinin normalize edilmiş verilerde daha iyi sonuçlar vermesidir [50].

3.3.3 Derin Öğrenme için Verinin Ön İşlenmesi

Derin öğrenme modeline girdi olarak ham ses dosyası verileceği için herhangi bir ön işlem uygulanmadı. Yalnızca müzikler, modelin kabul ettiği uzunlukta parçalara bölündü.

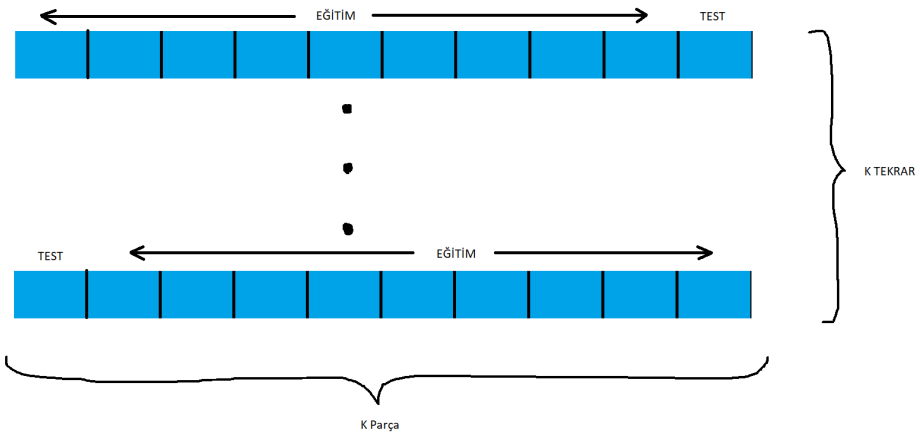
3.4 Sayısal İşaret İşleme

Bir önceki bölümde anlatılan özellik çıkarımı aşamasından elde edilen özellikler farklı makine öğrenmesi algoritmalarına verilerek sınıflandırma işlemi yapıldı. Bu aşamada elde edilen özellikler 2 farklı şekilde test edildi. İlk olarak bütün özellikler ile bir özellik vektörü oluşturuldu ve sınıflandırma bu şekilde yapıldı. İkinci olarak, yalnızca MFCC özellikleri kullanılarak aynı test tekrar edildi. Testler için toplam 8 farklı makine öğrenmesi algoritması kullanıldı. Bunlar: Çok Katmanlı Algılayıcı (Multi Layer Perceptron (MLP)), lojistik regresyon, doğrusal ayrımcılık analizi (Linear Discriminant Analysis (LDA)), K en yakın komşu (K Nearest Neighbour (KNN)), gauss naive bayes (Gaussian Naive Bayes (GaussianNB)), gradyan arttırma (Gradient Boosting) ve destek vektör makinesi (Support Vector Machine (SVM))'dir.

Testler 10 katlamalı çapraz doğrulama kullanılarak yürütüldü. Bu yöntem makine öğrenmesi modellerinin hata oranını daha iyi test edebilmek için uygulanır. Şekil 3.1'de k katlamalı çapraz doğrulama işlemi gösterilmiştir.

3.4.1 K Katlamalı Çapraz Doğrulama Yöntemi (K-fold Cross Validation)

- Test edilecek veri K eşit parçaya bölünür
- 1 parça test için geri kalan K-1 parça eğitim için kullanılır



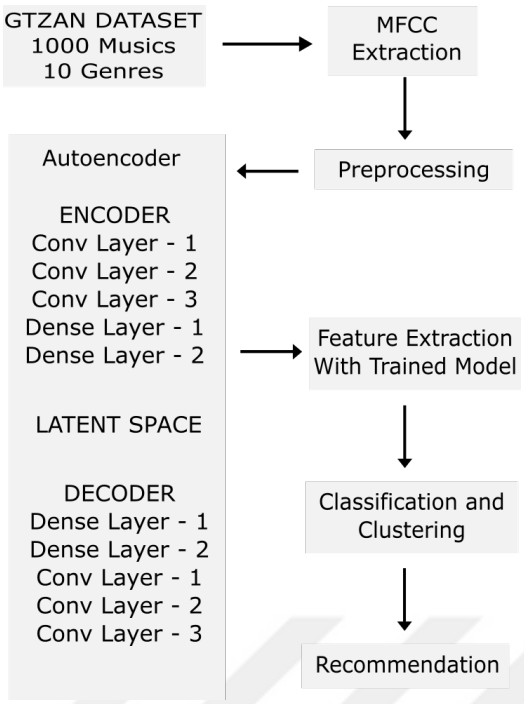
Şekil 3.1 K katlamalı çapraz doğrulama işlemi

- 2. adım K kez tekrar eder ve her seferinde test için kullanılan parça bir sonraki parça olarak seçilir
- 3. adımdaki tekrarlardan elde edilen sonuçlara dayalı olarak beklenen bir performans metriği belirlenir (ortalama kare hatası, yanlış sınıflandırma hata oranı gibi.)

3.5 Otomatik Kodlayıcı

Bu tez için kullanılan otomatik kodlayıcının kodlayıcı bölümünün modeli, daha önce yapılan bir çalışmada oluşturulan ve kullanılan MusicRecNet isimli bir model ile aynı özelliklere sahip olacak şekilde tasarlandı [9]. Kod çözücü bölüm için ise filtre sayısı kodlayıcı bölümdekinin tersi olacak şekilde tasarlandı. Bu sayede girişteki boyuta tekrar ulaşılabilmesi sağlandı. MusicRecNet'ten farklı olarak otomatik kodlayıcı yapısında amaç resmi az hata oranı ile yeniden üretmek olduğu için kayıp fonksiyonu olarak ortalama kare hatası (Mean Squared Error (MSE)) seçildi. Aynı zamanda son katmanda sınıflandırma yapılmayacağı için softmax algoritması yerine sigmoid algoritması kullanıldı. Son olarak kod çözücü bölüm için maksimum ortaklama (Max Pooling) yerine sık örnekleme (Up Sampling) katmanları kullanıldı. Bu tez çalışmasında kullanılan otomatik kodlayıcının akış şeması Şekil 3.2'de incelenebilir.

Bu çalışmada gizli temsil bölümünün boyutu için 64, 128, 256, 512, 1024 olmak üzere 5 farklı değer test edilmiştir. Gizli temsil boyutunun değiştirilmesindeki amaç sıkıştırma oranının sonuçlara etkisini incelemektir. Kullanılan otomatik kodlayıcının model parametreleri Tablo 3.1'de verilmiştir.



Şekil 3.2 Otomatik Kodlayıcı Akış Şeması

Otomatik kodlayıcıyı eğitmek için 5128 rastgele seçilmiş müzik kullanıldı. Test ve doğrulama içinse, sırası ile 570 ve 300 rastgele müzik kullanıldı. Burada toplam veri sayısının 6000 değil 5998 olmasının sebebi veri setindeki 2 müziğin son kısımlarının 5 saniyeden daha kısa olmasıdır.

Tablo 3.1 Otomatik Kodlayıcı Model Parametreleri

Parametre Türü	**Otomatik Kodlayıcı**
girdi boyutu	128 x 128
parti büyüklüğü (batch Size)	64
kodlayıcı filtre sayısı	32, 64, 128
kod çözücü filtre sayısı	128, 64, 32
kernel boyutu	3 x 3
kayıp fonksiyonu	MSE
aktivasyon fonksiyonu	sigmoid
optimize edici	Adam

Modelin kaynak kodu Ek-A'da verilmiştir.

3.6 Derin Öğrenme ve 1 Boyutlu Derin CNN

Bu tez çalışmasında son olarak kullanılan yöntem bir boyutlu Derin CNN ağıdır. İki boyutlu CNN'ler genellikle resim gibi veriler üzerinde kullanılır. Bu ağların iki boyutlu olarak adlandırılmasının sebebi üzerlerinde dolaştırılan kernellerin 2 farklı boyut boyunca kaydırılmasıdır. Örnek iki boyutlu CNN ve kernel Şekil 3.4'de

verilmiştir. Bir boyutlu CNN'in farkı ise kernel'in tek boyut boyunca kaydırılmasıdır. Bir boyutlu CNN'ler genellikle zaman serisi verileri (time-series data) ile beraber kullanılır. Müzik ve yazının zaman serisi şeklinde temsil edilebilmesi bir boyutlu CNN'in uygulanabilirliğini sağlar.

Kullanılan bir boyutlu CNN 110250 büyüklüğünde tek boyutlu bir diziyi girdi olarak kabul etmektedir. 4 adet evrişim bölümünden oluşur. Her bir bölüm; 1 adet bir boyutlu evrişim katmanı, 1 adet yığın normalleştirme (batch normalization) katmanı, 1 adet aktivasyon katmanı ve son olarakta 1 adet max pooling katmanından oluşur. Bu bölümlerden sonra çıktı bir adet lambda katmanına aktarılır ve gelen verilerin ortalamasını alarak bir sonraki katmana aktarır. Bu işlemde gelen girdinin boyutu 107 x 512 iken çıktının boyutu 512'dir. En son katman olan yoğun (dense) katmana gelen veriler ile sınıflandırma yapılır. Modelin genel yapısı Şekil 3.3'de verilmiştir. Modelde kullanılan parametreler ve fonksiyonlar Tablo 3.2'de bulunabilir.

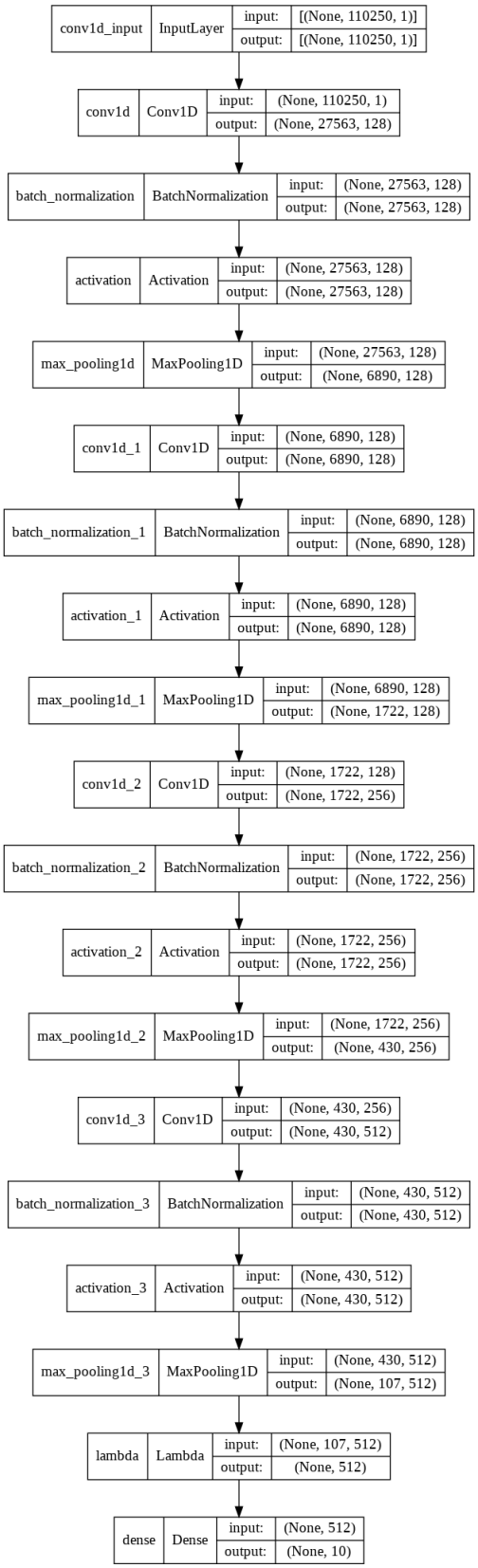
Tablo 3.2 1 Boyutlu CNN Model Parametreleri

Parametre Türü	**1 Boyutlu CNN**
girdi boyutu	110250 x 1
parti büyüklüğü (batch Size)	64
kernel başlatıcı (initializer)	glorot_uniform
kernel düzenleyici (regularizer)	l2
kernel boyutu	80, 3, 3, 3
büyük adım boyutu (strides)	4, 1, 1, 1
pooling size	4, 4, 4, 4
kayıp fonksiyonu	categorical_crossentropy
yoğun katman boyutu (Dense Layer Size)	10
aktivasyon fonksiyonu	softmax
optimize edici	Adam

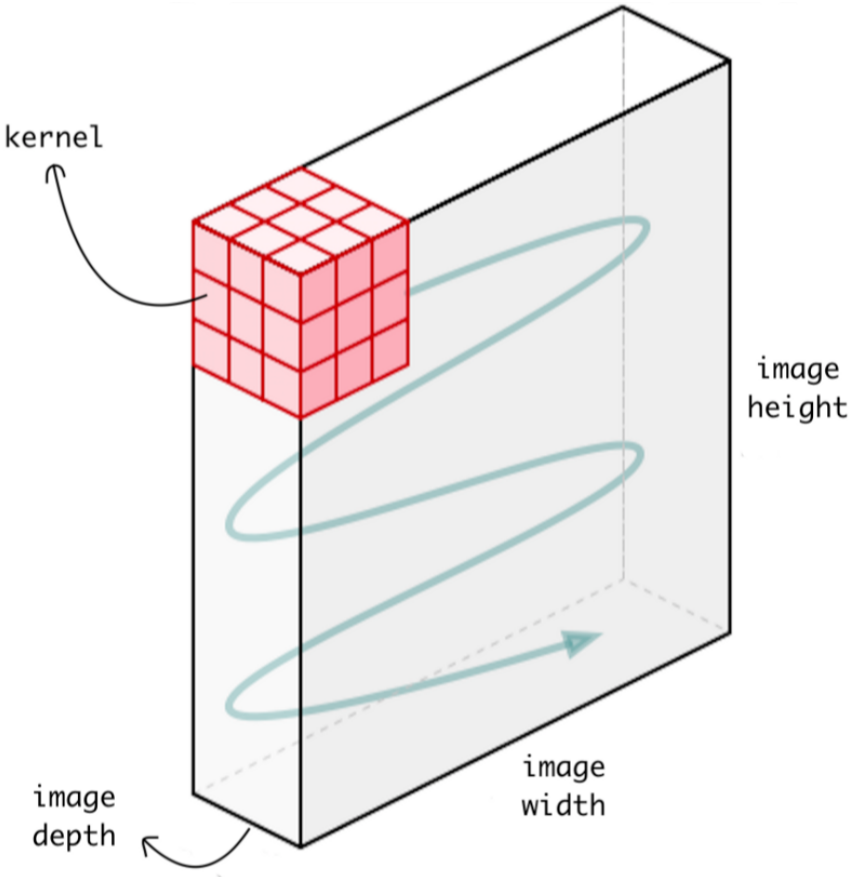
Modelin kaynak kodu Ek-A'da verilmiştir.

3.7 Sayısal İşaret İşleme & Otomatik Kodlayıcı Sınıflandırma ve Kümeleme Sonuçları

Bu bölümde yukarıda anlatılan özellik çıkarımı, makine öğrenmesi, yapay zeka ve derin öğrenme yöntemlerinin gerçeklenmesi ile elde edilen sonuçlar paylaşılacaktır. İlk olarak sayısal işaret yöntemleri ile müzikten çıkarılan özellikler toplamda uzunluğu 606 olan bir vektör şeklinde birleştirildi. Elde edilen özellik vektörleri ile, 10-kat çapraz doğrulama (10-fold cross validation) kullanılarak testler yapıldı. Elde edilen başarı oranları Tablo 3.3'de incelenebilir. En iyi sonucu SVM algoritması %88 ile vermiştir.



Şekil 3.3 Kullanılan 1 Boyutlu CNN Modeli



Şekil 3.4 2 boyutlu CNN ve kernel örneği [51]

Tablo 3.3 Müzikten Çıkarılan Bütün Özellikler Kullanılarak Yapılan Sınıflandırma Sonuçları K=10

Algorithm	Avg Accuracy (%)	Max Accuracy (%)
MLP	74.1	78
Logistic Regression	80.6	84
Random Forest	77.7	84
LDA	81.4	86
KNN	73.2	79
GaussianNB	64.7	73
Gradient Boosting	75.6	79
SVM (Poly)	80.7	86
SVM (Linear)	77.4	86
SVM (RBF)	**81.9**	**88**

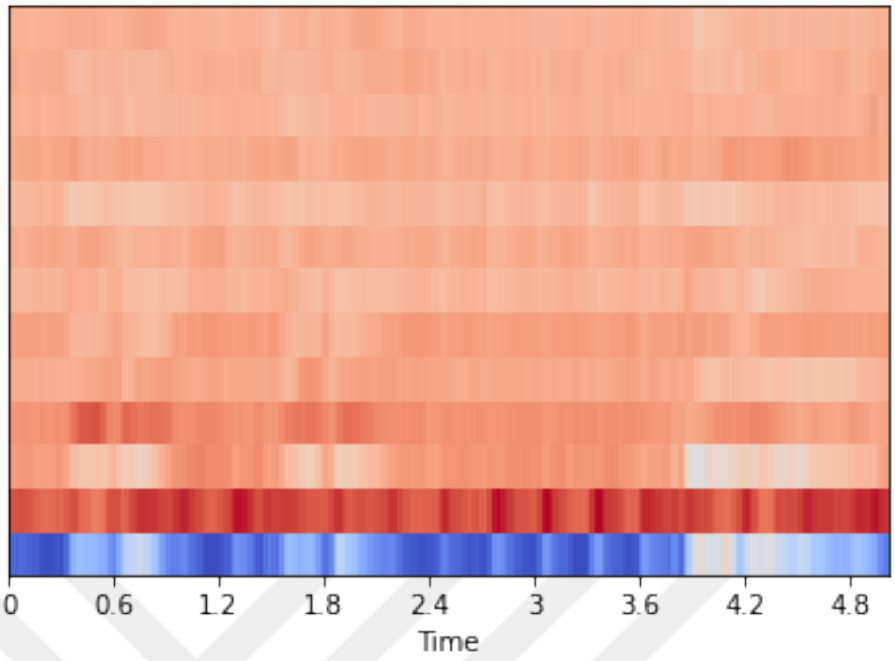
Yapılan ikinci testte, yalnızca müziklerden elde edilen MFCC özellikleri kullanılmıştır. Tablo 3.4'den görüleceği üzere en iyi sonuç LDA algoritması ile alınmıştır.

Tablo 3.4 Müzikten Çıkarılan Özelliklerden Yalnızca MFCC ile Yapılan Sınıflandırma Sonuçları K=10

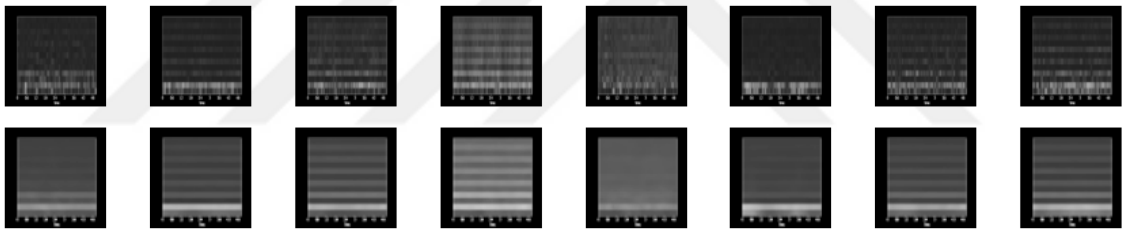
Algorithm	Avg Accuracy (%)	Max Accuracy (%)
MLP	69.7	79
Logistic Regression	72.5	78
Random Forest	70.3	78
LDA	**75.3**	**84**
KNN	61.8	67
GaussianNB	60.9	67
Gradient Boosting	70.3	79
SVM (Poly)	66.6	73
SVM (Linear)	69.4	85
SVM (RBF)	73.4	81

Yapılan bu iki test karşılaştırıldığında ikinci testte bir özellik hariç bütün özellikler çıkarılmasına rağmen neredeyse ilk testteki başarı oranları ile aynı başarı oranlarına sahip olunması MFCC özelliklerinin yalnız başına müziğin ayırt edici birçok özelliğini temsil ettiğini gösterir. Daha sağlıklı ve güvenilir sonuç almak ve uç örnekler ile karşılaşıldığında modelin yanlış cevap vermemesi için bütün özellikler kullanılarak daha kuvvetli bir model oluşturulabilir.

Otomatik kodlayıcı ile yapılan testte ise gizli temsil boyutu 64, 128, 256, 512 ve 1024 için elde edilen sınıflandırma doğruluk sonuçları Tablo 3.5, 3.6, 3.7 3.8 ve 3.9'te verilmiştir. Tablolar incelendiğinde başarı oranının 512 boyutlu temsil boyutuna kadar artıp daha sonra azalmaya başladığı görülmektedir. Aynı zamanda genel olarak başarı oranının oldukça düşük olduğu görülebilir. Bunun nedeni otomatik kodlayıcının



Şekil 3.5 Örnek MFCC resim temsili



Şekil 3.6 MFCC Yeniden Oluşturma Örneği

özellik çıkarımı yerine gürültü silme yönelimiyle çalışması olabilir. Aralarında en iyi sonucu veren MLP algoritması ile 512 boyutlu gizli temsile ait model olduğu için devam eden testlerde bu model kullanılmıştır.

İlk yapılan sayısal işaret işleme testindeki MFCC sonuçları ile bu sonuçlar karşılaştırılacak olursa MFCC özelliklerinin sıkıştırılmasının veri kaybına yol açtığı ve bu yüzden başarım oranının düştüğü yönünde yorum yapılabilir. Bütün bu bilgiler ve yorumlar sonucunda, MFCC özelliklerinin içerisindeki her bilginin müziğin belirleyici özelliklerini taşıdığını ve bu özelliklerin sıkıştırılma yoluyla kaybolması sonucu başarım oranının düştüğü sonucuna varılmıştır.

Aynı gizli temsil boyutu kullanılarak elde edilen özellik vektörleri ile yapılan kümeleme sonucu Tablo 3.10 ile gösterilmiştir. Kümeleme sonucunda müzik türlerinin genellikle 2 veya 3 kümede yoğunlaştıkları görülmektedir. Küme açısından

Tablo 3.5 64 boyutlu Gizli Temsil Kullanılarak Elde Edilen Sınıflandırma Sonuçları

Classifier	**One Pass**	**Avg accuracy, %**	
		Five-fold	Ten-Fold
MLP	49	40	43
Logistic Regression	44	38	39
Random Forest	43	36	36
LDA	45	39	40
KNN	37	33	34
SVM	49	34	36

Tablo 3.6 128 boyutlu Gizli Temsil Kullanılarak Elde Edilen Sınıflandırma Sonuçları

Classifier	**One Pass**	**Avg accuracy, %**	
		Five-fold	Ten-Fold
MLP	57	41	42
Logistic Regression	49	42	43
Random Forest	43	37	38
LDA	47	40	42
KNN	35	30	31
SVM	50	34	36

Tablo 3.7 256 boyutlu Gizli Temsil Kullanılarak Elde Edilen Sınıflandırma Sonuçları

Classifier	**One Pass**	**Avg accuracy, %**	
		Five-fold	Ten-Fold
MLP	57	43	44
Logistic Regression	50	43	44
Random Forest	42	34	36
LDA	49	39	40
KNN	32	27	27
SVM	49	34	37

Tablo 3.8 512 boyutlu Gizli Temsil Kullanılarak Elde Edilen Sınıflandırma Sonuçları

Classifier	**One Pass**	**Avg accuracy, %**	
		Five-fold	Ten-Fold
MLP	58	42	43
Logistic Regression	51	43	44
Random Forest	42	37	36
LDA	48	32	36
KNN	31	25	26
SVM	48	33	36

bakıldığında ise 3 ve 8 numaralı kümeler hariç diğer kümelerde bir türün baskın olduğu gözlemlenmiştir. Örneğin 1 numaralı kümede baskın olan tür klasik (classical) iken 2 numaralı kümede pop baskın olan tür olmuştur. Her ne kadar

Tablo 3.9 1024 boyutlu Gizli Temsil Kullanılarak Elde Edilen Sınıflandırma Sonuçları

Classifier	**One Pass**	**Avg accuracy, %**	
		Five-fold	Ten-Fold
MLP	57	41	42
Logistic Regression	50	42	43
Random Forest	42	36	37
LDA	47	18	17
KNN	28	24	25
SVM	48	33	37

Tablo 3.10 512 boyutlu Gizli Temsil Kullanılarak Elde Edilen Kümeleme Sonuçları

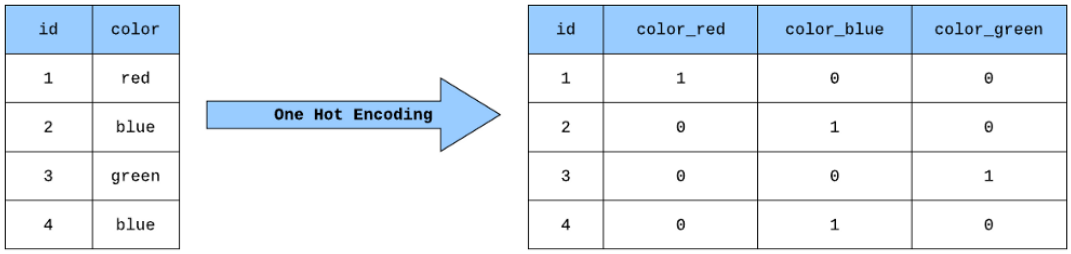
Music Genres	**Cluster Numbers / Genre Count**									
	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
blues	6	31	21	13	111	91	68	14	160	80
classical	-	196	3	30	205	-	105	-	26	
country	37	11	91	15	66	32	230	-	111	4
disco	51	2	232	22	5	68	35	1	166	11
hiphop	151	-	120	64	3	94	27	3	103	30
jazz	27	82	35	21	175	71	116	-	49	17
metal	8	1	35	21	3	36	118	143	78	153
pop	187	4	304	29	14	11	39	1	4	-
reggae	114	4	44	85	18	191	37	-	98	2
rock	25	4	110	22	16	65	159	30	137	24

kümelerde baskın türler bulunsada 2 numaralı kümede olduğu gibi pop türüne yakın sayıda bulunan disco türü bu kümenin hangi müzik türüne ait olduğunun belirlenmesini zorlaştırmaktadır. Jazz, metal, blues diğer türlere göre daha dengeli bir dağılıma sahiptir. Bu o türlerin sınıflandırılmasının, diğerlerine göre daha zor olabileceğinin bir göstergesidir. Ancak bu vektörler kullanılarak yapılan sınıflandırma sonuçlarının düşük olması yapılan bu kümeleme işleminin de çok güvenilir sonuçlar vermeyebileceğine işaret etmektedir. Bu yüzden kümeleme işlemi, sınıflandırma bakımından daha yüksek başarı sağlayan özellik vektörleri ile yapılırsa daha sağlıklı sonuçlar alınacağı düşünülmektedir.

Kümelenmiş müzikler ile müzik öneri sistemi için bir test denemesi yapılmıştır. Bu testte her bir tür için 10 adet müzik, kümelemeden alınan türdeki müziğe olan uzaklıklarına göre seçilmiştir. Tablo 3.11 ile önerilen müziklerin hangi sınıfa ait olduğu incelenebilir. Tablodan görüleceği üzere yapılan öneriler farklı türlerden oluşmaktadır.

Tablo 3.11 Kümeleme Kullanılarak Yapılan Müzik Tavsiyesi

Music Genres	10 different Recommendations
blues	hiphop, hiphop, jazz, jazz, metal, metal, pop, blues, metal, metal
classical	metal, jazz, classical, classical, classical, metal, metal, pop, reggae, jazz
country	pop, hiphop, metal, metal, jazz, reggae, rock, pop, pop, country
disco	reggae, metal, metal, metal, disco, jazz, pop, metal, reggae, disco
hiphop	hiphop, reggae, blues, classical, disco, hiphop, metal, classical, rock, metal
jazz	blues, blues, jazz, metal, country, jazz, blues, country, rock, classical
metal	country, classical, hiphop, classical, blues, pop, metal, hiphop, metal, reggae
pop	hiphop, country, pop, classical, disco, classical, classical, classical, rock, pop
reggae	metal, reggae, blues, metal, metal, reggae, classical, jazz, rock, country
rock	blues, jazz, rock, pop, blues, pop, pop, pop, rock, pop

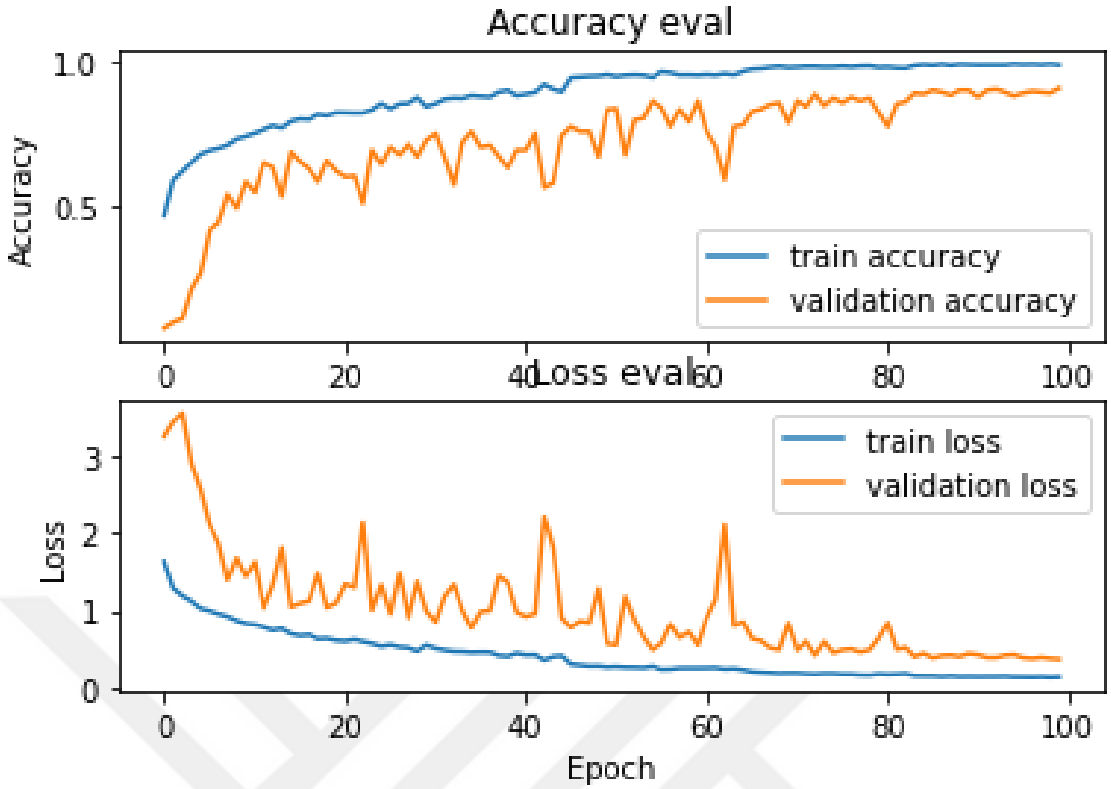


Şekil 3.7 One Hot Encoding Örneği [51]

3.8 1 Boyutlu CNN Sonuçları

Oluşturulan 1 boyutlu CNN modeli 100 epoch boyunca eğitilmiştir. Eğitim için 5'er saniyeye ayrılmış ham ses dosyaları kullanılmıştır. Bu ses dosyalarının kategorileri one hot encoding kullanılarak kodlanmıştır. Bu sayede model, son katmanında 10 adet seçenek arasından olasılıksal olarak en yüksek hangisini görüyorsa onu seçer ve sınıflandırma tamamlanır. İşlem sonucunda tür ile ilgili olan bit 1 diğerleri 0 olacak şekilde atama yapılır. One hot encoding kısaca kategorik değişkenlerin sınıf sayısı kadar bit sayısı kullanarak kodlama yapılması olarak ifade edilebilir [52]. Şekil 3.7'de basit bir one hot encoding örneği verilmiştir.

Veri setindeki örneklerin yaklaşık %85'i (5121) eğitim için kullanılırken, %5'i (300) doğrulama ve %10'u (570) da test verisi olarak kullanılmıştır. Eğitim boyunca eğitim oranı (learning rate) azaltılması metodu uygulanmıştır. Bunun için doğrulama (validation) verisinden elde edilen başarı, her epoch bir önceki epoch ile karşılaştırılmış ve eğer 10 epoch boyunca doğruluk oranında bir gelişme yaşanmazsa öğrenme oranı yarı yarıya azaltılmıştır. Yapılan eğitim boyunca elde edilen başarım ve kayıp, bir grafik üzerinde Şekil 3.8'de paylaşılmıştır. Eğitim tamamlandıktan sonra test için ayrılan veri seti modele verilerek modelden sınıflandırma sonuçları alınmış ve



Şekil 3.8 1 Boyutlu CNN Eğitim Boyunca Elde Edilen Başarı ve Kayıp Grafiği

gerçek kategorileri ile karşılaştırılmıştır. Buradan 0.34 kayıp oranı ve 0.91 doğruluk oranı elde edilmiştir.

4
SONUÇ VE ÖNERİLER

4.1 SONUÇ

Bu tez çalışması ile, müzik veri setlerinden zaman frekans analizi yöntemi ile özellik çıkarımı, derin öğrenme ve yapay zekâ ile özellik çıkarımı ve özellik çıkarımı yapılmadan elde edilen sınıflandırma sonuçları incelenmiş ve sonuçlar karşılaştırılmıştır. Zaman frekans analizi yöntemiyle özellik çıkarma bölümünde MFCC özelliklerinin diğer tüm özelliklere göre daha üstün performans gösterdiği ve müziğin karakteristik özelliklerin birçoğunu içinde barındırdığı sonucuna varılmıştır. Ancak tek başına kullanıldığında istenilen oranda başarı vermediği için diğer müzik özellikleri ile kullanılmasının daha iyi olacağı kanısına varılmıştır. Otomatik kodlayıcı ile yapılan özellik çıkarımında ise bu alanda önceki yapılan çalışmaların aksine sonuçlarda bir iyileşme görülememiştir. Bunun sebebi elde edilen MFCC özelliklerinin resim biçimine dönüştürülüp girdi formatına çevrilmesi ile oluşan veri kaybı ya da bozulması olabileceği gibi, otomatik kodlayıcının yaptığı sıkıştırma sonrası meydana gelen veri kaybı da olabilir. Derin öğrenme ile yapılan testte ise 1 boyutlu CNN'e ham ses dosyaları verilerek sınıflandırma yapılmıştır. 1 boyutlu CNN üç test arasında %91 ile en yüksek başarı oranına sahip yöntem olmuştur.

Yapılan kümeleme işlemlerinin sonuçları incelendiğinde kullanılan özelliklerin sınıflandırma başarısındaki düşüklüğü kümeleme işleminin de başarısız olmasına neden olmuştur. Bu düzensiz dağılım ile yapılan müzik öneri testi ise beklendiği üzere anlamlı sonuçlar vermemiştir.

Elde edilen sonuçlar sayısal işaret işleme ile elde edilen yöntemlerin müziğin özelliklerinin birçoğunu çıkarabildiği ve yeterli bir sınıflandırma başarısı yapabilmesine karşın otomatik kodlayıcı ile elde edilen özelliklerin sınıflandırma yapıldığında güzel sonuçlar vermediğini göstermiştir. Ancak kullanılan 1 boyutlu derin öğrenme mimarisi ile özellik çıkarımı yapılmaksızın elde edilen sınıflandırma başarısının diğer iki sonuçtan daha başarılı olması sayısal işaret işleme yöntemlerinin bazı ayırt edici özellik ya da özellikleri gözden kaçırdığının bir kanıtı olabilir.

4.2 ÖNERİLER

Gelecekteki çalışmalarda kullanılan bir boyutlu derin öğrenme ağının her bir adımı incelenerek, her adımda elde edilen özelliklerin neye karşılık olabileceği tartışılıp yeni bir özellik bulunup bulunamayacağı incelenebilir. Son katmandan bir önceki katmana özellik vektörü gibi davranarak o katmandan alınan veriler ile sınıflandırma ve kümeleme işlemi denenebilir.

Yapılan diğer çalışmalar incelendiğinde çok az bir kısım hariç genellikle aynı veri setinin kullanıldığı gözlemlenmiştir. Ancak bu veri setinin dezavantajlarının bulunması yapılan çalışmaların daha kapsamlı bir veri seti oluşturularak yapılması halinde daha başarılı olabileceği anlamına gelebilir. Bunun için gelecekte hem tür bakımından hem de şarkı sayısı ve çeşidi bakımından GTZAN veri setinden daha kapsamlı küresel bir müzik veri seti oluşturulup çalışmalar bu veri seti üzerinde tekrar edilebilir. Özellikle yapay zekâ ve derin öğrenme kullanılarak yapılan çalışmalarda, bu yöntemlerin verinin boyutu ve dengesi ile doğru orantılı olarak başarılarının artmasından dolayı, yeni bir veri seti ihtiyacı bu yöntemler için bir gereksinim durumuna gelmiştir.

Bu alanda yapılacak gelecek çalışmalarda yalnızca müzik ve türlerinden oluşan bir veri seti yerine aynı zamanda kullanıcı müzik dinleme geçmişinden oluşan bir veri seti de kullanılarak müzik öneri sistemleri daha da geliştirilebilir. Bunun için uygun ve düzenlenmiş bir veri seti bulunmaması belirtilen çalışmanın yapılabilmesi için önce bir veri seti oluşturma ihtiyacını doğurmaktadır. Bu kapsamda bu alanda çalışma yapacak araştırmacılar ilk adım olarak daha kapsamlı ve düzenli bir veri seti hazırlayarak daha sonra bu veri seti üzerinde çalışmalarını sürdürebilirler.

KAYNAKÇA

[1] J. S. Downie, "Music information retrieval," *Annual review of information science and technology*, vol. 37, no. 1, pp. 295–340, 2003.

[2] G. Tzanetakis, P. Cook, "Musical genre classification of audio signals," *IEEE Transactions on speech and audio processing*, vol. 10, no. 5, pp. 293–302, 2002.

[3] R. Basili, A. Serafini, A. Stellato, "Classification of musical genre: A machine learning approach.," in *ISMIR*, 2004.

[4] C. Xu, N. C. Maddage, X. Shao, F. Cao, Q. Tian, "Musical genre classification using support vector machines," in *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03).*, IEEE, vol. 5, 2003, pp. V–429.

[5] M. Haggblade, Y. Hong, K. Kao, "Music genre classification," *Department of Computer Science, Stanford University*, 2011.

[6] I. Panagakis, E. Benetos, C. Kotropoulos, "Music genre classification: A multilinear approach," in *ISMIR*, 2008, pp. 583–588.

[7] A. Elbir, H. O. İlhan, G. Serbes, N. Aydın, "Short time fourier transform based music genre classification," in *2018 Electric Electronics, Computer Science, Biomedical Engineerings' Meeting (EBBT)*, IEEE, 2018, pp. 1–4.

[8] A. Elbir, H. B. Çam, M. E. Iyican, B. Öztürk, N. Aydin, "Music genre classification and recommendation by using machine learning techniques," in *2018 Innovations in Intelligent Systems and Applications Conference (ASYU)*, IEEE, 2018, pp. 1–5.

[9] A. Elbir, N. Aydin, "Music genre classification and music recommendation by using deep learning," *Electronics Letters*, vol. 56, no. 12, pp. 627–629, 2020.

[10] T. Feng, "Deep learning for music genre classification," *private document*, 2014.

[11] I.-Y. Jeong, K. Lee, "Learning temporal features using a deep neural network and its application to music genre classification.," in *Ismir*, 2016, pp. 434–440.

[12] D. S. Lau, R. Ajoodha, "Music genre classification: A comparative study between deep learning and traditional machine learning approaches," in *Proceedings of Sixth International Congress on Information and Communication Technology*, Springer, 2022, pp. 239–247.

[13] T. Kim, J. Lee, J. Nam, "Sample-level cnn architectures for music auto-tagging using raw waveforms," in *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, IEEE, 2018, pp. 366–370.

[14] S. Allamy, A. L. Koerich, "1d cnn architectures for music genre classification," *arXiv preprint arXiv:2105.07302*, 2021.

[15] J. Bergstra, N. Casagrande, D. Erhan, D. Eck, B. Kégl, "Aggregate features and adaboost for music classification," *Machine learning*, vol. 65, no. 2, pp. 473–484, 2006.

[16] A. Karatana, O. Yildiz, "Music genre classification with machine learning techniques," in *2017 25th Signal Processing and Communications Applications Conference (SIU)*, IEEE, 2017, pp. 1–4.

[17] C. N. Silla, A. L. Koerich, C. A. Kaestner, "A machine learning approach to automatic music genre classification," *Journal of the Brazilian Computer Society*, vol. 14, no. 3, pp. 7–18, 2008.

[18] M. McKinney, J. Breebaart, "Features for audio and music classification," 2003.

[19] M. Kos, Z. Kačič, D. Vlaj, "Acoustic classification and segmentation using modified spectral roll-off and variance-based features," *Digital Signal Processing*, vol. 23, no. 2, pp. 659–674, 2013.

[20] N. Madhu, "Note on measures for spectral flatness," *Electronics letters*, vol. 45, no. 23, pp. 1195–1196, 2009.

[21] F. Zheng, G. Zhang, Z. Song, "Comparison of different implementations of mfcc," *Journal of Computer science and Technology*, vol. 16, no. 6, pp. 582–589, 2001.

[22] D. A. Ghosal, S. Saha, B. Dhara, R. Chakraborty, "Music classification based on mfcc variants and amplitude variation pattern: A hierarchical approach," *International Journal of Signal Processing, Image Processing and Pattern Recognition*, vol. 5, pp. 131–150, Mar. 2012.

[23] J. R. Higgins *et al.*, *Sampling theory in Fourier and signal analysis: foundations*. Oxford University Press on Demand, 1996.

[24] J. R. Partington, B. Ünalmış, "On the windowed fourier transform and wavelet transform of almost periodic functions," *Applied and Computational Harmonic Analysis*, vol. 10, no. 1, pp. 45–60, 2001.

[25] C. Harte, M. Sandler, M. Gasser, "Detecting harmonic change in musical audio," in *Proceedings of the 1st ACM workshop on Audio and music computing multimedia*, 2006, pp. 21–26.

[26] N. Aydin, S. Padayachee, H. S. Markus, "The use of the wavelet transform to describe embolic signals," *Ultrasound in medicine & biology*, vol. 25, no. 6, pp. 953–958, 1999.

[27] G. Li, J. Zhang, "Music personalized recommendation system based on improved knn algorithm," in *2018 IEEE 3rd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, IEEE, 2018, pp. 777–781.

[28] Z. Fu, G. Lu, K. M. Ting, D. Zhang, "Learning naive bayes classifiers for music classification and retrieval," in *2010 20th International Conference on Pattern Recognition*, IEEE, 2010, pp. 4589–4592.

[29] G. M. Bressan, B. C. de Azevedo, E. Lizzi, "A decision tree approach for the musical genres classification," *Applied Mathematics & Information Sciences*, vol. 11, no. 6, pp. 1703–1713, 2017.

[30] J. Han, J. Pei, M. Kamber, *Data mining: concepts and techniques*. Elsevier, 2011.

[31] I. H. Witten, E. Frank, "Data mining: Practical machine learning tools and techniques with java implementations," *Acm Sigmod Record*, vol. 31, no. 1, pp. 76–77, 2002.

[32] P. Cortez, "Data mining with neural networks and support vector machines using the r/rminer tool," in *Industrial conference on data mining*, Springer, 2010, pp. 572–583.

[33] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[34] F. Rosenblatt, "The perceptron: A probabilistic model for information storage and organization in the brain.," *Psychological review*, vol. 65, no. 6, p. 386, 1958.

[35] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural networks*, vol. 61, pp. 85–117, 2015.

[36] X. Chen, Zhang, Xue, "Deep convolutional neural network for mapping smallholder agriculture using high spatial resolution satellite image," *Sensors*, vol. 19, p. 2398, May 2019. DOI: 10.3390/s19102398.

[37] N. Srivastava, "Improving neural networks with dropout," *University of Toronto*, vol. 182, no. 566, p. 7, 2013.

[38] *What is a convolutional neural network?* [Online]. Available: https://www.mathworks.com/discovery/convolutional-neural-network-matlab.html.

[39] M. A. Kramer, "Nonlinear principal component analysis using autoassociative neural networks," *AIChE journal*, vol. 37, no. 2, pp. 233–243, 1991.

[40] I. Goodfellow, Y. Bengio, A. Courville, *Deep learning*. MIT press, 2016.

[41] D. E. Rumelhart, G. E. Hinton, R. J. Williams, "Learning internal representations by error propagation," California Univ San Diego La Jolla Inst for Cognitive Science, Tech. Rep., 1985.

[42] K. Siwek, S. Osowski, "Autoencoder versus pca in face recognition," in *2017 18th International Conference on Computational Problems of Electrical Engineering (CPEE)*, IEEE, 2017, pp. 1–4.

[43] L. Qiu, S. Li, Y. Sung, "3d-dcdae: Unsupervised music latent representations learning method based on a deep 3d convolutional denoising autoencoder for music genre classification," *Mathematics*, vol. 9, no. 18, p. 2274, 2021.

[44] M. Defferrard, "Structured auto-encoder with application to music genre recognition," Tech. Rep., 2015.

[45] M. Zhao, D. Wang, Z. Zhang, X. Zhang, "Music removal by convolutional denoising autoencoder in speech recognition," in *2015 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*, IEEE, 2015, pp. 338–341.

[46] A. Tikhonov, I. P. Yamshchikov, *et al.*, "Music generation with variational recurrent autoencoder supported by history," *arXiv preprint arXiv:1705.05458*, 2017.

[47] X. Guo *et al.*, *Fig. 1. the structure of proposed convolutional autoencoders (cae) for...* Nov. 2020. [Online]. Available: https://www.researchgate.net/figure/The-structure-of-proposed-Convolutional-AutoEncoders-CAE-for-MNIST-In-the-middle-there_fig1_320658590.

[48] B. L. Sturm, "The state of the art ten years after a state of the art: Future research in music information retrieval," *Journal of New Music Research*, vol. 43, no. 2, pp. 147–172, 2014.

[49] S. Gupta, J. Jaafar, W. W. Ahmad, A. Bansal, "Feature extraction using mfcc," *Signal & Image Processing: An International Journal*, vol. 4, no. 4, pp. 101–108, 2013.

[50] S. Bhanja, A. Das, "Impact of data normalization on deep neural network for time series forecasting," *arXiv preprint arXiv:1812.05519*, 2018.

[51] S. Verma, *Understanding 1d and 3d convolution neural network: Keras*, Oct. 2021. [Online]. Available: https://towardsdatascience.com/understanding-1d-and-3d-convolution-neural-network-keras-9d8f76e29610#:~:text=In%5C%201D%5C%20CNN%5C%2C%5C%20kernel%5C%20moves,kernel%5C%20moves%5C%20in%5C%202%5C%20directions..

[52] C. Seger, *An investigation of categorical variable encoding techniques in machine learning: Binary versus one-hot and feature hashing*, 2018.

A KAYNAK KODLAR

A.1 Otomatik Kodlayıcı Modeli Kaynak Kodu

```
# ENCODER
input_image = Input(shape=input_shape, name="encoder_input")
encoder = input_image
for i in range(len(conv_filters)):
  encoder = Conv2D(filters=conv_filters[i],
                   kernel_size=conv_kernels[i],
                   padding="same",
                   name=f"encoder_conv_layer_{i+1}")(encoder)
  encoder = ReLU(name=f"encoder_relu_{i+1}")(encoder)
  encoder = MaxPooling2D(pool_size=(2, 2))(encoder)
  encoder = Dropout(0.1)(encoder)


shape_before_latent_space = K.int_shape(encoder)[1:]  # Ignore batch


encoder = Flatten()(encoder)
encoder = Dense(latent_space_dim * 4, name="encoder_dense")(encoder)
encoder = Dropout(0.1)(encoder)

# latent space
encoder = Dense(latent_space_dim, name="encoder_output")(encoder)

encoder = Model(input_image, encoder, name="encoder")
encoder.summary()

# DECODER
decoder_input = Input(shape=latent_space_dim, name="decoder_input")
decoder = decoder_input

decoder = Dense(latent_space_dim * 4, name="decoder_dense_1")(decoder)
decoder = Dropout(0.1)(decoder)

num_neurons = np.prod(shape_before_latent_space)
```

```
decoder = Dense(num_neurons, name="decoder_dense_2")(decoder)

decoder = Reshape(shape_before_latent_space, name="reshpae")(decoder)

for i in reversed(range(1, len(conv_filters))):
  decoder = Conv2DTranspose(filters=conv_filters[i],
                            kernel_size=conv_kernels[i],
                            padding="same",
                            name=f"decoder_conv_transpose_layer_{len(conv_filters) - i}")(decoder)
  decoder = ReLU(name=f"decoder_relu_{len(conv_filters) - i}")(decoder)
  decoder = UpSampling2D(size=(2, 2))(decoder)
  decoder = Dropout(0.1)(decoder)

decoder = Conv2DTranspose(filters=1,
                          kernel_size=conv_kernels[0],
                          padding="same",
                          name=f"decoder_conv_transpose_layer_{len(conv_filters)}")(decoder)
decoder = UpSampling2D(size=(2, 2))(decoder)
decoder = Activation("sigmoid", name="sigmoid_layer")(decoder)

decoder = Model(decoder_input, decoder, name="decoder")
decoder.summary()

# autoencoder
autoencoder_input = input_image
autoencoder_output = decoder(encoder(input_image))

autoencoder = Model(autoencoder_input, autoencoder_output, name="autoencoder")
autoencoder.summary()

optimizer = Adam(learning_rate=0.0001)
mse_loss = MeanSquaredError()
autoencoder.compile(optimizer=optimizer, loss=mse_loss)
```

A.2 1 Boyutlu CNN Modeli Kaynak Kodu

```
model = keras.Sequential()

model.add(keras.layers.Conv1D(128, input_shape=[X_train.shape[1], 1], kernel_size=80, strides=4, padding='same',
  kernel_initializer='glorot_uniform',
  kernel_regularizer=keras.regularizers.l2(l=0.0001)))
```

```
model.add(keras.layers.BatchNormalization())
model.add(keras.layers.Activation('relu'))
model.add(keras.layers.MaxPooling1D(pool_size=4, strides=None))

model.add(keras.layers.Conv1D(128, kernel_size=3, strides=1,
    padding='same', kernel_initializer='glorot_uniform',
    kernel_regularizer=keras.regularizers.l2(l=0.0001)))
model.add(keras.layers.BatchNormalization())
model.add(keras.layers.Activation('relu'))
model.add(keras.layers.MaxPooling1D(pool_size=4, strides=None))

model.add(keras.layers.Conv1D(256, kernel_size=3, strides=1,
    padding='same', kernel_initializer='glorot_uniform',
    kernel_regularizer=keras.regularizers.l2(l=0.0001)))
model.add(keras.layers.BatchNormalization())
model.add(keras.layers.Activation('relu'))
model.add(keras.layers.MaxPooling1D(pool_size=4, strides=None))

model.add(keras.layers.Conv1D(512, kernel_size=3, strides=1,
    padding='same', kernel_initializer='glorot_uniform',
    kernel_regularizer=keras.regularizers.l2(l=0.0001)))
model.add(keras.layers.BatchNormalization())
model.add(keras.layers.Activation('relu'))
model.add(keras.layers.MaxPooling1D(pool_size=4, strides=None))

model.add(keras.layers.Lambda(lambda x : keras.backend.mean(x, axis=1)))
model.add(keras.layers.Dense(10, activation='softmax'))

model.compile(optimizer='adam', loss='categorical_crossentropy',
    metrics=['accuracy'])
model.summary()
```

TEZDEN ÜRETİLMİŞ YAYINLAR

Konferans Bildirisi

1. Y. Atahan, A. Elbir, A. Enes Keskin, O. Kiraz, B. Kirval and N. Aydin, "Music Genre Classification Using Acoustic Features and Autoencoders," 2021 Innovations in Intelligent Systems and Applications Conference (ASYU), 2021, pp. 1-5, doi: 10.1109/ASYU52992.2021.9598979.