BMBAD BJESR Mühendislik Bilimleri ve Araştırmaları Dergisi
Journal of Engineering Sciences and Researches

Konuşma İşaretlerinin Derin Evrişimsel Otokodlayıcı ve Artık Vektör Nicemleme Tabanlı Sıkıştırılması

Deep Convolutional Autoencoder and Residual Vector Quantization-Based Compression of Speech Signals

[1]Tahir BEKİRYAZICI, [2]Gürkan AYDEMİR, [3]Hakan GÜRKAN

[1,2,3]Bursa Teknik Üniversitesi, Mühendislik ve Doğa Bilimleri Fakültesi, Yıldırım / Bursa, Türkiye

[1]tahir.bekiryazici@btu.edu.tr, [2]gurkan.aydemir@btu.edu.tr,
[3]hakan.gurkan@btu.edu.tr

Araştırma Makalesi/Research Article

ARTICLE INFO

Article history

Received : 14 March 2024
Accepted : 15 April 2024

Keywords:

Speech Compression, Causal Convolutional Neural Network, Residual Vector Quantization, Deep Autoencoder

ABSTRACT

This paper proposes a compression method based on an original version of deep convolutional autoencoders, frequently used in the literature, and the compression rate is increased using residual vector quantization instead of constant step quantization. In the proposed method, the first encoder part of an autoencoder is utilized to map the input speech signal to a lower dimensional (code) space, and then the code is further compressed via residual vector quantization. The compression method offers different ratios due to two different decoder structures operating in parallel and the two codebooks. The method's performance is evaluated on the TIMIT dataset using the Perceptual Evaluation of Speech Quality metric. The proposed speech compression method achieved perceptual evaluation of speech quality scores of 1.665 and 1.985 for 1.25 and 2.5 kilobits per second transmission rates, respectively. The obtained compression rates are above the deep learning-based compression methods in the literature and at the same level as the traditional methods. At the same time, speech quality is better than the methods that provide the same compression levels.

MAKALE BİLGİSİ

Makale Tarihleri

Gönderim : 14 Mart 2024
Kabul : 15 Nisan 2024

Anahtar Kelimeler:

Konuşma Sıkıştırma, Nedensel Evrişimsel Sinir Ağları, Artık Vektör Nicemlemesi, Derin Otokodlayıcı

ÖZET

Bu çalışmada, konuşma işaretlerini sıkıştırmak için literatürde çokça kullanılan derin öğrenme tabanlı otokodlayıcı yapısının özgün bir örneği geliştirilmiş ve sabit basamaklı nicemleme yerine artık vektör nicemlemesi kullanılarak sıkıştırma oranı geliştirilmiştir. Önerilen sıkıştırma yönteminde, öncelikle giriş konuşma işaretini daha düşük boyutlu bir uzaya atayan otokodlayıcı kullanılmakta ve ardından otokodlayıcı çıkışı, artık vektör nicemlemesi ile daha da sıkıştırılmaktadır. Sıkıştırma yöntemi, birbirine paralel çalışan iki farklı kod çözücü yapısı ve iki kod kitapçığı sayesinde farklı sıkıştırma oranları sunmaktadır. Yöntemin başarımı konuşma kalitesini algısal değerlendirme metriği kullanılarak TIMIT veri kümesi ile değerlendirilmiştir. Önerilen konuşma sıkıştırma yöntemi, saniye başına 1,25 ve 2,5 kilo bit iletim hızları için sırasıyla 1,665 ve 1,985 konuşma kalitesini algısal değerlendirme skoru elde etmiştir. Elde edilen sıkıştırma oranları literatürde yer alan derin öğrenme tabanlı yöntemlerin üzerinde ve geleneksel yöntemlerle aynı seviyededir. Konuşma kalitesi ise benzer sıkıştırma oranı sunan yöntemlere göre daha iyidir.

ORCID ID: [1]0000-0002-0664-649X
[2]0000-0001-9213-576X
[3]0000-0002-7008-4778

1. GİRİŞ

Konuşma işaretleri insan vücudu içerisinde gerçekleşen bir dizi fizyolojik olay sonucunda meydana gelmektedir. Bu işaretlerin üretimi esnasında akciğer, ses telleri, gırtlak, dil, çene, dudak gibi organlar rol oynamaktadır. Ses tellerinin hava ile titreşmesi sonucu oluşan ses dalgaları, ağız ve burun gibi anatomik yapılar tarafından şekillendirilerek konuşma işaretlerine dönüşmektedir [1-2].

Konuşma, telekomünikasyon, multimedya uygulamaları, dijital ses yayıncılığı ve sanal asistanlar gibi çeşitli alanlarda yaygın bir şekilde kullanılmaktadır. Gelişen teknoloji ile günlük hayatta kullanılan ses ve konuşma verilerinin hacmi de giderek artmaktadır. Büyük miktarlardaki sıkıştırılmamış konuşma verisinin depolanması önemli miktarda hafıza gerektirmekte ve depolama maliyetlerinin artmasına neden olmaktadır. Bunların yanı sıra, sıkıştırılmamış konuşma işaretlerinin iletilmesi, yüksek veri hızları gerektirdiğinden ağ bant genişliğini zorlamaktadır. Yüksek veri hızları ayrıca ağ gecikmesine neden olarak iletişim sistemlerinin yanıt verebilirliğini etkilemekte ve böylece gerçek zamanlı etkileşimleri de bozmaktadır. Bu gibi önemli problemlerin üstesinden gelebilmek için konuşma işaretlerinin temel algısal kaliteyi bozmadan etkin bir şekilde sıkıştırılması gerekmektedir. Bu konuda geliştirilen konuşma sıkıştırma yöntemleri, konuşma işaretlerinin uygun maliyetli olarak daha küçük alanlarda depolanmasını sağlamakta ve iletim esnasında gereken veri hızını azaltmayı amaçlamaktadır.

En genel anlamda sıkıştırma yöntemleri kayıplı ve kayıpsız sıkıştırma olmak üzere iki başlık altında toplanmaktadır. Kayıpsız sıkıştırma yöntemlerinde bilgi kaybının sıfır olması hedeflendiği için sıkıştırma oranları çok küçük seviyelerde kalmakta ve bu yüzden kullanım alanları da çok dar olmaktadır. Kayıpsız sıkıştırma yöntemleri veri kaybının çok hassas olduğu uygulamalarda tercih edilmektedir. Bu yöntemlere örnek olarak aritmetik kodlama ve huffman kodlama örnek olarak verilebilir [3].

Kayıplı sıkıştırma yöntemlerinde ise sıkıştırılmak istenen ses işaretinde sıkıştırma oranına bağlı olarak veri kaybı meydana gelmektedir. Buradaki asıl amaç kaybedilecek olan verinin önemsiz olan kısımlardan oluşması ve düşük bilgi kaybı ile yüksek sıkıştırma sağlanmasıdır. Kayıplı sıkıştırma yöntemleri dalga biçimi kodlama (waveform coders), parametrik kodlama (parametric coders) ve hibrit kodlayıcılar olmak üzere kendi arasında üç gruba ayrılmaktadır. Dalga biçimi kodlayıcılar, konuşma işaretlerinin dalga biçiminin doğrudan temsiline dayanmaktadır. Parametrik kodlayıcı yöntemleri, konuşma işaretlerini daha az sayıda parametre ile temsil etmek üzerine odaklanmaktadır. Hibrit konuşma sıkıştırma yöntemlerinde ise dalga biçimi kodlayıcı ve parametrik kodlayıcıların özelliklerini birleştiren farklı bir yaklaşım kullanılmaktadır [4-5].

Doğrusal tahmine dayalı kodlama (Linear Predictive Coding-LPC), kod uyarımlı doğrusal tahmin (Code-Excited Linear Prediction-CELP) ve karışık uyarımlı doğrusal tahmin (Mixed Excitation Linear Prediction-MELP), konuşma sıkıştırmada üç önemli yöntemdir ve her biri telekomünikasyon sistemlerinin gelişiminde önemli bir rol oynamıştır [6-8]. LPC yöntemi, gelecekteki örnekleri tahmin etmek için geçmiş örnekleri kullanarak konuşma işaretlerini modelleyen temel bir yaklaşımdır. LPC-10 yöntemi saniye başına 2,4 kilo bit (kilobits per second-kbps) gibi çok düşük iletim hızlarında sıkıştırma yapmasına karşılık robotik bir ses üretmektedir. LPC'yi temel alan CELP yöntemi ise sentez yoluyla analiz yaklaşımı (analysis-by-synthesis) sunmakta ve orijinal ile sentezlenmiş konuşma arasındaki uyuşmazlığı en aza indirmek için vektör niceleme kullanmaktadır. Bu yöntem önceden tanımlanmış bir dizi kod arasından en uygun eşleşmeyi seçmektedir. MELP yöntemindeyse, uyarım sinyaline hem periyodik (perdeli) hem de periyodik olmayan (gürültü benzeri) bileşenleri dahil ederek hibrit bir yaklaşım sunmaktadır. Bu hibrit yaklaşım, özellikle gürültülü ortamlarda sentezlenen konuşmanın doğallığını ve anlaşılabilirliğini artırmaktadır. Bir başka konuşma sıkıştırma yöntemi olan SYMPES yönteminde ise konuşma işaretleri konuşmacıdan ve dilden bağımsız önceden tanımlı temel tanım ve zarf vektörleri ile bir kazanç katsayısı kullanılarak düşük bit oranlarında sıkıştırılmıştır. [9-10].

Son yıllarda görüntü işleme birimi (Graphics processing unit-GPU) teknolojilerinde artarak devam eden gelişmeler, derin öğrenme tabanlı konuşma sıkıştırma yöntemlerinin yeteneklerini önemli ölçüde artırmıştır. Bu sayede geliştirilen derin öğrenme tabanlı yöntemler karmaşık sinir ağı mimarilerini daha etkin bir şekilde kullanarak veri boyutunu azaltırken, sıkıştırılan verileri yüksek kaliteli bir şekilde yeniden elde edilebilmektedir. GPU'lar tarafından sağlanan hızlandırma, daha karmaşık modellerin eğitimini kolaylaştırarak sıkıştırma oranı ve ses kalitesi açısından üstün performans elde eden son teknoloji konuşma sıkıştırma yöntemlerinin geliştirilmesine olanak sağlamıştır. Literatürde derin öğrenme tabanlı ses sıkıştırma yöntemleri incelendiğinde, derin sinir ağları (DNN), evrişimsel sinir ağları (CNN), derin kalıntı ağları (deep residual networks), uzun-kısa süreli hafıza (LSTM), geçitli tekrarlayan ünite (gated recurrent unit-GRU) ve başlangıç (inception) mimarisi tabanlı modellerin kullanıldığı görülmektedir.

Derin öğrenme tabanlı konuşma sıkıştırma yöntemlerine temel olan konuşma şekli sentezi için üretken model olan WaveNET, ses üretirken bir sonraki ses örneğinin olasılık dağılımını tahmin ederek, önceki tüm zaman adımlarındaki örnekler üzerine koşullandırılmış bir yöntem kullanmaktadır [11].

Derin öğrenme tabanlı yapılan uçtan uca sıkıştırma yönteminde kalıntı (residual) sinir ağları ve otokodlayıcı yapıları (SC-DNN) beraber kullanılan model önerilmiştir [12]. Otokodlayıcı sıkıştırma mimarisi kodlayıcı ve kod çözücü yapılarında 4 farklı kalıntı bloğu içermektedir. Derin öğrenme modelinin eğitimi için tek bir kayıp fonksiyonu kullanılmasının yerine, farklı fonksiyon toplamlarından oluşan maliyet fonksiyonu (objective functions) kullanılmıştır.

Bir diğer çalışmada derin evrişimsel sinir ağları tabanlı mimari ile uçtan uca bir sıkıştırma yöntemi kullanılmıştır. Sunulan modelde, başlangıç ağ yapısı kullanılarak bir boyutlu katmanlardan oluşan kodlayıcı ve kod çözücü yapıları kullanılmıştır. Model girişine verilen farklı örnek sayılarına göre farklı miktarda sıkıştırma oranı sunmaktadır. Fakat modelin farklı girişler için tekrar eğitilmesi gerekmektedir [13].

Budanmış (Pruned) CELP konuşma sıkıştırma yöntemine dayalı geliştirilen yöntemde gürültüsüz otokodlayıcı yapısı kullanılarak başarım arttırılmaya çalışılmıştır [14]. Yöntemin kodlayıcı kısmında ilk m tane CELP katsayıları seçilerek budama işlemi gerçekleştirilmiş ve iletim için gerekli bit sayısı azaltılmıştır. CELP katsayılarından gelen gürültüyü gidermek için kod çözücü çıkışında gürültüsüz otokodlayıcı yapısı ile kalite artırımı hedeflenmiştir. Bu modelde her bir işaretin sıkıştırma oranı sabittir ve bunun yanı sıra düşük bit hızında sıkıştırma yapılamamaktadır.

Kalıntı ağlarına dayalı yapılan başka bir sıkıştırma yönteminde kaskad kalıntı ağ yapısı ile uçtan uca sıkıştırma modeli gerçekleştirilmiştir [15]. Modelde derin öğrenme yapısının haricinde sıkıştırma oranını arttırmak için Huffman kodlama yöntemi uygulanmıştır. Model çıkışında minimum 8,85 kbps olacak şekilde farklı değerlerde sıkıştırma değerleri elde edilmiştir. Fakat düşük bit hızları elde edilememiştir.

Yapılan başka bir çalışmada değişimsel otokodlayıcı (variational autoencoders-VAEs) ve tekrarlayan sinir ağlarına (Recurrent Neural Networks-RNN) dayalı uçtan uca geliştirilen sıkıştırma modeli önerilmiştir. Yapılan çalışmada elde edilen sonuçlar sinyal bozulma oranı (Signal to Distortion Ratio-SDR) metriği kullanılarak verilmiştir [16].

Gerçekleştirilen diğer bir çalışmada, derin sinir ağı tabanlı konuşma kodlayıcısının çok zaman ölçekli algısal kayıp (multi-time-scale perceptual loss) fonksiyonları kullanılarak Resnet tipi geçitli doğrusal birim (Resnet-type gated linear units-ResGLUs) yığınlarından oluşan model ile optimizasyonu yapılmıştır. Giriş çerçevesi alt çerçevelere bölünerek, nicemleme gürültüsü ve maskeleme eşikleri hem genel hem de yerel olarak hesaplanıp birleştirilmiştir. Resnet tipi birimlerle oluşturulan kodlayıcı, minimum 9 kbps seviyelerinde iletim hızını kaliteli bir şekilde sunmaktadır [17].

Derin öğrenmeye dayalı kaynağa duyarla konuşma kodlama modeli (SANAC) geliştirilerek yapılan çalışmada, gürültülü konuşma işaretleri kaynak ayırma ve sinir kodlama tekniklerinin karmaşık bir kombinasyonu yoluyla sıkıştırılmaktadır. Konuşma işaretinden arka plan gürültüsünü ayırma işlemi bir otokodlayıcı yapısı ile gerçekleştirilmiştir. Sıkıştırma işlemi sonrasında konuşma işareti ile arka plan gürültüsü ayrı ayrı kod çözücü katmanından geçirilerek çıkışta tekrar birleştirilmekte ve girişe verilen ses sinyali minimum hata ile oluşturulmaktadır. Bu yöntemin çıkışında elde edilen konuşma işaretlerinde, diğer modellerin aksine arka plan gürültüsü giderilmiş olmamaktadır. Gürültü bileşeni de ayrıca kodlandığından dolayı, model çıkışında yeniden elde edilmektedir [18].

Düşük sinyal-gürültü oranı (SNR) ortamlarında konuşma sıkıştırma ve iyileştirme amacıyla yapılan diğer çalışmada, görüntü sıkıştırmada başarılı olan OPINE-Net+ (OPtimization-INspired Explicable deep Network) ağını [19], konuşma işaretlerini sıkıştıracak şekilde uyarlanmıştır. Bu yöntem, konuşma dizilerinden bir örnekleme matrisinin uyarlanabilir öğrenimini, ardından bir başlatma ağı ve yeniden oluşturma ağı aracılığıyla yeniden yapılandırmayı içermektedir [20].

Yapılan bir diğer çalışmada, konuşma sıkıştırma için ölçeklenebilir ve verimli bir sinirsel dalga formu kodlama sistemi gerçekleştirilmiştir. Oluşturulmuş olan bu sistemde kodlayıcı ve kod çözücü kısımlarında evrişimsel sinir ağı temelli bir sinirsel dalga formu kodeki (neural waveform codec-NWC) kullanılmaktadır. Oluşturulan konuşma sıkıştırma sistemi 12-20-32 kbps iletim hızlarında sıkıştırma sonucu verebiliyorken, yüksek kbps iletim hızlarında daha iyi sonuç vermektedir [21].

Sinir ağlarına dayalı geliştirilen bir diğer konuşma sıkıştırma yönteminde, evrişimsel katmanlarla birlikte GRU (Gated Recurrent Units) yapısını içeren kodlayıcı ve GAN (Generative Adversarial Networks) tabanlı çok aşamalı kod çözücüden oluşan mimari yapı kullanılmıştır. Bu sıkıştırma yöntemi özellikle uzun süreli konuşma özniteliklerini etkili bir şekilde yakalama noktasında başarılıdır. Sıkıştırma yöntemi 8kbps iletim hızını desteklemektedir. Mimarinin kodlayıcı kısmı 9,8M, kod çözücü kısmı için 6,3M gibi çok yüksek miktarda parametre kullanılmıştır [22].

Derin öğrenme tabanlı konuşma sıkıştırma yöntemleri geleneksel yöntemlere göre algısal olarak daha kaliteli sıkıştırma sunmaktadır. Ancak elde edilen sıkıştırma oranları genellikle LPC-CELP gibi geleneksel yöntemlerin çok gerisinde kalmaktadır. Derin öğrenme tabanlı yöntemlerde konuşma işaretleri çerçevelere ayrılmakta, bu çerçeveler derin sinir ağları ile daha küçük boyutlu bir uzaya (kod bölgesine) aktarılmaktadır. Elde edilen kodlardaki her bir sayı sabit adımlarla nicemlenmektedir. Bu nedenle nicemleme dolayısıyla elde edilen sıkıştırma oranı kısıtlı kalmakta asıl sıkıştırma derin sinir ağı tarafından sağlanmaktadır. Bu çalışmada konuşma sıkıştırma için literatürde önerilen kaliteli sıkıştırma sağlayan evrişimsel otokodlayıcı yapılarından esinlenilerek nedensel evrişimsel katmanlar ve başlangıç katmanları içeren özgün bir hibrit derin otokodlayıcı mimarisi oluşturulmuştur. Dahası konuşmanın otokodlayıcının kodlayıcı kısmı ile sıkıştırılması ile elde edilen işaretler klasik nicemleme yerine artık vektör nicemlemesi (residual vector quantization-RVQ) ile sıkıştırılarak sıkıştırma oranı artırılmıştır. Bu sayede geleneksel yöntemlere göre daha kaliteli, derin öğrenme tabanlı yöntemlere göre ise daha yüksek sıkıştırma oranlı sıkıştırma elde edilmiştir.

Önerilen yöntemin yapısı gereği iki farklı oranda sıkıştırılmış işaret elde edilmektedir. Modelin kodlayıcı yapısında birbirine paralel çalışan evrişimsel sinir ağları ve başlangıç ağları modelleri bulunmaktadır. Kodlayıcı yapısı, girişe verilen işaretleri 4:1 oranında ve 8:1 oranında sıkıştıracak şekilde tasarlanmıştır. Bu yapısı sayesinde sıkıştırma miktarını arttırmak için kodlayıcı çıkışından elde edilen işaretlere artık vektör niceleme işlemi uygulanmakta ve

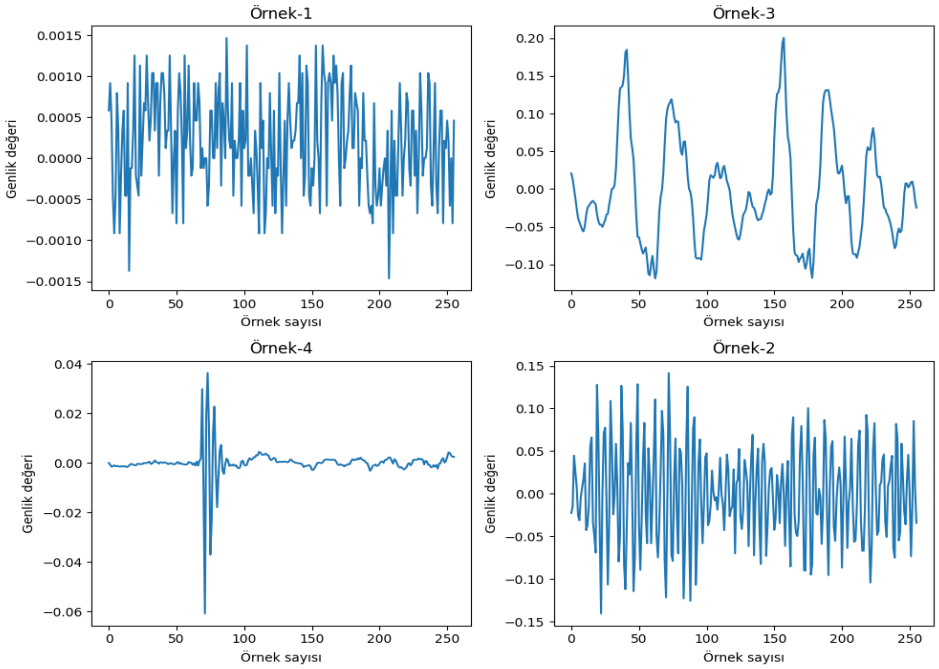
çok düşük bit iletim hızlarında kodlanmış işaretler elde edilmektedir. Önerilen sıkıştırma yöntemi tekrar eğitime ihtiyaç duymadan farklı hızlarda sıkıştırma yapabilmektedir. Yöntemin başarımını test etmek için PESQ metriği kullanılmış ve sıkıştırma oranı metriklerine göre farklı yöntemler ile karşılaştırmalı olarak sunulmuştur.

2. MATERYAL VE METOT

Çalışmanın bu bölümü, ön işleme adımları ve önerilen sıkıştırma modeli hakkında detayları içermektedir.

2.1. Ön İşleme Süreci

Derin öğrenme tabanlı modellerin eğitimi ve test aşamasında kullanılacak veriler için ön işleme adımları oldukça önemlidir. Önerilen konuşma sıkıştırma yönteminde kullanılacak olan veriler için ön işleme iki kısımdan oluşmaktadır. Ön işleme adımlarının ilk kısmında konuşma verileri -1/+1 arasına normalize edilirken sonraki adımda her bir konuşma işareti 256 örnek içerecek şekilde çerçevelere ayrılmaktadır. Şekil 1'de ön işleme adımlarından geçirilerek elde edilmiş olan çerçeve örnekleri rastgele seçilerek gösterilmiştir.



Şekil 1. Ön işleme adımından geçirilmiş örnekler.

2.2. Nedensel Evrişimsel Sinir Ağları ve Başlangıç (Inception) Ağları

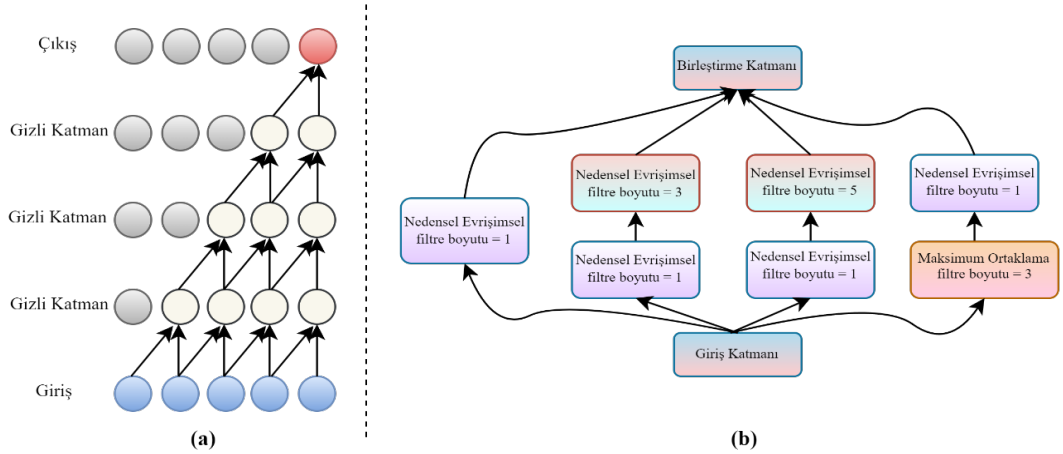
Evrişimsel sinir ağları (Convolutional Neural Network-CNN), görüntü sınıflandırma, nesne algılama ve görüntü bölütleme gibi görsel verileri içeren uygulamalar için yaygın olarak kullanılan derin sinir ağı katmanlarıdır. Evrişimsel katmanlar, etkili bir şekilde verideki uzamsal desenleri yakalayarak öznitelik haritaları çıkarmaktadır. Geleneksel CNN mimarilerinde, evrişimsel katmanlar giriş verileri üzerinde konvolüsyon işlemi yapmak için süzgeçler (kernel) kullanmaktadır. Burada gerçekleştirilen konvolüsyon işlemi sonrasında öznitelik haritaları oluşmaktadır. Havuzlama (pooling) katmanları ile elde edilen öznitelik haritalarını aşağı örnekleyerek önemli bilgiler korunmakta ve uzamsal boyutları azaltılmaktadır. Bu öznitelik çıkarma işlemi, CNN'lerin girdi verilerinin giderek daha soyut temsillerini öğrenmesini sağlamakta sınıflandırma, sıkıştırma ve tahmin görevleri için etkili rol oynamaktadır.

Zaman serisi verileri, her biri belirli bir zaman adımına karşılık gelen bir dizi veri noktasından oluşmaktadır. Geleneksel CNN'ler tüm girdi verisini aynı anda işlemekte ve gelecekteki bilgilere erişmelerini sağlamaktadır. Ancak zaman serisi verileri için bu yaklaşım yanlıştır. Bu haliyle model, gelecekteki verilere erişerek geleceği tahmin etmeye çalışabileceğinden istenmeyen sonuçlara yol açmaktadır. Bu problemi çözmek için nedensel evrişimsel katmanlar ortaya atılmıştır [11].

Nedensel CNN ağları zaman serisi verilerini işlemek için tasarlanmış özel bir CNN türüdür ve çalışma yapısı olarak normal CNN ağları ile aynı şekilde çalışmaktadır. Nedensel ifadesi, mevcut zaman adımında, yalnızca geçmiş zamandan gelen bilgilerin kullanıldığını ifade etmektedir. Bir başka deyişle bu ağlar, zaman serisi verilerini işlerken gelecekteki bilgiyi kullanmadan evrişim işlemini gerçekleştirmektedir.

Başlangıç (inception) mimarileri olarak da bilinen başlangıç sinir ağları ise görüntü sınıflandırma görevleri için tasarlanmış derin evrişimsel sinir ağlarıdır. Başlangıç ağlarının temel özelliği, aynı katman içinde farklı ölçeklerde çoklu paralel evrişimsel işlemler kullanmalarıdır.

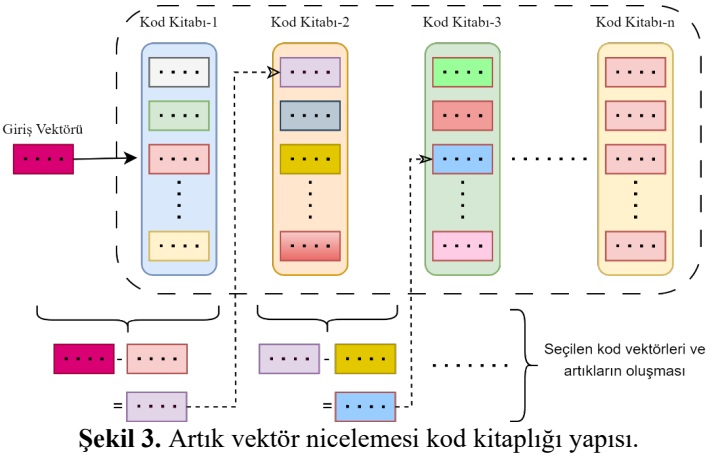
Başlangıç ağlarının temel yapı taşını oluşturan başlangıç modülü, çoklu paralel evrişimsel işlemlerden ve ardından birleştirme işleminden oluşmaktadır. Başlangıç ağları aynı katman içinde farklı süzgeç boyutlarına (1x1, 3x3 ve 5x5) sahip yapıların bir kombinasyonunu kullanmaktadır. Bu, ağın çeşitli ölçek ve çözünürlüklerdeki öznitelikleri aynı anda yakalamasını sağlamaktadır. Tek boyutlu başlangıç modelleri ise tek boyutlu veri yapılarını veya tensörlerini girdi olarak almakta ve bir boyutlu evrişimsel işlemlere tabi tutmaktadır. Sinir ağları bağlamında, tek boyutlu gösterimler genellikle zaman serisi verileri, konuşma işaretleri veya metin gibi sıralı verileri temsil etmek için kullanılmaktadır [23]. Şekil 2'de nedensel evrişim sinir ağı yapısı ve örnek bir boyutlu başlangıç mimarisi gösterilmektedir. Önerilen yöntem içerisinde kullanılan bağlangıç ağlarında da nedensel evrişimsel katmanlar kullanılmıştır.



Şekil 2. (a) Nedensel evrişimsel katman yapısı, (b) Başlangıç ağ yapısı.

2.3. Artık Vektör Nicemlemesi

Vektör nicemleme (vector quantization – VQ), kayıplı sıkıştırma yöntemlerinde kullanılan bir boyut azaltma yöntemidir. VQ, büyük bir veri noktası kümesini (vektörler), kod vektörleri (code vectors) veya merkezler (centroids) olarak bilinen daha küçük temsili vektör kümesiyle ifade etmek için kullanılmaktadır. Nicemleme süreci, verilerin kümelere ayrılması ve her bir özgün veri noktasının en yakın olduğu kümenin merkez noktası ile temsilini içermektedir. Tek katmanlı nicemleyici kullanmak, veriyi doğru bir şekilde temsil etmek için pratik bir çözüm olmamakla birlikte bu çözümde büyük bir kod kitabına ihtiyaç duyulmaktadır. Bu sorunu çözmek için artık vektör nicemleme (Residual vector quantization – RVQ) yöntemi kullanılmaktadır.
Geleneksel vektör niceleme yönteminin bir uzantısı olan artık vektör nicemleme yöntemi, kodlanmış verilerin sıkıştırma verimliliğini ve yeniden elde edilme doğruluğunu artırmayı amaçlamaktadır. Giriş vektörlerini doğrudan kod kitabındaki en yakın kod kelimelerine nicemleyen geleneksel nicemleme yönteminin aksine, RVQ giriş vektörleri ile en yakın kod vektörleri arasındaki artıkları (farkları) da kodlamaktadır. Bu yaklaşım, RVQ'nun daha ince ayrıntıları yakalamasını ve özellikle yüksek korelasyonlu veya yapılandırılmış veriler için daha iyi yeniden yapılandırma kalitesi elde edilmesini sağlamaktadır [24]. Şekil 3'te artık vektör nicemleme yapısı gösterilmektedir.



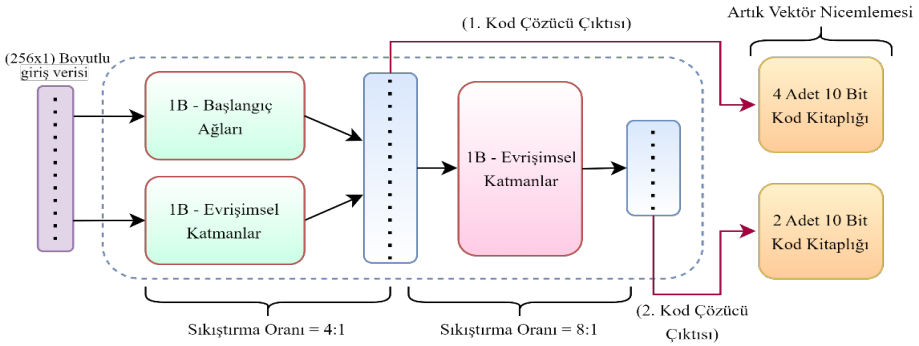
Şekil 3. Artık vektör nicelemesi kod kitaplığı yapısı.

Birincil kod kitabı, giriş vektörlerini en yakın kod vektörüne nicelemektedir. Girdi verisinin birincil kod kitabı içerisindeki en yakın temsili ile arasındaki fark ise ikinci kod kitabını oluşturmaktadır. Bu artık kod kitabı oluşturma işlemi belirlenen miktarda kod kitabı sayısına ulaşıncaya kadar devam etmektedir. RVQ'da her bir sonraki kod kitabı, artıklar üzerinde işlem yapmakta ve önceki kod kitapları tarafından tespit edilemeyen veri

yapısını daha da irdeleyerek oluşturmaktadır. Bu sayede her bir kod kitabı, veri kümesinin farklı yönlerini ve detaylarını yakalayabilen katmanlı bir yaklaşım oluşturmaktadır. Bu aşamalı artık kod kitabı yaklaşımı, RVQ'nun verimli sıkıştırmayı sürdürürken yüksek boyutlu vektörlerin giderek daha doğru yaklaşımlarını sağlamasına olanak tanımaktadır.

2.4. Derin Öğrenme Tabanlı Otokodlayıcı Modeli

Önerilen konuşma sıkıştırma yöntemi, derin öğrenme tabanlı bir boyutlu otokodlayıcı modeli ve artık vektör nicemleme yöntemlerinin birlikte kullanılmasıyla oluşturulmuştur. Otokodlayıcı yapısı, kodlayıcı ve kod çözücü olmak üzere iki kısımdan meydana gelmektedir. Model içerisinde tek bir kodlayıcı yapısı mevcutken, kod çözücü tarafında birbirine paralel çalışan iki farklı yapı mevcuttur. Modelin kodlayıcı yapısı içerisinde de hem bir boyutlu başlangıç ağlarından oluşan hem de bir boyutlu evrişimsel katmanlardan oluşan yapılar mevcuttur. Bu yapıların bir araya gelerek oluşturduğu kodlayıcı yapısı 2 farklı sıkıştırma (sıkıştırma oranları sırasıyla, 4:1 ve 8:1) oranı sunacak şekilde çıktılar verebilmektedir. Sıkıştırılmış verilerin elde edildiği katmanlarda süzgeç sayısı 1 olacak şekilde ayarlanmış ve bu şekilde bir boyutlu vektörler elde edilmiştir. Şekil 4'te önerilen yöntemin kodlayıcı yapısı gösterilmektedir.



Şekil 4. Önerilen konuşma sıkıştırma yönteminin kodlayıcı yapısı.

Başlangıç ağ modülleri, evrişimsel katmanların paraleline eklenerek, modelin birden fazla ölçekte ve çözünürlükte çeşitli özellikleri yakalayabilmesi ve karmaşık girdi verilerini etkili bir şekilde sıkıştırarak temsil edebilmesini sağlamıştır. Genel olarak, otokodlayıcı modelinde bir boyutlu başlangıç ağlarının ve evrişimsel katmanların kullanılması sıkıştırma kaybını en aza indirirken, girdi verilerinin doğal yapısının etkili bir şekilde yakalamasını sağlamıştır. Önerilen modelin kodlayıcı kısmına ait katman detayları Şekil 5'te gösterilmektedir.

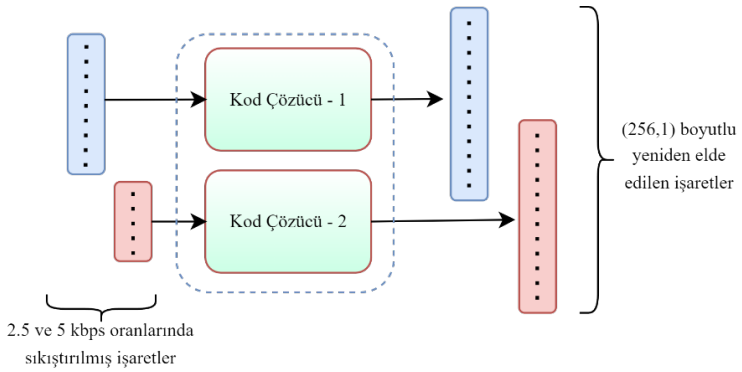
Katman Adı	Filtre Sayısı / Boyutu / Akt. Fonksiyonu
1B-Evrişimsel Adım Sayısı = 1	64 / 8 / tanh
1B-Evrişimsel + BN Adım Sayısı = 1	128 / 4 / tanh
1B-Evrişimsel + BN Adım Sayısı = 1	128 / 4 / tanh
1B-Evrişimsel Adım Sayısı = 2	128 / 8 / tanh
1B-Evrişimsel Adım Sayısı = 2	64 / 2 / tanh
1B-Evrişimsel Adım Sayısı = 1	64 / 1 / tanh

Katman Adı	Filtre Sayısı / Boyutu / Akt. Fonksiyonu
1B-Evrişimsel Adım Sayısı = 1	128 / 8 / tanh
1B-Başlangıç Ağı + BN	64-96-128-16-32-32 / tanh
1B-Evrişimsel Adım Sayısı = 2	64 / 2 / tanh
1B-Evrişimsel Adım Sayısı = 1	128 / 8 / tanh
1B-Başlangıç Ağı + BN	64-96-96-16-32-16 / tanh
1B-Evrişimsel Adım Sayısı = 2	64 / 6 / tanh

Katman Adı	Filtre Sayısı / Boyutu / Akt. Fonksiyonu	
Birleştirme Katmanı	-	
1B-Evrişimsel Adım Sayısı = 1	1 / 1 / tanh	1. Kodlayıcı Çıkışı
1B-Evrişimsel Adım Sayısı = 1	16 / 8 / tanh	
1B-Evrişimsel Adım Sayısı = 1	32 / 7 / tanh	
1B-Evrişimsel Adım Sayısı = 1	64 / 4 / tanh	
1B-Evrişimsel Adım Sayısı = 1	32 / 7 / tanh	
1B-Evrişimsel Adım Sayısı = 1	1 / 1 / tanh	2. Kodlayıcı Çıkışı

Şekil 5. Kodlayıcı yapısı katman detayları.

Konuşma sıkıştırma modelinin kod çözücü yapısında ise 2 farklı kod çözücü modeli bulunmaktadır. Bunun amacı kodlayıcı kısmından gelen 2 farklı sıkıştırma oranı ile sıkıştırılmış verileri tekrar özgün boyutuna getirilebilmektir. Kod çözücü yapılarında boyut artırma işlemlerinin yapıldığı katmanlarda genelde kullanılan üst örnekleme (upsampling) katmanlarının aksine adım sayısı (stride) 2 olan devrik evrişimsel katmanlar tercih edilmiştir. Önerilen yönteme ait kod çözücü yapısı Şekil 6'da gösterilmektedir.



Şekil 6. Önerilen modelin kod çözücü yapısı.

İlk kod çözücü (Kod Çözücü-1) yapısı 6 adet devrik evrişimsel katmandan oluşmaktadır. Bu katmanların 2 tanesinde adım sayısı 2 olarak ayarlanmış ve boyut artırma işlemi uygulanmıştır. Son katmanda aktivasyon fonksiyonu olarak lineer tercih edilirken diğer katmanlarda tanh kullanılmıştır. İkinci kod çözücü yapısında ise 9 adet devrik evrişimsel katman kullanılırken, boyut arttırma işlemi 3 katmanda yapılmıştır. Kod çözücü I ve II yapılarına ait detaylar sırasıyla Tablo 1 ve Tablo 2'de gösterilmektedir.

Tablo 1. Önerilen yöntemin kod çözücü-I yapısına ait detaylar

Katman Adı	Süzgeç Boyutu / Akt. Fonk.	Katman Çıktı Boyutu
1B-Devrik Evrişimsel Adım Sayısı = 1	8 / tanh	(1x64), 16
1B-Devrik Evrişimsel Adım Sayısı = 2	4 / tanh	(1x128), 64
1B-Devrik Evrişimsel Adım Sayısı = 1	8 / tanh	(1x128), 16
1B-Devrik Evrişimsel Adım Sayısı = 2	4 / tanh	(1x256), 16
1B-Devrik Evrişimsel Adım Sayısı = 1	8 / tanh	(1x256), 32
1B-Devrik Evrişimsel Adım Sayısı = 1	1 / lineer	(1x256), 1

Tablo 2. Önerilen yöntemin kod çözücü-II yapısına ait detaylar

Katman Adı	Süzgeç Boyutu / Akt. Fonk.	Katman Çıktı Boyutu
1B-Devrik Evrişimsel Adım Sayısı = 1	8 / tanh	(1x32), 32
1B-Devrik Evrişimsel Adım Sayısı = 1	7 / tanh	(1x32), 64
1B-Devrik Evrişimsel Adım Sayısı = 2	4 / tanh	(1x64), 32
1B-Devrik Evrişimsel Adım Sayısı = 1	7 / tanh	(1x64), 32
1B-Devrik Evrişimsel Adım Sayısı = 2	4 / tanh	(1x128), 64
1B-Devrik Evrişimsel Adım Sayısı = 1	8 / tanh	(1x128), 16
1B-Devrik Evrişimsel Adım Sayısı = 2	4 / tanh	(1x256), 16
1B-Devrik Evrişimsel Adım Sayısı = 1	8 / tanh	(1x256), 32
1B-Devrik Evrişimsel Adım Sayısı = 1	1 / lineer	(1x256), 1

Önerilen konuşma sıkıştırma yönteminin eğitimi 350 eğitim döngüsü (epochs) boyunca gerçekleştirilmiştir. Eğitim paket boyutu (1024) olarak belirlenmiştir. Eğitim esnasında aşırı uymanın önüne geçmek için erken durdurma (early stopping) yöntemi kullanılmıştır. Kayıp fonksiyonu olarak fonksiyonu olarak ortalama karesel hata (MSE) tercih edilirken, en iyileme için Adam yöntemi kullanılmıştır [25]. Konuşma sıkıştırma yönteminin eğitilebilir toplam parametre sayısı 649 524'tür.
Özet olarak önerilen modelin girişine verilen 256 boyutlu çerçeveler kodlayıcı kısmında 2 farklı oranda (4:1 ve 8:1) sıkıştırılmaktadır. Sıkıştırılan işaretler RVQ ile nicemlenerek 2,5 ve 1,25 kbps hızlarında iletilmektedir. 4:1 oranında sıkıştırılmış işaret için 4 tane 10 bitlik kod kitaplığı kullanılırken, 8:1 oranında sıkıştırma için 2 tane 10 bitlik kod kitaplığı tercih edilmiştir. Farklı oranlarda sıkıştırılıp iletilen işaretler kod çözücü kısmında sıkıştırma oranlarına göre 2 farklı modelden geçirilerek üst örnekleme işlemi yapılmakta ve kod çözücü çıkışında 256 boyutlu yeniden oluşturulan işaretler elde edilmektedir.

3. BENZETİM SONUÇLARI

3.1. TIMIT Veri Kümesi

Bu çalışmada önerilen konuşma sıkıştırma yönteminin eğitim ve test aşamaları için TIMIT veri kümesi kullanılmıştır. TIMIT veri kümesi, 438 erkek ve 192 kadın olmak üzere toplam 630 konuşmacının kayıtlarını içermektedir. Veri kümesi, 630 konuşmacının her biri tarafından Amerikan İngilizcesinin sekiz ana lehçesini temsil eden on cümlelik kayıtlarından oluşmaktadır. Veri kümesi içerisinde toplam 6300 veri örneği bulunmaktadır [26]. TIMIT veri kümesinin ayırt edici bir özelliği temiz, yüksek kaliteli kayıtlar içermesidir. Barındırdığı konuşma işaretleri ağırlıklı olarak arka plan gürültüsünden arındırılmıştır. Veri kümesindeki her konuşma kaydı 16-bit, 16 kHz dalga biçimlerinden oluşmaktadır. Bu veri kümesi içerisindeki ham konuşma kayıtları için gereken iletim bit hızı 256 kbps olarak hesaplanmaktadır.
Eğitim ve test aşamalarında, her bir konuşma kaydı [-1, +1] aralığına normalize edilmiş ve 256 örnekten oluşan çerçevelere bölünmüştür. Yeni oluşturulan veri kümesi daha sonra %10'u test için, %80'i eğitim için ve kalan %10'u da eğitim sürecinde doğrulama için kullanılmak üzere alt kümelere ayrılmıştır. Bu bölütleme, sıkıştırma yönteminin çok çeşitli konuşma örnekleri üzerinde eğitilmesini ve görülmeyen verilere etkili bir şekilde genelleştirilebilmesini sağlamıştır.

3.2. Başarım Değerlendirme Metrikleri

Sıkıştırma yöntemlerinin başarımları, çeşitli değerlendirme metrikleri kullanılarak değerlendirilebilmektedir. Başarım metrikleri arasında en yaygın olarak ortalama görüş skoru (MOS- Mean Opinion Score) kullanılmaktadır. MOS yöntemi, konuşma sıkıştırma algoritmalarının etkinliğini değerlendirirken insan algısına dayalı öznel bir bakış açısı sağlamaktadır [27].
Öznel değerlendirme metriği olan algısal görüş skorunun haricinde son yapılan çalışmalarda konuşma kalitesinin belirlenmesi için algısal değerlendirme metriği (Perceptual Evaluation of Speech Quality-PESQ) kullanılmaya başlanmıştır [28]. Bu yöntem, önceden eğitilmiş dinleyiciler kullanılarak gerçek dinleme testlerinin bilgisayar ortamında değerlendirilmesini sağlamaktadır. Bu değerlendirme metriği, farklı sıkıştırma tekniklerinin performansını insan algısına dayalı olarak değerlendirir. PESQ skorları -0,5 (çok kötü) ile 4,5 (çok iyi) arasında değişmektedir.
Sıkıştırma oranı (Compression rate - CR) ise özgün konuşma verisinin boyutu ile sıkıştırılmış konuşma verisinin boyutu arasındaki oranı ifade etmektedir. Bu metrik, sıkıştırma yöntemleri tarafından elde edilen sıkıştırma derecesini ölçmektedir. Bunun yanı sıra bps (bit per second), genellikle ses ve konuşma dosyalarının iletilmesi sırasında bir veri iletim hızı birimi (bit per second-bps) olarak kullanılmaktadır. Konuşma sıkıştırma uygulamalarında bps, bir saniyede işlenen veya iletilen bit miktarını ifade etmektedir. Matematiksel olarak, sıkıştırma miktarını ve iletim hızını gösteren ifadeler denklem (1) ve (2)'de sırasıyla gösterilmektedir.

$$CR = \frac{b_{özg}}{b_{ydo}} \tag{1}$$

$$bps = \frac{F_s * Q * F}{T} \tag{2}$$

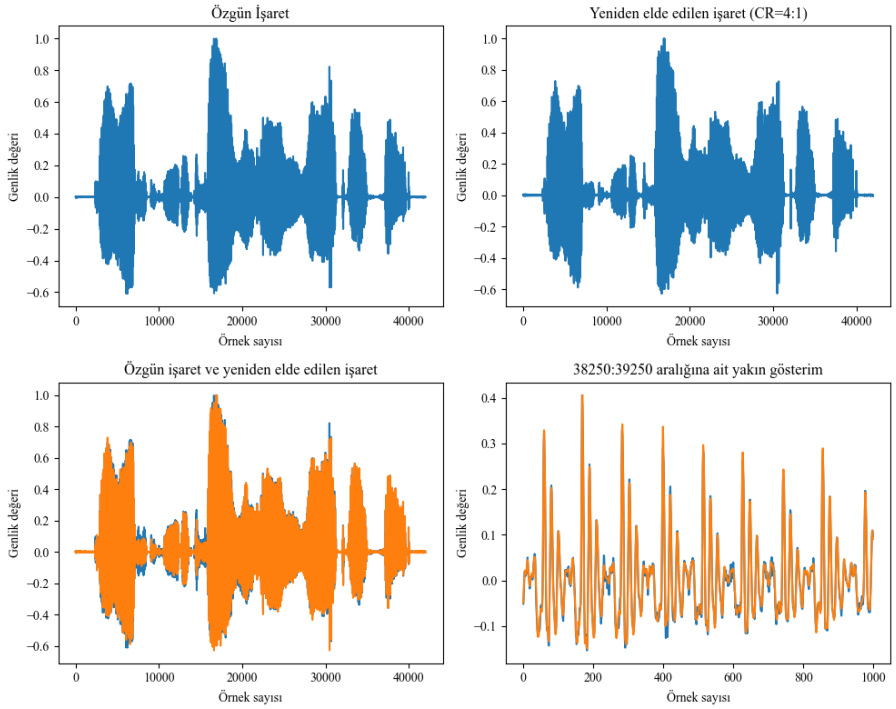
Denklem 1'de $b_{özg}$ ve b_{ydo} ile sırasıyla özgün işaretin ve yeniden elde edilen işaretin bit sayısını göstermektedir. Denklem 2'de ise bps, F_s, Q ve T ifadeleri ile sırasıyla iletim hızı, örnekleme frekansı, nicemleme değeri ve süre belirtilmektedir.

3.3. Başarım Sonuçlarının Değerlendirilmesi

Önerilen yöntemin çalışma zamanını değerlendirmek üzere Intel i9 işlemci (2.30 GHz), NVIDIA T2000 GPU ve 64 GB RAM bileşenlerden oluşan dizüstü bilgisayar kullanılmıştır. Elde edilen sonuçlara göre 2,5 kbps bit hızı sıkıştırma için gereken hesaplama süresi 3,72 milisaniye iken 1,25 kbps için gereken süre 2,94 milisaniyedir. Önerilen yöntemin başarımı PESQ, CR ve bps metrikleri ile iki farklı durum için incelenmiştir.

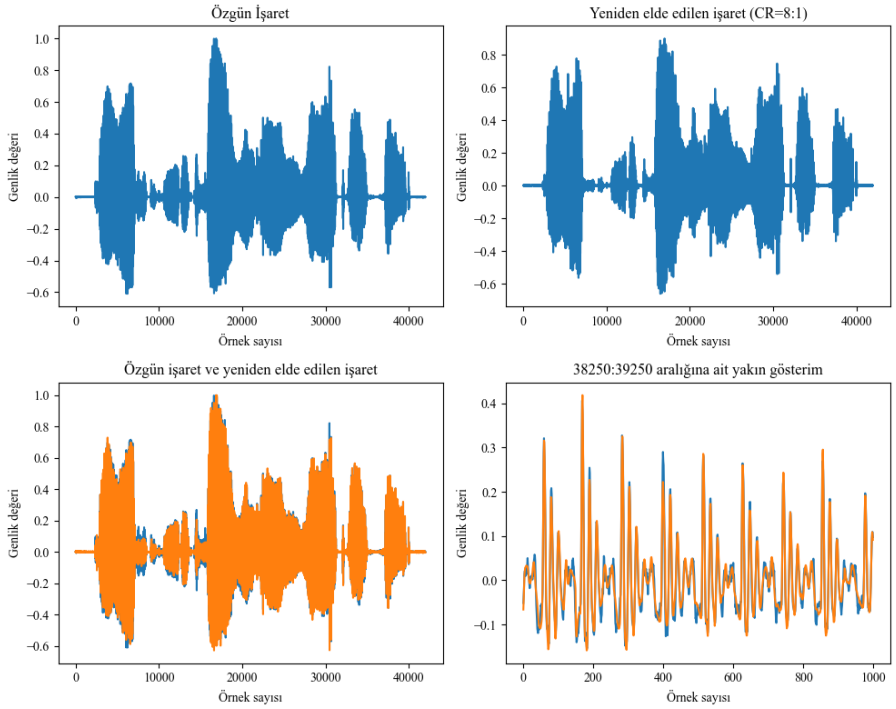
3.4. RVQ Olmadan Derin Öğrenme Tabanlı Sıkıştırma

Konuşma sıkıştırma yönteminin sadece derin öğrenme tabanlı kodlayıcı ve kod çözücü yapısı kullanılarak (RVQ kullanılmadan) eğitilmesi ve test edilmesi sonucu 4:1 oranında sıkıştırılıp yeniden elde edilen işarete ait grafikler Şekil 7'de gösterilmektedir. Şekilde gösterilen işarete ait PESQ değeri 3,11 olarak elde edilmiştir.



Şekil 7. Derin otokodlayıcı yapısı ile 4:1 oranında sıkıştırılıp yeniden elde edilen işaret.

Şekil 8'de ise aynı işaretin 8:1 oranında sıkıştırılması ile elde edilen sonuçlar gösterilmektedir. Sıkıştırma oranının artması ile ses kalitesinde azalma meydana gelmiş ve PESQ değeri 2,95 olarak elde edilmiştir.



Şekil 8. Derin otokodlayıcı yapısı ile 4:1 oranında sıkıştırılıp yeniden elde edilen işaret.

Artık vektör nicemleme yapısı olmadan elde edilen sıkıştırma sonuçları incelendiğinde Tablo 2'de yer alan değeler elde edilmiştir.

Tablo 3. 4:1 ve 8:1 sıkıştırma oranına sahip derin otokodlayıcı modelinin PESQ sonuçları

##	Min. PESQ Değeri	Mak. PESQ Değeri	Ort. PESQ Değeri	Standard Sapma
CR = 4:1	2,336	3,897	3,134	0,272
CR = 8:1	1,726	3,384	2,570	0,274

Önerilen derin otokodlayıcı modeline ilişkin test sonuçları incelendiğinde model girişine verilen özgün konuşma işaretlerinin, minimum 1,726 ve maksimum 3,897 PESQ değerleri ile yeniden elde edildiği gözlemlenmiştir. Tablo 3'te verilen sonuçlar incelendiğinde önerilen derin otokodlayıcı modelinin CR = 4:1 için 3,314 ve CR = 8:1 için 2,57 ortalama PESQ değerleri sağladığı ve her iki sıkıştırma oranı için standart sapmanın yaklaşık 0,27 olduğu görülmektedir.

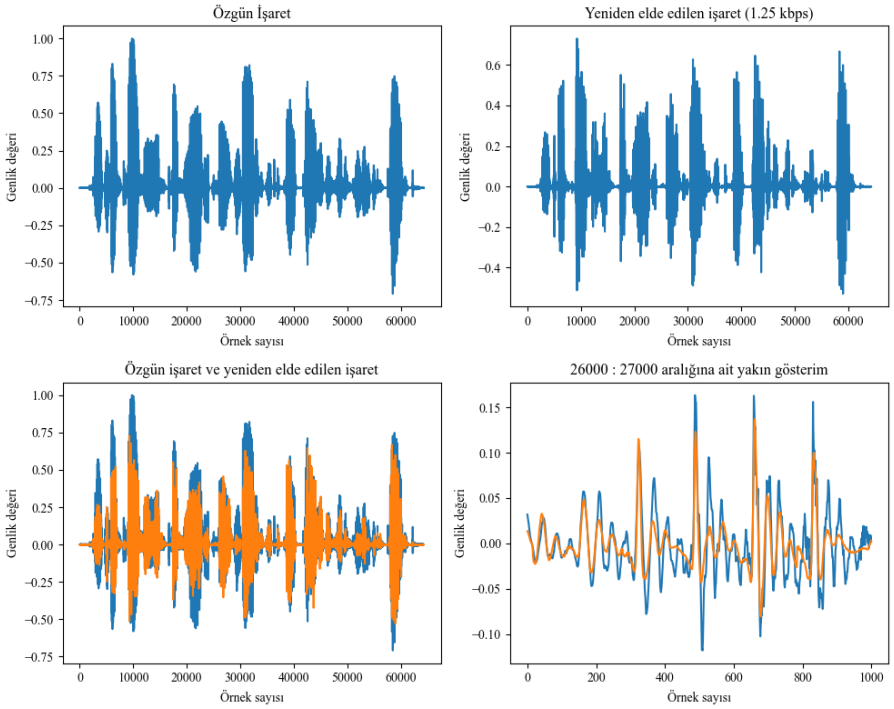
3.5. Derin Otokodlayıcı ve RVQ Tabanlı Sıkıştırma

Önerilen sıkıştırma modelinin sıkıştırma oranını arttırmak için kodlayıcı çıkışlarında değişken sayıda kod kitapçığı kullanılarak artık vektör nicemleme yöntemi kullanılmıştır. Kodlayıcı-1 çıkışından gelen (64:1) boyutlu sıkıştırılmış vektörler, 4 tane 10 bitlik kod kitapçığı ile nicemlenirken, kodlayıcı-2 çıkışından gelen (32:1) boyutlu sıkıştırılmış işaretler 2 tane 10 bitlik kod kitapçığı ile nicemlenmiştir. Bu sayede sıkıştırma oranı 4:1 olan işaretlerin iletim hızı 2,5 kbps olurken, 8:1 olan işaretler 1,25 kbps hızlarında iletilebilmişlerdir. Bu iletim hızlarına karşılık gelen PESQ değerleri Tablo 4'te verilmiştir.

Tablo 4. 1,25 ve 2,5 kbps iletim hızına sahip sıkıştırma modelinin PESQ sonuçları

##	Min. PESQ Değeri	Mak. PESQ Değeri	Ort. PESQ Değeri	Standard Sapma
1,25 kbps	1,273	2,307	1,665	0,184
2,5 kbps	1,382	2,724	1,985	0,206

1,25 kbps iletim hızı ile sıkıştırılmış 1,692 PESQ değerine sahip işarete ait konuşma işareti Şekil 9'da gösterilmektedir.



Şekil 9. 1.25 kbps hızında iletilen konuşma işareti.

Önerilen konuşma sıkıştırma yönteminin 630 TIMIT test verisi için farklı sıkıştırma oranlarında elde etmiş olduğu sonuçlar Tablo 5'te karşılaştırmalı olarak sunulmuştur. Elde edilen sonuçlara göre SC-DNN en iyi PESQ başarımını elde etmiştir, yöntem 9,02 bit hızını ancak yakalayabilmiştir. Makalede önerilen yöntem 2,5 kbps bit hızında kendisinden çok daha fazla bit hızı gerektiren P-CELP metoduna yakın bir PESQ değeri elde etmiştir. Ayrıca Deep-Vocoder ve SANAC gibi derin öğrenme tabanlı yöntemlere göre daha iyi konuşma kalitesinde yaklaşık üçte bir oranında bit hızında sıkıştırma gerçekleştirebilmiştir. Yazarların bilgisine göre literatürde önerilen yöntemin sağladığı bit hızına yaklaşabilen sadece LPC-10 yöntemi bulunmaktadır. Bu yöntem ise önerilen yöntemin çok altında PESQ skoru sağlamaktadır. Ayrıca 1,25 kbps gibi düşük bir bit hızına kabul edilebilir konuşma kalitesinde hiçbir yöntem ulaşamamıştır.

Tablo 5. Yöntemlerin sıkıştırma başarımları

Sıkıştırma Yöntemi	Veri Kümesi	Bit Hızı	Ortalama PESQ Değeri
SC-DNN [12]	TIMIT	9,02 kbps	**3,629**
SANAC [18]	TIMIT	9 kbps	1.66
P-CELP [14]	MHINT	7,5 kbps	2,18
Deep-Vocoder [13]	VoxForge	6,25 kbps	1,977
LPC-10 [6]	TIMIT	2,4 kbps	1,1316
Önerilen Yöntem	TIMIT	2,5 kbps	1,985
Önerilen Yöntem	TIMIT	**1,25 kbps**	1,665

4. SONUÇ

Bu çalışmada derin otokodlayıcı ve artık vektör nicemlemenin birlikte kullanıldığı konuşma sıkıştırma yöntemi önerilmiş ve kapsamlı deneylerle değerlendirilmiştir. Önerilen model, TIMIT veri seti kullanılarak farklı bit hızlarında test edilmiştir. Başarım sonuçları hem nicemleme işlemi olmadan sadece uçtan uca derin öğrenme tabanlı olacak şekilde incelenmiş, hem de artık vektör nicemleme ile sıkıştırma oranı artırılarak incelenmiştir. Önerilen yöntemin düşük bit hızlarında iletim yapabileceği ortaya konmuştur. Konuşma sıkıştırma yönteminde minimum sayıda eğitilebilir parametre kullanmaya çalışılarak hesaplama maliyeti düşürülmüştür. Ayrıca önerilen derin öğrenme tabanlı sıkıştırma yöntemi farklı oranlarda sıkıştırma miktarı sunarken bunun için tekrar eğitilme gerek duymayacak şekilde uçtan uca tasarlanmıştır. Bu sonuçlar, önerilen yöntemin gerçek dünya uygulamalarında kullanılma potansiyelini göstermektedir.

Elde edilen sonuçlara göre önerilen yöntem literatürde yer alan derin öğrenme tabanlı konuşma sıkıştırma çalışmalarının çok üzerinde bir bit hızı sunmaktadır. Ayrıca bunu yaparken elde edilen konuşma kalitesi de bu sıkıştırma yöntemleri ile yarışabilecek düzeydedir. Ancak çok daha düşük sıkıştırma oranı sunan yöntemlerin bazıları çok yüksek PESQ skorları sunmaktadır. Daha eski çalışmalarda yer alan ve sıkıştırma oranı önerilen yönteme yakın olan LPC-10 gibi yöntemler incelendiğinde ise önerilen yöntemin sağladığı konuşma kalitesinin daha yüksek olduğu gözlemlenmiştir. Özetle önerilen yöntem klasik konuşma sıkıştırma algoritmalarının sıkıştırma oranını yeni olan derin öğrenme modellerinin sıkıştırma kalitesiyle sağlayarak gelecekteki çalışmalara farklı bir bakış açısı sağlayacaktır. Ayrıca yazarların güncel bilgisine göre literatürde 1,25 kbps gibi düşük bir bit hızına kabul edilebilir konuşma kalitesinde hiçbir yöntem ulaşamamıştır.

Önerilen yöntemin sağladığı bit hızı ve konuşma kalitesi yöntemi konuşma depolama ve iletişim uygulamaları için uygun bir aday yapmaktadır. Ancak klasik konuşma sıkıştırma yöntemleri ile kıyaslandığında hesaplama süresi çok uzundur. Bu nedenle telsiz, cep telefonu gibi düşük hesaplama imkanlarına sahip cihazlarda algoritmanın gerçek zamanlı çalışması mümkün görünmemektedir. Önerilen yöntemin belirli bir işlemci gücüyle kullanılabilecek olması bu yöntemle ilgili en kısıtlayıcı özelliktir. Dahası önerilen yöntem otokodlayıcı ve RVQ tabanlı olduğu için konuşma işareti üzerine bu iki işlemden kaynaklı gürültü binmektedir. Bu da elde edilen konuşma kalitesi skorunun düşük olmasına sebebiyet vermektedir.

İleride yapılacak olan çalışmalarda literatürde daha düşük sıkıştırma oranına sahip derin öğrenme tabanlı yöntemlerin sağladığı yüksek PESQ skorlarına sıkıştırma oranı korunarak ulaşılmaya çalışılacaktır. Bunun için RVQ ve otokodlayıcı sebebiyle konuşma işareti üzerine binen gürültüyü giderecek yöntemler kullanılacaktır. Ayrıca önerilen yöntem konuşma işaretini bir zaman serisi gibi değerlendirmektedir. Halbuki LPC-10 gibi geleneksel yöntemler dalga formunun yakalanmasına dayanmakta ve dalgaların tam konumlarının çok önemli olmadığını göstermektedir. Yeni çalışmalarda birebir zaman serisini kodlamak yerine dalga formunu yakalayabilecek evrişimsel ağlar denenecektir. Böyle bir yöntem sıkıştırma oranını artırırken işaretin algısal kalitesini koruyacaktır.

Yazar Katkıları

Yazarlar eşit oranda katkı sağlamıştır.

Çıkar Çatışması

Makale yazarları aralarında herhangi bir çıkar çatışması olmadığını beyan ederler.

Destek ve Teşekkür Beyanı

Bu çalışma Bursa Teknik Üniversitesi Bilimsel Araştırma Projeleri birimi tarafından desteklenmiştir. Proje no: 230D005

KAYNAKÇA

[1] P.K. Mongia, and R.K. Sharma, "Estimation and statistical analysis of human voice parameters to investigate the influence of psychological stress and to determine the vocal tract transfer function of an individual," Journal of Computer Networks and Communications, vol. 2014, no. 17, pp. 1-17, 2014.

[2] T.F. Quatieri, “Discrete-time speech signal processing: principles and practice,” Pearson Education India, 2002.

[3] P. Warkade, and A. Mishra, “Lossless Speech Compression Techniques: A Literature Review,” International Journal of Innovative Research in Computer Science & Technology, vol. 3, pp. 25-32, 2015.

[4] T. Ogunfunmi, and M. Narasimha, “Principles of speech coding.” CRC Press, 2010.

[5] L. Rabiner, and R. Schafer, “Theory and applications of digital speech processing.” Prentice Hall Press, USA, 2010.

[6] D. O'Shaughnessy, “Linear predictive coding”, IEEE potentials, vol. 7, pp. 29-32, 1988

[7] M. Schroeder, and B. Atal, “Code-excited linear prediction (CELP): High-quality speech at very low bit rates”, IEEE International Conference on Acoustics, Speech, and Signal Processing. vol. 10, pp. 937-940, 1985.

[8] T. Unno, T.P. Barnwell, and K. Truong, “An improved mixed excitation linear prediction (MELP) coder,” IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings., vol. 1, pp. 245-248, 1999.

[9] Ü. Güz, H. Gürkan, and B.S. Yarman, “A new method to represent speech signals via predefined signature and envelope sequences,” EURASIP Journal on Advances in Signal Processing, vol. 2007, pp. 1–17, 2006.

[10] B.S. Yarman, Ü. Güz, and H. Gürkan, “On the comparative results of ‘sympes: A new method of speech modeling’,” AEU-International Journal of Electronics and Communications, vol. 60, no. 6, pp. 421–427, 2006.

[11] A. van den Oord et al., “WaveNet: A Generative Model for Raw Audio,” arXiv, Sep. 19, 2016.

[12] S. Kankanahalli, “End-to-end optimized speech coding with deep neural networks,” IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 2521-2525, 2018.

[13] H.Y. Keles, J. Rozhon, H.G. Ilk, and Voznak, M., “DeepVoCoder: A CNN model for compression and coding of narrow band speech,” IEEE Access, vol. 7, pp. 75081-75089, 2019.

[14] Y.T. Lo, S.S. Wang, Y. Tsao, and S.Y.A. Peng, “Pruned-CELP Speech Codec Using Denoising Autoencoder with Spectral Compensation for Quality and Intelligibility Enhancement,” IEEE International Conference on Artificial Intelligence Circuits and Systems, pp. 150-151, 2019.

[15] K. Zhen, J. Sung, M.S. Lee, S. Beack, and M. Kim, “Cascaded cross-module residual learning towards lightweight end-to-end speech coding,” arXiv:1906.07769, 2019.

[16] D.N. Rim, I. Jang, and H. Choi, "Deep neural networks and end-to-end learning for audio compression," arXiv:2105.11681, 2021.

[17] J. Byun, S. Shin, Y. Park, J. Sung, and S. Beack, “Optimization of deep neural network (DNN) speech coder using a multi time scale perceptual loss function,” in Proceedings of the Annual Conference of the International Speech Communication Association, pp. 4411–4415, 2022.

[18] H. Yang, K. Zhen, S. Beack, and M. Kim, “Source-aware neural speech coding for noisy speech compression,” IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 706-10, 2021.

[19] J. Zhang, C. Zhao, and W. Gao, “Optimization-inspired compact deep compressive sensing,” IEEE Journal of Selected Topics in Signal Processing, vol. 14, no. 4, pp. 765-774, 2020.

[20] M. Zhang, S. Liu, and Y. Wu, “Compression and Enhancement of Speech with Low SNR based on Deep Learning,” IEEE International Conference on Machine Learning, Big Data and Business Intelligence, pp. 242-248, 2022.

[21] K. Zhen, J. Sung, M. S. Lee, S. Beack, and M. Kim, “Scalable and efficient neural speech coding: A hybrid design,” IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 30, pp. 12-25, 2022.

[22] R. Lotfidereshgi, and P. Gournay, “Practical cognitive speech compression,” IEEE Data Science and Learning Workshop, pp. 1-6, 2022.

[23] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, and Rabinovich, “A. Going deeper with convolutions,” IEEE conference on computer vision and pattern recognition, pp. 1-9, 2015.

[24] N. Zeghidour, A. Luebs, A. Omran, J. Skoglund, and M. Tagliasacchi, “Soundstream: An end-to-end neural audio codec,” IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 30, pp. 495-507, 2021.

[25] D.P. Kingma, and J. Ba, “Adam: A method for stochastic optimization,”, arXiv:1412.6980, 2014

[26] J.S. Garofolo, “TIMIT acoustic-phonetic continuous speech corpus LDC93S1,” Linguistic Data Consortium, 1993. [Online]. Available: https://catalog.ldc.upenn.edu/LDC93S1

[27] R.F. Kubichek, “Standards and technology issues in objective voice quality assessment,” Digital Signal Processing, vol. 1, no. 2, pp. 38–44, 1991.

[28] A.W. Rix, J.G. Beerends, M.P. Hollier, and A.P. Hekstra, “Perceptual evaluation of speech quality (PESQ)-a new method for speech quality assessment of telephone networks and codecs,” IEEE international conference on acoustics, speech, and signal processing, vol. 2, pp. 749-752, 2001.