İZMİR KÂTİP ÇELEBİ ÜNİVERSİTESİ
2010
FEN BİLİMLERİ ENSTİTÜSÜ

Sesli Komut Algılama

Yazılım Mühendisliği Ana Bilim Dalı

Yüksek Lisans

Görkem Şen

No: Y210234039

Tez Danışmanı: Doç.Dr. Aytuğ Onan

Ocak 2022

Sesli Komut Algılama

Öz

Kişilerarası iletişim en yaygın olarak konuşma ile sağlanır. Ses, konuşmanın temel ve çok önemli bileşenidir. Akciğerlerden gelen havayı ses organlarıyla şekillendirerek kulak veya hassas aletler tarafından algılanabilen titreşimler dönüştürülmesi sesin en basit tanımı olarak nitelendirilebilir. Konuşma, boğaz ve ağızdaki bu titreşimlerin ve insan zihni tarafından belirli bir gramer altyapısında algılanabilecek karmaşık bir yapıdaki bir dönüşümdür. Yukarıdaki bilgiler ışığında, tezin amacı, insanların belirli komutları karakterize etmek için oluşturdukları ses verilerini analiz etmektir; bu analize uygun çıkacak özelliklere sahip yazılım oluşturmaktır. Ses ve konuşma tanıma, günümüz dünyasında çok popüler hale gelen bir teknoloji olma yolunda ilerlemektedir. Özellikle otomobil teknolojilerinde yüksek hızlarda sürerken, radyoyu sesinizle kontrol etmek güvenlik ve konfor açısından büyük bir gelişmedir. Proje kapsamında yapılan çalışmada, ses işleme teknolojisinin getirdiği bu konfor ve rahatlığı destekleyecek şekilde çalışma yapılması hedeflenmektedir. Derin öğrenme, insan beyninin çalışmasını taklit eden yapay bir zeka işlevidir. Veri işleme ve karar verme için kullanılabilecek modeller oluşturmak üzere tasarlanmıştır. Çok büyük bir sinir ağı ve büyük miktarda erişilebilir veri gerektirir.Makine öğrenimi daha basit kavramlar kullanırken derin öğrenme, insanların nasıl düşündüklerini ve öğrendiklerini taklit etmek için tasarlanmış yapay sinir ağlarıyla çalışır. Sinir ağları, tıpkı insan beyninin nöronlardan oluştuğu gibi katmanlardan oluşur. Ayrı katmanlardaki düğümler bitişik katmanlara bağlanır. Ağın sahip olduğu katman sayısından daha derin olduğu söylenir. İnsan beynindeki tek bir nöron, diğer nöronlardan binlerce sinyal alır. Yapay bir sinir ağında, sinyaller düğümler arasında seyahat eder ve ilgili ağırlıkları atar. Daha ağır bir düğümün bir sonraki düğüm katmanı üzerinde daha fazla etkisi olacaktır. Son katman, bir çıktı üretmek için ağırlıklı girdileri derler. Derin öğrenme sistemleri, büyük

miktarda veri işlendiğinden ve birkaç karmaşık matematiksel hesaplama içerdiğinden güçlü donanım gerektirir.Proje kapsamında linux üzerinde coqui aracı kullanılarak sesi metine doğru şekilde dönüştürme amaçlanmıştır. Coqui STT, bir ses akışını girdi olarak alır ve bu ses akışını belirlenen alfabedeki bir karakter dizisine dönüştürür. Bu dönüştürme iki temel adımla mümkün olur: İlk olarak ses, alfabedeki karakterler üzerinden bir olasılıklar dizisine dönüştürülür. İkincisi, bu olasılık dizisi bir karakter dizisine dönüştürülür.İlk adım bir Derin Sinir Ağı tarafından mümkün kılınır ve ikinci adım bir N-gram dil modeli tarafından mümkün kılınmaktadır . Sinir ağı, ses ve karşılık gelen metin transkriptleri üzerinde eğitilir ve N-gram dil modeli, bir metin külliyatı üzerinde eğitilir (bu, genellikle sesin metin transkriptlerinden farklıdır). Nöral model, konuşmadan metni tahmin etmek için eğitilir ve dil modeli, önceki metinden metni tahmin etmek için eğitilir. Çok yüksek bir seviyede, ilk kısmı (akustik model) fonetik bir kopyalayıcı olarak ve ikinci kısmı (dil modeli) bir yazım ve dilbilgisi denetleyicisi olarak düşünebiliriz.

Anahtar Kelimeler: makine öğrenimi,ses komutu,derin öğrenme,yapay sinir ağı

linux,coqui ,STT,N-gram.

Voice Command Detection

Abstract

Interpersonal communication is most commonly achieved through speech. Voice is the basic and very important component of speech. The simplest definition of sound can be described as transforming the air coming from the lungs into vibrations that can be perceived by the ear or sensitive instruments by shaping the sound organs. Speech is a transformation of these vibrations in the throat and mouth and a complex structure that can be perceived by the human mind in a certain grammatical infrastructure. In the light of the above information, the aim of the thesis is to analyze the voice data that people generate to characterize certain commands; is to create software with the features that will be suitable for this analysis. Voice and speech recognition is on its way to becoming a very popular technology in today's world. Controlling the radio with your voice is a huge improvement in safety and comfort, especially when driving at high speeds in automobile technologies. In the study carried out within the scope of the project, it is aimed to work in a way that will support this comfort and convenience brought by sound processing technology. Deep learning is an artificial intelligence function that mimics the work of the human brain. It is designed to create models that can be used for data processing and decision making. It requires a huge neural network and a large amount of accessible data. Machine learning uses simpler concepts, while deep learning works with artificial neural networks designed to mimic how people think and learn. Neural networks are made up of layers, just like the human brain is made up of neurons. Nodes in separate layers are connected to adjacent layers. The

network is said to be deeper than the number of layers it has. A single neuron in the human brain receives thousands of signals from other neurons. In an artificial neural network, signals travel between nodes and assign their respective weights. A heavier knot will have more impact on the next node layer. The last layer compiles the weighted inputs to produce an output. Deep learning systems require powerful hardware as large amounts of data are processed and involve a few complex mathematical calculations. Within the scope of the project, it is aimed to convert the voice to text correctly by using the coqui tool on linux. Coqui STT takes an audio stream as input and converts this audio stream into a string of characters in the specified alphabet. This conversion is made possible by two basic steps: First, the sound is converted into a sequence of possibilities over the characters in the alphabet. Second, this probability string is converted to a character string. The first step is made possible by a Deep Neural Network and the second step is made possible by an N-gram language model. The neural network is trained on audio and corresponding text transcripts, and the N-gram language model is trained on a corpus of text (this is often different from text transcripts of audio). The neural model is trained to predict text from speech and the language model is trained to predict text from previous text. At a very high level, we can think of the first part (acoustic model) as a phonetic replicator and the second part (language model) as a spelling and grammar checker.

Keywords: machine learning, voice command, deep learning, artificial neural network linux, coqui , STT, N-gram.

İçindekiler

Şekiller Listesi

Kısaltmalar Listesi

MFCC	Mel Frekansı Kepstrum Katsayıları
DFT	Ayrık Fourier Dönüşümü
FTT	Hızlı Fourier Dönüşümü
ASR	Otomatik Konuşma Tanıma
HZ	Frekans
DCT	Ayrık Kosinüs Dönüşümü
RNN	Tekrarlayan Sinir Ağı
GPU	Grafik İşleme Ünitesi
WER	Kelime Hata Oranı
LSTM	Uzun Kısa Süreli Bellek
NLP	Doğal Dil İşleme
GRU	Geçitlenmiş Özyinelemeli Birimler
STT	Konuşmayı Metne Çevirme
FIR	Sonlu Darbe Tepkisi
IIR	Sonsuz Darbe Tepkisi
MA	Hareketli Ortalama
O(N)	Big-O Notasyonu

Bölüm 1

Giriş

1 Konuşma Tanımada Temel Kavramlar

İnsanlar arasında geliştirilen sesli iletişim modelini anlamak ve temel kavramlara hakim olmak insan- makine arası iletişim modelini kurmak açısından büyük önem taşımaktadır. Bu kavramlar insan sesi, fonem, ses tanıma, ses analizi, ses kodlama ve ses sentezi olarak sıralanabilir.

İnsan Sesi: Bütün omurgalılar ağız, akciğerler ve ses tellerini kullanarak ses çıkarırlar. Sesin oluşması için önce akciğerlerden gelen hava soluk borusuna dolar ve buradan dışarı çıkar. Soluk borusunun üst bölümünde ise gırtlak yer alır. Gırtlakta ses telleri yer alır. Bu teller akciğerlerden gelen hava ile titreşir ve insan sesinin çıkmasını sağlar. Sesin kaynağı spektral düzlemde temel ve harmonik bileşenlerden oluşur ve gırtlaktan başlayan hareketli ve hareketsiz organların meydana getirdiği konuşma boşluğunun şekline göre ses tonlarını oluştururlar.

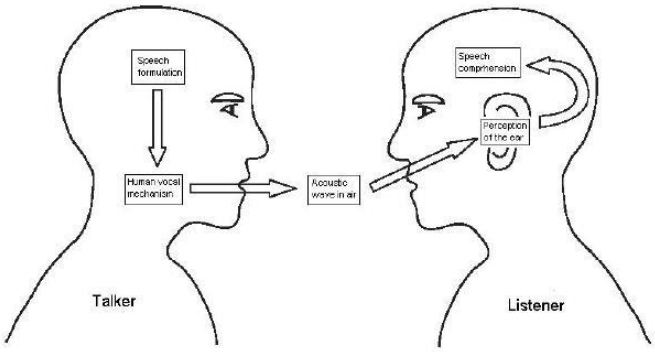
Fonem: Anlam içeren ve değişmesi ile dildeki bulunduğu bir kelimenin anlamını değiştiren en küçük ses birimine denir.

Ses Tanıma: Mikrofon veya benzeri bir ses kayıt cihazı tarafından alınmış akustik bir sinyalin kelime topluluğuna çevrim işlemi olarak nitelendirilebilir.

Ses Analizi: Sesin özelliklerinin çıkarılabilmesi için yapılan işlemlerdir.

Ses Kodlama: İşlem sırasında gözlenebilir bir sinyal kalitesi kaybı olmadan, sinyalin sayısal gösterimi olan bit oranın azaltmayı amaç edinen kodlama algoritmalarına denir.

Ses Sentezi: Metinden yapay olarak ses üretme işlemine verilen isimdir.



Şekil 1.1 : İnsanlar arası konuşma protokolü

1.1 Konuşma Tanıma Kullanım Alanları

Konuşma tanıma kullanım uygulamaları çok geniş bir çerçevede kullanılmaktadır. Kullanıldığı alanlarda hız, uygulama kolaylığı, bekleme süresini azaltma gibi katkılar sunmaktadır. Bu alanlar dikte-yazdırma, komut-kontrol, telefonla hizmet, tıbbi alanlar ve gömülü sistemlerdir.

Dikte – Yazdırma: Toplantı, mahkeme gibi organizasyonlarda konuşmaları algılayıp yazıya geçirme işlemini yapan uygulamalardır.

Komut- Kontrol: Bunlar, bir komut kümesi ve komuta karşılık gelen bir dizi işle tarafından oluşturulan uygulamalardır. Komutta mikrofon tarafından alınan ses ifadesi verileri algılanır ve algılanan komuta karşılık gelen işlev yürütülür.

Telefonla Hizmet: Özellikle banka ve devlet kuruluşları olmak üzere müşteri temsilcisine bağlanmadan yapılabilecek işlemler için geliştirilmiş teknolojidir.

Bu uygulamada sizi bir dijital asistan karşılar. Size işlem yapmak istediğiniz anahtar kelimeleri sorar ve sizin verdiğiniz cevap doğrultusunda sorununuzu çözmeye çalışır.

Tıbbi Alanlar: Tıbbi yetersizliklerden dolayı ellerini ve klasik veri girişi elamanlarından olan klavye, fare vb. operatörleri kullanmayan insanlara, konuşma tanıma vasıtasıyla veri girişi imkânı sunulmaktadır. Özel donanımlar ve kurulabilecek sistemler sayesinde, bu durumda olan bir kişi sesli komutlarla, örnek olarak televizyonu kontrol edebilir, müzik setini kumada edebilir vb. işlemleri kolaylıkla yapabilir.

1.2 Konuşma Tanıma Sistemlerinin Sınıflandırılması

Konuşma tanıma sistemlerini 4 alt başlıkta inceleyebiliriz.

a) Sesin Sürekliliğine Göre

b) Konuşmacıya Bağımlılığına Göre

c) Ses Tanıma Sisteminde Temel Alınan Birime Göre

d) Metin Bağımlılığına Göre

1.2.1 Sesin Sürekliliğine Göre

Sesin sürekliliğine göre 4 alt başlık altında incelenmektedir.

1.2.1.1 Ayrık Kelimeler

Bu tür konuşma tanıma sistemlerinde, sistem tarafından kelimeler arası kısa boşluklar beklenir.

1.2.1.2 Sürekli Konuşma

Konuşmacının en doğal haliyle konuşması beklenir. Dikte – yazdırma uygulamaları bu sınıflandırmaya girer.

1.2.1.3 Bağlı Kelimeler

Ayrık kelimelere çok benzer. Ayrık kelimelere göre kelimeler arası boşluklar daha kısadır. Konuşmacının daha kısa boşluklarla kelimeleri seslendirmesini destekleyen sistemler bu sınıfa girer.

1.2.1.4 Doğal Konuşma

Sürekli konuşma sınıflandırmasına çok benzer. Fakat insan doğasından kaynaklanan, doğal konuşma özelliklerini de algılayabilecek seviyede geliştirilmişlerdir. Bu tür konuşma özelliklerine, 'hımm', 'eee' vb. mırıldanmalar dahildir.

1.2.2 Konuşmacıya Bağımlılığına Göre

Bu sınıfta kişiye bağımlı ve kişiden bağımsız olarak 2 alt sınıf oluşmaktadır.

1.2.2.1 Kişiye Bağımlı

Bu tür ses tanıma sistemlerinde farklı bir konuşmacının sesi tanınmaz. Sadece sistemde kayıtlı kişiler için sistem tepki verir.

1.2.2.2 Kişiden Bağımsız

Farklı bir konuşmacı için şablonların güncellenmesi gerekmez. Sistem yaş, cinsiyet, aksan ayırmadan tepki verir. Kişiye bağımlı sistemlere göre çok daha zor bir geliştirme süreci vardır. Ses sinyalinin özelliği kişiden kişiye hatta aynı kişinin farklı zaman, duygu ve düşünce hallerinde bile farklılık gösterebilir.

1.2.3 Temel Alınan Birime Göre

Aşağıda yer alan alt başlıklarda detaylı anlatılmaktadır.

1.2.3.1 Sözcük Tabanlı

Bu sistemlerde en küçük birim sözcüklerdir. Oluşturulan referans şablonlar sözcüktür. Doğruluk dereceleri daha yüksektir, fakat bu sistem gereksinimini arttırır. Genellikle komut - kontol uygulamalarında kullanılırlar.

1.2.3.2 Fonem Tabanlı

Bu sistemlerde en küçük birim ise fonemlerdir. Fonem, bir dilde anlam ayıracı en küçük ses birimine denir. Bundan dolayı da şablon sayısı, oluşturulması, saklanması, işlenmesi çok az bir sistem gereksinimi ile karşılanır. Fonemlerin tabanlı yapıların kullanılmasında ki en büyük problem ise başlangıç ve bitişlerinin tespitinin zor olmasıdır. Ayrıca doğruluk oranı sözcük tabanlı sistemlere göre düşüktür. Olumlu özelliği ise hata durumlarında geri dönüşler yaparak düzeltme imkanı olmasıdır.

1.2.4 Metin Bağlılığına Göre

Aşağıda yer alan alt başlıklarda detaylı anlatılmaktadır.

1.2.4.1 Metne Dayalı Tanıma

Bu tür sistemlerde kullanılan test verisi, eğitim verisi ile sınırlı tutulur. Yani sistem, eğitimde kullanılan kelimelerin farklı seslendirilişleri ile test edilir.

1.2.4.2 Metinden Bağımsız Tanıma

Ağın eğitimde kullanılan kelimelerin yanı sıra bu kelimelerin eğitim dışı farklı kombinasyonlarına da cevap verebilir.

Örnek vermek gerekirse;

Sistem “sekiz” ve “yetmiş” kelimeleri ile eğitilmişse bu sistem “yetmişsekiz” kelimesini de tanır.

Bölüm 2

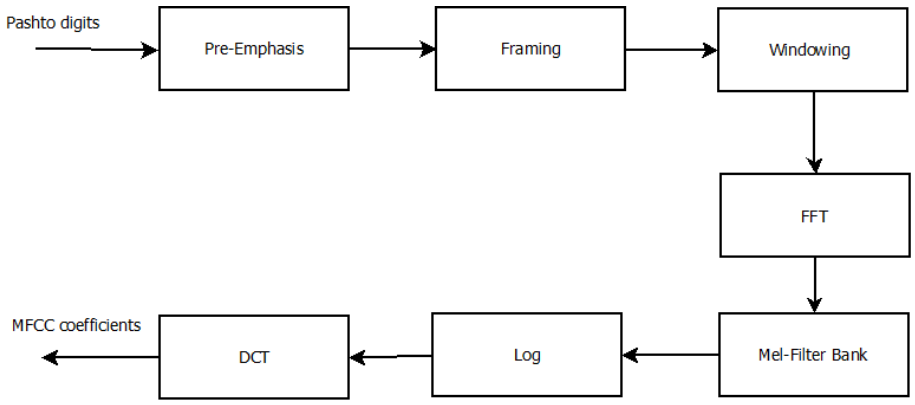
Sinyal İşleme

2 Sayısal Sinyallerin İşlenmesi

Sayısal sinyal sahip olduğu özelliklerin çıkarımı için bazı işlemlerden geçirilir. Özellik çıkarma, ilk ham veri kümesinin işlenmek üzere daha yönetilebilir gruplara indirgeneceği boyutsallık azaltma sürecine verilen addır. Bu büyük veri kümelerinin bir özelliği, işlemek için çok fazla bilgi işlem kaynağı gerektiren çok sayıda değişkendir. Özellik çıkarma, değişkenleri seçen veya özelliklerle birleştiren yöntemlerin genel adıdır ve işlenmesi gereken veri miktarını etkili bir şekilde azaltırken, orijinal veri kümesini doğru ve eksiksiz bir şekilde açıklar. Özetlemek gerekirse Elimizdeki sayısal datayı işimize yarayacak en optimum ve küçük boyutlara indirgemeye çalıştığımız kısımdır. Bu başlık altında ele alacağımız kısımlar ses sinyalinden akustik özellik çıkarımı üzerine olacaktır. Bu akustik öznitelikler, konuşma sinyaline ait özgün bilgiler taşımaktadır ve oluşturulacak olan akustik modelde giriş olarak kullanılacaktır. Çalışmada MFCC öznitelik çıkarma yöntemi kullanılacaktır.

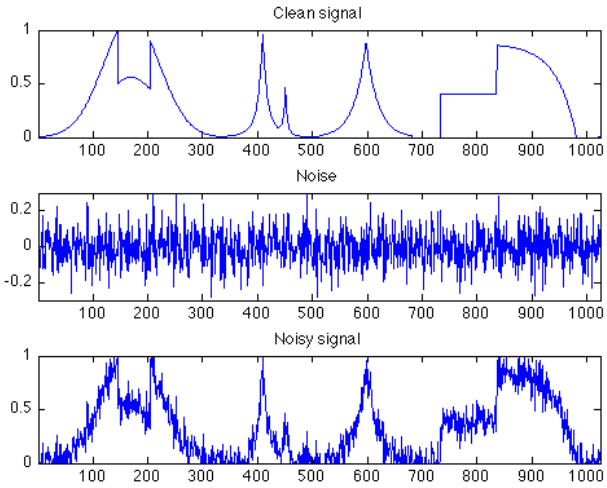
2.1 MFCC Öznitelik Çıkarma Yöntemi

MFCC (Mel Frequency Cepstral Coefficients) öznitelik çıkarma yöntemi konuşma tanıma sisteminin başarısında önemli bir etkendir. Öznitelik çıkarma adımları Şekil 2.1' de verilmiştir.



Şekil 2.1 MFCC işlem adımları

Önvurgu (Preemphasis): Sinyal işleme konusunda en önemli sorunlardan bir tanesi sinyali gürültüden temizlemektir. İletilmek istenen bilgi sinyali ile bu sinyalin üzerindeki gürültü bileşenlerinin genlikleri arasındaki orana sinyal gürültü oranı (S/N) denir. Ön vurgu adımındaki en büyük amaç bu bu oranı mümkün olduğu kadar yüksek tutmaktır. Bu kapsamda çözüm ise, yüksek frekanslı ancak düşük genlikli harmoniklerin, daha sonra kolay tanınıp temizlenmesi için abartılarak yükseltilmesidir. Önvurgu tekniği sinyal işlemede filtre kullanımı ile sağlanmaktadır. Filtre Tasarımı: Bir filtre zaman alanında ve frekans alanında olmak üzere 2 yol ile tasarlanabilir. IIR filtreler için genlik ve fazı frekans alanında tanımlanırken, FIR filtrelerin impulse cevabı da zaman alanında tanımlanarak tasarım yapılır. Şekil 2.2'de verilen yapıda gürültünün sinyalden ayrılması detaylı bir şekilde gösterilmiştir.



Şekil 2.2 Gürültünün temiz sinyalden ayrılması

IIR Filtre Tasarımı: Infinite impulse response filtreler yani sınırsız sinyal tepkisi yaratan filtreler, akım çıkışı örnek değerini elde etmek için akım giriş örneği değerini, geçmiş giriş ve çıkış örneklerini kullanır Bir başka deyişle, çıktı verileri özyineleme ışığında girdide tekrar kullanılarak sürekli, dinamik bir akış sağlanır. Sistemin son tepkisi girdiler yanı sınra önceki çıktılar ile belirlenir. IIR filtreleri ayrık olarak tasarlamanın 2 yolu vardır.

1- Filtre analog olarak tasarlanıp örnekleme ya da dönüşüm yöntemleriyle ayrıklaştırılır.

2- Filtre, doğrudan yaklaşım ya da eniyileme yöntemleriyle ayrık olarak tasarlanır. En çok kullanılan analog filtre tipleri Buttterworth, Chebyshev ve Elliptic filtrelerdir. Alçak geçiren genlik tanımlamarı en düşük dereceli olarak ilk Elliptic filtreyle sonra sırasıyla Chebyshev ve Butterworth filtre ile gerçeklenebilir. Ancak aralarında en düz genlik cevabına sahip filtre Butterwort filtredir. FIR Filtre Tasarımı: Finite impulse response yani sınırlı sinyal tepkisi yaratan filtreler, bir akım çıkış örneği değeri elde etmek için sadece mevcut ve geçmiş giriş dijital örneklerini kullanır. Geçmiş çıktı örneklerini kullanmaz. FIR filtre özellikleri aşağıda verildiği gibidir.

1- FIR filtre tasarımı ayrık bir işlemdir.

2- Analog bölgede eşdeğer bir FIR filtre karşılığı yoktur.

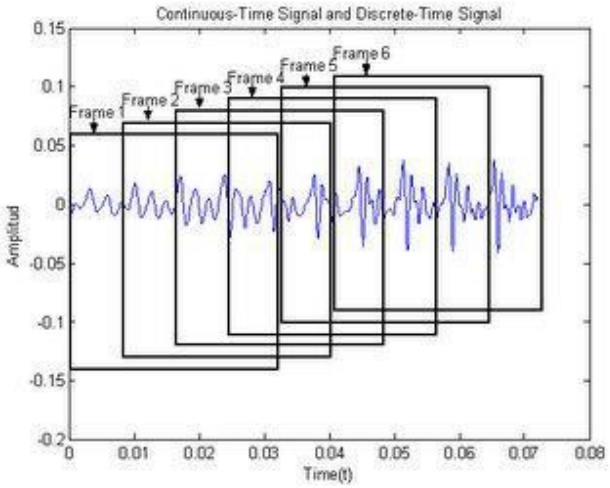
3- FIR filtre tanımlamaları daha çok zaman bölgede yapılır.

4- FIR filtrelerin iki avantajı kararlılıkları ve lineer fazlı olmalarıdır. Aynı özellikleri sağlayan IIR filtrelere kıyasla FIR filtrelerin katsayı sayısı (dereceleri) çok daha fazladır.

5- En basit FIR filtre Moving Average (MA) filtresidir

6- Genlik cevabı anlamında oldukça zayıf olan bir alçak geçiren filtredir.

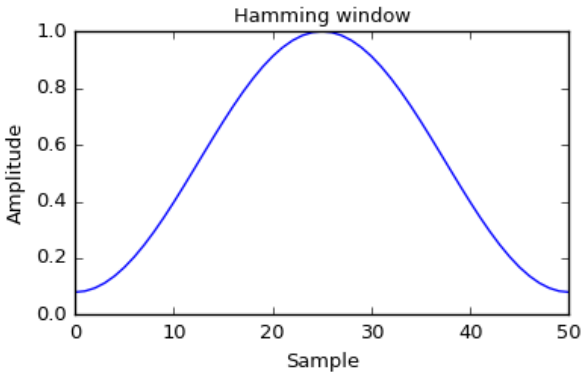
7- Band genişliği yalnızca filtrenin derecesi ile kontrol edilir. Çerçeveleme, konuşmanın durağan olmayan (istatistikleri zamanla değişen) yapısından dolayı ses sinyalini belirli çerçevelere (epoch) bölme ihtiyacı duyarız. Bu işlemde sinyalden elde ettiğimiz her bir çerçeveyi de alt pencerelere böleriz. Böylelikle her bir pencere için durağana yakın istatislikler elde ederiz. Uygulamamızda 20 ms'lik (50 adet çerçeve = 1 saniye) çerçeve örnekleri alınmıştır. Şekil 2.3'de bu yapı görülmektedir.



Şekil 2.3 Frame blocking (Çerçeveleme)

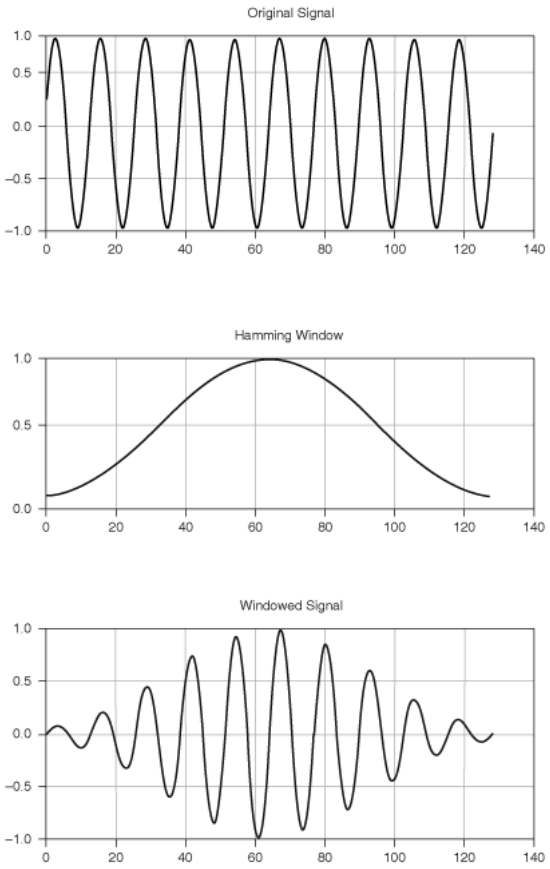
Pencereleme, pencereleme aşamasında, alınan sinyal içerisindeki devamsız kısımların dikkate alınmaması ses tanıma için kritik bir eşiktir. Bu işlem pencereleme ile gerçekleşmektedir. Pencereleme, elde edilen ses sinyalinde spektral analiz yapabilmemizi sağlar. En yaygın kullanılan pencereleme metodu ise Hamming

Penceresidir. Bunun sebebi ise, pencere yan loblarının sinyal örneklemesinin gürültü tabanının çok altında kalmasını gerektiren herhangi bir durum olmamasıdır. Bunu ise ideal dürtü tepkisinin düzgün bir şekilde kesilmesini ve daha iyi görünen bir frekans tepkisini sağlamak için giderek daha karmaşık kosinüs fonksiyonlarını kullanarak sağlamaktadır. Şekil 2.4'de Hamming Penceresi gösterilmiştir.



Şekil 2.4 Hamming window

Diğer pencereleme fonksiyonları ise sırasıyla Hanning, Blackman ve Diktörgen pencereleme yöntemleridir.Şekil 2.5'de de görüleceği üzere Hamming penceresinin bir sinyale nasıl uygulanacağı görülmektedir.



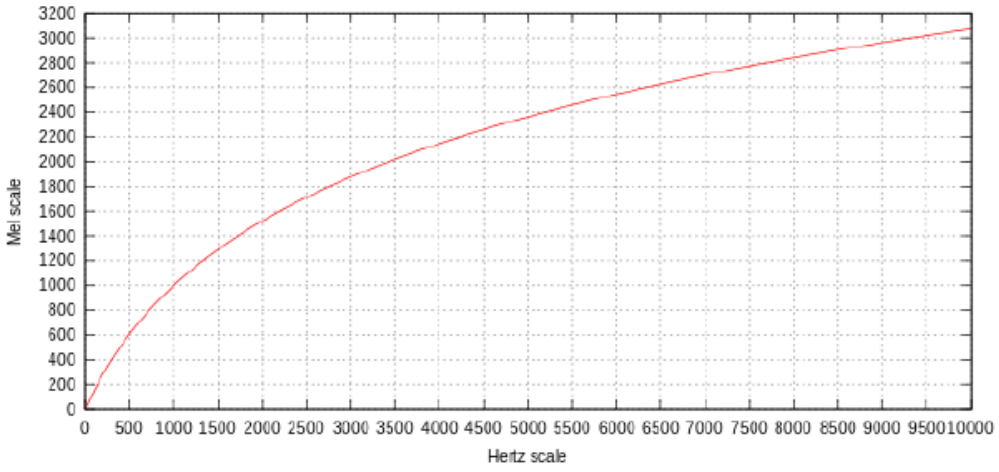
Şekil 2.5 Pencereleme metodu uygulanmış bir sinyal

Ayrık Zaman Fourier Dönüşümü (DFT): Çerçevelere bölüp pencreleme uyguladığımız her bir parça için FFT dönüşümü sağlanır. Ana amacımız ise her çerçevenin güç spektrumunu hesaplayabilmektir. Çünkü, gelen seslerin frekansına bağlı olarak farklı noktalardaki titreşimler insan vücudunda koklea (kulaktaki bir organ) tarafından motive edilir. Kokleadaki titreşen (küçük kılları sallayan) yere bağlı olarak, farklı sinirler, beyne belirli frekansların mevcut olduğunu bildirir. Çerçevede hangi

frekansların mevcut olduğunu belirleyerek bizim için benzer bir iş gerçekleştirebilmek için güç spektrumunu doğru bir şekilde hesaplayabilmekten geçer. Her bir frame için güç spektrumu N= 512 olmak üzere aşağıdaki denklem ile hesaplanır.

$$P=\frac{FFT(xi)^2}{N}$$

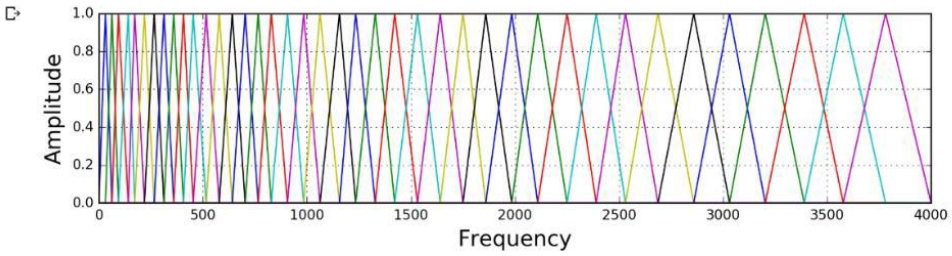
Fast Fourier Transform (FFT), Ayrık Fourier Dönüşümü'nün (DFT) daha hızlı bir versiyonudur. FFT, DTF ile aynı amaç için bazı akıllı algoritmalar kullanır, ancak işlemleri çok daha hızlı şekilde gerçekleştirir. Discrete Fourier Transform (DFT), frekans spektrum alanında oldukça önemli bir kavramdır. Çünkü zaman alanındaki ayrık bir sinyali frekans domenine çevirmek için bir araç olarak kullanılır. DFT ve FFT'yi basitçe karşılaştırmak gerekirse Ayrık Fourier Dönüşümü (DFT) olarak adlandırılan, zaman alanı sinyallerini frekans alanı bileşenlerine dönüştüren algoritmadır. DFT, adından da anlaşılacağı gibi, gerçekten ayrıktır; ayrık zamanlı etki alanı veri kümeleri ayrık frekans gösterimine dönüştürülür. Basit bir ifadeyle, zaman alanı gösterimi ile frekans alanı gösterimi arasında bir ilişki kurar. Hızlı Fourier Dönüşümü veya FFT, büyük dönüşümlerin hesaplama süresini ve karmaşıklığını azaltan bir hesaplama algoritmasıdır. Yani FFT sadece DFT'nin hızlı hesaplanması için kullanılan bir algoritmadır. Bir DFT zaman karmaşıklığında O($N2$) olarak gerçekleştirilebilirken, FFT zaman karmaşıklığını O(N logN) sırasına göre azaltır. Mel Filtre Uygulanması: Mel bir tonun algılanan frekansının ölçüsüdür. Bu ölçü tonun fiziksel frekansıyla lineer olarak örtüşmez. Çünkü insan duyma sistemi frekansları lineer olarak algılayamaz. Bundan dolayı mel skalası deneyi yapılmıştır. 1000 Hz'lik bir ton frekans olarak seçilmiş ve buna 1000 mels denilmiştir. Daha sonrasında ise algıladıkları frekans 2 katına çıkarılmıştır ve bu frekans 2000 mels olarak adlandırılmıştır. Devam eden süreçte bu deney frekasnının 10, 100, 0.5 vb katları için tekrarlanmıştır. Bu sonuçlar sırasıyla 10000 mels, 100000 mels ve 500 mels olarak adlandırılmıştır. Bunun sonuunda ise gerçek fiziksel frekans (Hz) ile algılanan frekans (Mel) arasında bir eşleşme yapma imkânı doğmuştur. Bu eşleşmenin 1 Khz'in üstünde logaritmik altında ise lineer olduğu görülmüştür.



Şekil 2.6 Mel skalası

Çoğu mel ölçekli formül 1000 Hz'de tam olarak 1000 mels verir. Kesme frekansı (örneğin 700 Hz, 1000 Hz veya 625 Hz), formülün normal formundaki tek serbest parametredir. Bazı erimeyen işitsel frekans ölçeği formülleri aynı formu kullanır, ancak çok daha düşük kopma frekansıyla 1000 Hz'de 1000 ile eşleme yapmak zorunda değildir; örneğin Glasberg & Moore'un (1990) ERB oranı ölçeği 228.8 Hz kırılma noktası, ve Greenwood'un koklear frekans yeri haritası (1990) 165.3 Hz kullanır. Mel skalası için diğer fonksiyonel formlar Umesh ve diğerleri tarafından araştırılmıştır. Logaritmik bir bölgeye ve doğrusal bir bölgeye sahip geleneksel formüllerin, bu eğrilerden yaptıkları aşağıdaki ölçümler tablosuna dayanarak Stevens ve Volkmann'ın eğrilerinden ve diğer bazı formlardan gelen verilere uymadığına dikkat çekmektedir. Spektral tahmini hala Otomatik Konuşma Tanıma (ASR) için gerekli olmayan birçok bilgi içermektedir. Özellikle koklea, birbirine yakın aralıklı iki frekans arasındaki farkı ayırt edemez. Bu etki, frekanslar arttıkça daha belirgin hale gelir. Bu nedenle, çeşitli frekans bölgelerinde ne kadar enerji bulunduğuna dair bir fikir edinmek için periodogram kümelerini alıp toplarız. Bu, Mel filtre bankamız tarafından gerçekleştirilir: ilk filtre çok dardır ve 0 Hertz yakınında ne kadar enerji bulunduğunu gösterir. Frekanslar yükseldikçe, varyasyonlar konusunda daha az endişe duyduğumuz için filtrelerimiz genişler. Filtre bankası üçgen bant geçiren frekans karakteristiğine sahiptir ve bant genişliği sabit mel frekansı aralıklarıyla belirlenir. Filtre bankası 13

doğrusal aralıklı ve 27 logaritmik aralıklı filtreden oluşur. Şekil 2.7'de ve Şekil 2.8'de oluşturulan mel filtre bankası görseli ve band aralıkları görülmektedir.



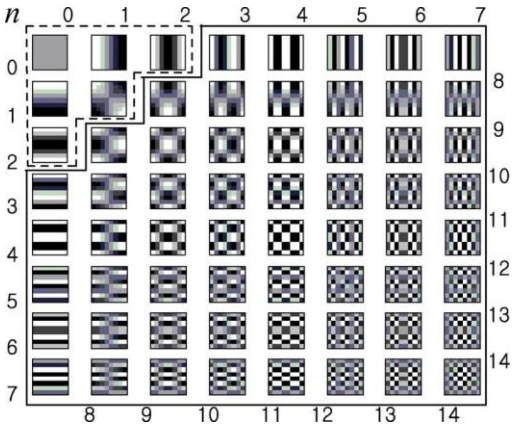
Şekil 2.7 Colab mel filtre bankası

Index	Center Freq. (Hz)	BW (Hz)
1	100	100
2	200	100
3	300	100
4	400	100
5	500	100
6	600	100
7	700	100
8	800	100
9	900	100
10	1000	124
11	1149	160
12	1320	184
13	1516	211
14	1741	242
15	2000	278
16	2297	320
17	2639	367
18	3031	422
19	3482	484
20	4000	556
21	4595	639
22	5278	734
23	6063	843
24	6964	969

Şekil 2.8 Mel skala aralıkları

Filtre bankası enerjilerini elde ettikten sonra, bunların logaritmasını alırız. Genellikle bir sesin algılanan ses seviyesini ikiye katlamak için 8 kat daha fazla enerji koymamız gerekir. Bu, sesin başlaması yüksekse, enerjideki büyük değişikliklerin o kadar farklı gelmeyebileceği anlamına gelir. Bu sıkıştırma işlemi özelliklerimizin insanların gerçekte duyduklarıyla daha yakından eşleşmesini sağlar. Logaritma, kanal normalizasyon tekniği olan cepstral ortalama çıkarmayı kullanmamızı sağlar.

DCT Matris: Son adımda ise elde ettiğimiz logaritmik mel filtre bankası enerjilerinin DCT'sini hesaplamaktır. DCT (Discrete Cosine Transform) bir işaretin frekans bileşenlerine ayrılmasıdır. Bunun sebebi ise mel spektrumunun logaritması zaman bölgesine çevrilmesi olduğundan, frekans tanım bölgesine geçmek için DCT yönteminin seçilmesidir. Şekil 2.9'da DCT Matris uygulaması görülmektedir.



Şekil 2.9 DCT matris uygulaması

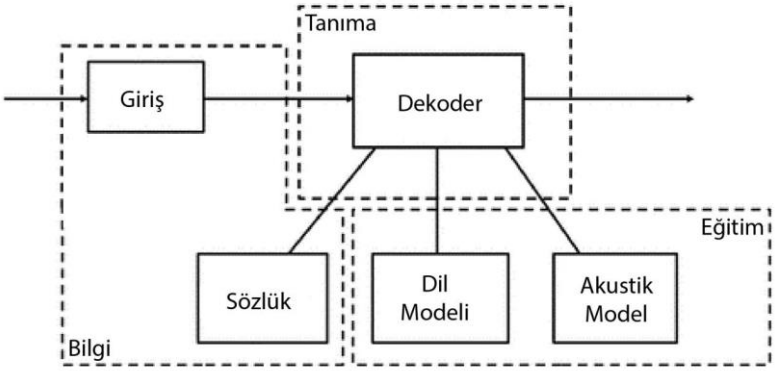
DCT matrisini elde ettikten sonra discrete bir konuşma sinyaline ait MFCC katsayılarını elde etmiş olduk. Fakat her MFCC katsayısının gücü aynı değildir, giderek azalmaktadır. Buyüzden MFCC katsayıları ile işlem yaparken ilk en güçlü 10 MFCC katsayı işleme alınır.[5]

Bölüm 3

Uçtan Uca Ses Tanıma

3.1 Uçtan Uca Ses Tanıma Modeli

Ses tanıma sistemlerine ait litaratürlerde son dönemde sıkça görülmeye başlayan uçtan uca ses tanıma modeli; dilden, konudan ve konuşmacıdan bağımsız işlemlerin başarıyla tamamlanması için geliştirilmeye çalışılan bir modeldir. Uçtan uca ses tanıma modelinin temel özelliği Şekil 3.1'de gösterilen tüm ses tanıma adımlarında yapay zekâ veya derin öğrenme metotlarını kullanmasıdır. Derin öğrenme metotları bu adımlarda klasik modellerde kullanılan metotları desteklemek için kullanılabileceği gibi tek başına da kullanılabilmektedir.



Şekil 3.1 Uçtan uca ses tanıma model adımları

Derin öğrenme metotları, kullanıldığı sistemlerde kullanıcı bağımlılığını en aza indirgemesi ile ön plana çıkmaktadır. Sistem kurulurken karşılaşılabilecek hataları en aza indirgeyerek daha istikrarlı ve hatalara dayanıklı sistemler kurulabilmesine olanak sağlamaktadır. Oluşturulan modelde derin öğrenme metotlarının kullanımından dolayı sistemin yeterli miktarda veri ile eğitilebilmesi büyük önem arz etmektedir. Bu

noktada yeteri miktarda veriyi eğitmek performans açısından sistemi zorlayarak süreçlerin uzamasına sebebiyet verecektir. Ancak yapay zekâ işlemlerinde GPU kullanımının artması ile beraber bu işlem sürelerinde önemli miktarda azalma gözlemlenmiştir.

Baidu araştırma ekibinin sunmuş olduğu uçtan uca ses tanıma modeli, derin öğrenme metotlarından RNN kullanılarak geliştirilmiştir. Yapılan çalışmalardan elde edilen sonuçlar incelendiğinde tanıma sistemlerinin uçtan uca modellenerek aktif olarak kullanılabilecek seviyelere geldiği görülebilmektedir (Amodei, 2016).

DeepSpeech modelinin Şekil 3.2'de belirtilen veri setlerini kullanarak elde ettiği kelime hata oranı (WER) sonuçlarının insanların aynı verileri dinleyerek elde ettiği sonuçlara yakın olması bu ve benzeri sistemler üzerine yapılan çalışmalarının sonuçlarına olan güvenilirliği arttıracaktır.

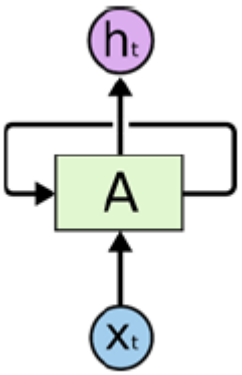
	Test Seti	DeepSpeech2 (WER %)	Mechanical Turk (WER %)
Okuma	WSJ 92	3.10	5.03
	WSJ 93	4.42	8.08
	LibriSpeech test-temiz	5.15	5.83
	LibriSpeech test-diğer	12.73	12.69
Aksanlı	VoxForge Amerikan-Kanadalı	7.94	4.85
	VoxForge Genel	14.85	8.15
	VoxForge Avrupalı	18.44	12.76
	VoxForge Hintli	22.89	22.15
Gürültülü	CHiME Gerçek	21.59	11.84
	CHiME Sim	42.55	31.33

Şekil 3.2 DeepSpeech uçtan uca ses tanıma modeli kelime hata oranı sonuçları (Amodei,2016)

3.2 Tekrarlayan Yapay Sinir Ağları(RNN)

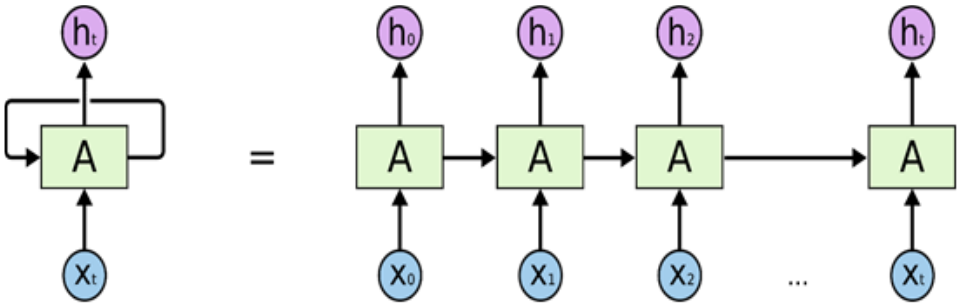
Tekrarlayan yapay sinir ağları bir yapay sinir ağı alt sınıfıdır. Yapay sinir ağlarında kullanılan düğümler arasında döngüsel bağların kurulduğu bir modeldir. İleri beslemeli sinir ağlarından farklı olarak, RNN'ler kendi giriş belleklerini, girdileri işlemek için kullanabilirler. RNN model olarak doğal öğrenme yöntemlerinden tecrübe ile öğrenmeyi temel almaktadır. İnsanlar her adımda öğrenmeye yeniden başlamazlar, her adımda eski tecrübelerinden yararlanarak öğrenmeye devam ederler. Ancak geleneksel yapay sinir ağlarında insanlarda bulunan bu tecrübe ile anlamlandırma özniteliği bulunmaz ve bu onların en büyük eksikliğidir. Örneğin, videodaki tüm karelere bakarak aktiviteler sınıflandırmak istendiğinde, geleneksel sinir ağları kareler arasında insanlar gibi anlamlandırma kuramadığından, sınıflandırma yapamayacaktır. Tekrarlayan Yapay Sinir Ağları ise bir döngü oluşturarak, geçmiş bilgilerin kullanılmasını sağlayacak ve böylelikle kareler arasında anlamlandırma yaparak sınıflandırma

yapabilecektir. Şekil 3.3'de basit tekil tekrarlayan sinir ağ görüntülenmektedir. A ismi verilen dikdörtgen bir yapay sinir ağındaki düğümü temsil etmektedir. Ağın girdi değeri X'dir. Yapay sinir ağının çıktı değeri h'dir. Düğümden değerlendirme sonucu çıkan değer yine kendisine dönerek, döngüsel bir eğitim modeli oluşturmaktadır. Bu döngü ile geçmiş verilerde kullanılabildiğinden yeni bilgi, eski bilgi harmanlanarak bir sınıflandırma yapılabilmektedir.



Şekil 3.3 Tekil tekrarlayan sinir ağ gösterimi (Olah, 2015)

Zaman diliminde, aynı hücre kendini birden fazla tekrar edebilmektedir. RNN uzun süreçte tekrarlayan adımlarda kullanılarak daha detaylı öğrenme imkânı sunabilmektedir. Şekil 3.4'de tekil tekrarlayan yapay sinir ağlarının geçmiş verileri geri besleme olarak her adımda kullanması görselleştirilmiştir.

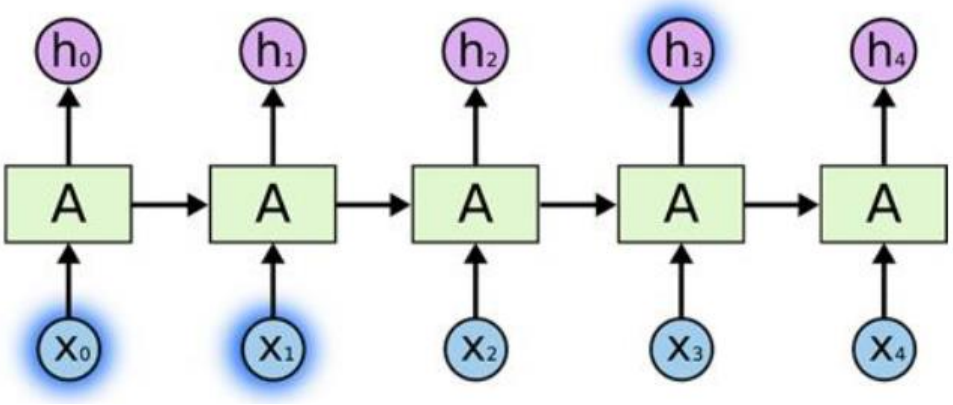


Şekil 3.4 Tekil tekrarlayan sinir ağ geçmiş veri kullanımı (Olah, 2015)

RNN'ler döngüsel olarak çalıştıklarından, sıralı gelişen aktiviteleri birbirleriyle anlamlandırabilmektedir. Akış içerisindeki aktivitelerin anlamlandırılarak yüksek doğrulukla sınıflandırabilmesinden dolayı son yıllarda kullanıcı etkileşimleri, dil modelleme ve ses tanıma sistemlerinde yaygın olarak kullanılmaktadır. RNN ile oluşturulan sistemler geçmiş verileri kullanarak zaman temelli problemlerde başarılı sonuçlar vermektedir. Ancak bu sistemlerde hangi aktiviteler ne kadar süre ile hatırlanacak bilinmemektedir. Bütün bilgiler, modelin içerisinde tutulmaktadır. Aktiviteler için bazı bilgiler önemliyken, bazı bilgiler gereksizdir. Bu yüzden bazı aktivitelerin sınıflandırılmasında, tüm geçmişin saklanmasına gerek yoktur.

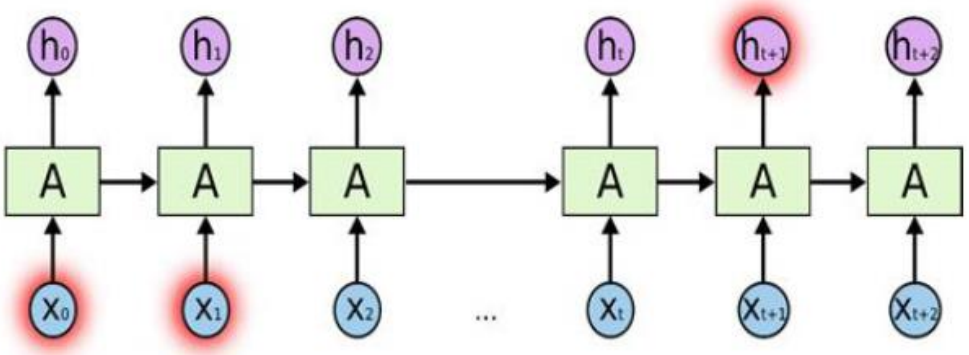
Aktivite sınıflandırılmasında, gerekli bilgi çok önceden oluşmuş ise, bu bilgiye ulaşılamayabilir. Bazı durumlarda ise, bir önceki karedeki olay ile şimdiki karedeki olay birbiri ile bağlantılıdır. Örneğin, aktivite sınıflandırılması yapılacak olan videoda, sırasıyla bazı kişiler yemek masasına oturuyor olsun; böyle bir videoda, önceden masaya oturmuş kişilerden, bir sonraki gelen kişinin de masaya oturacağını tahmin etmek RNN için zor değildir. Çünkü bir sonraki karenin tahmin edilebilmesi için

gerekli olan masaya oturma aktivitesi, çok yakın zamanda gerçekleşmiştir. Şekil 3.5'de hafızadan hatırlanabilir bir RNN gösterilmektedir.



Şekil 3.5 RNN'nin geçmiş verileri kullanarak hatırlama işlemi gösterimi (Olah, 2015)

Bazı aktiviteler, yemek masasına oturma aktivitesinden çok daha karmaşıktır. Örneğin, sınıflandırma yapılacak video, yemek masasında oturulup yemeğe başlandıktan sonra gelen bir kişinin aktivitesi olsun. Yemeğe geç gelen kişinin, yemeğe oturacağının tahminin yapılması insanlar için zor değildir. Fakat RNN için yemeğe oturma aktivitesi bitip, aralara başka aktiviteler girdiği için, yeni kişinin yemeğe oturacağını tahmin etmek zor olmaktadır. Şekil 3.6'da hafızadan hatırlama için bir engel olan ara olayların artması durumu görselleştirilmiştir.



Şekil 3.6 RNN fazla geçmiş veri kullanımının olumsuz etkileri (Olah, 2015)

Teoride, RNN'lerin uzun geçmişteki aktiviteleri, iç mimarisinde kendini tekrarlamakta olduğu için, hatırlayabilme kapasitesine sahiptir. Ancak hatırlanabilmesi için parametrelerin titizlikle seçilmesi gerekmektedir.[4]

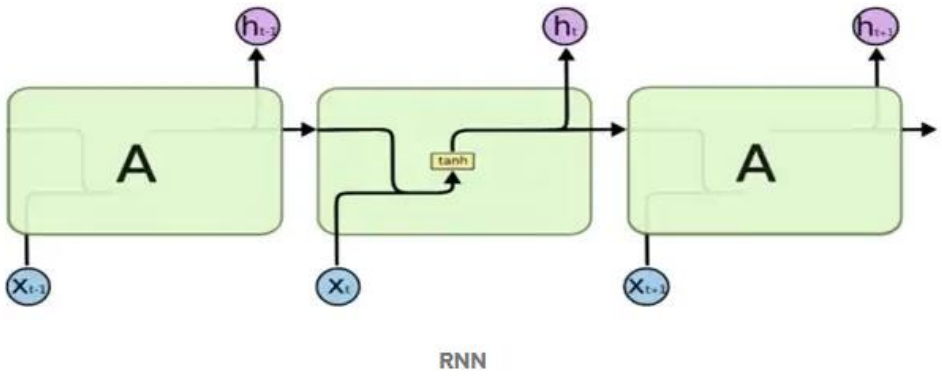
3.3 Uzun -Kısa Vadeli Bellek(LSTM)

LSTM'i daha derinden incelemeye başlamadan önce Özyinelemeli Sinir Ağını(RNN) ve neden bu ağın özel bir yanı olduğunu anlamamız gerekiyor. Bu cümleleri okurken şuan okuduğunuz kelimeleri daha iyi anlayabilmenizi sağlayan şey cümlenin önceki kelimelerini de hatırlıyor olmanızdır. Doğal Dil İşleme(NLP) problemlerinde RNN yapısı geçmiş bilgileri de hatırlayabildiği için başarılı sonuçlar verir. RNN bir önceki adımın çıkışını mevcut adımın girişi olarak kullanır. Fakat diğer klasik derin öğrenme ağlarında giriş ve çıkışlar birbirinden bağımsız olarak çalışır, bu nedenle bir sonraki kelimeyi tahmin etmek gibi temel NLP problemlerinde iyi sonuç veremezler. Sonuç olarak RNN ürettiği her çıkışın bir önceki adıma bağlı olarak ilerlemesini sağlar. Önceki adımlarda hesaplanan sonuçları hafızasında(memory) tutmaya çalışır. Teoride RNN'ler uzun dizilerde(long-sequence) iyi sonuçlar verebiliyor olması gerekirken pratikte elde edilen tecrübeler sonucunda bunu başaramadığı görülür.

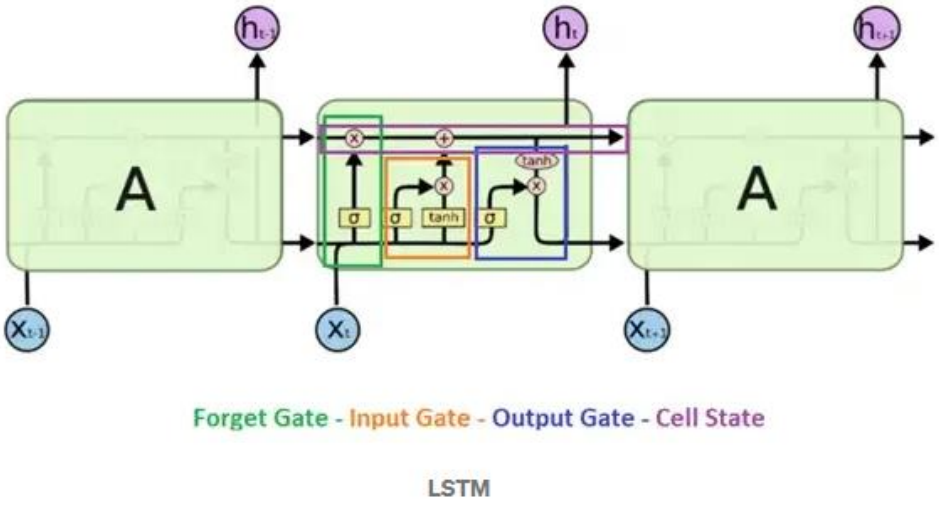
RNN kısa vadeli bir hafızası olmasından dolayı bazı problemlerde iyi çalışamaz. Giriş verisi olarak yeterince uzun bir cümleyi RNN'e verdiğimizde, geçmiş bilgileri hatırlamakta yetersiz kalır ve bu nedenle tahminde bulunmakta zorluk çeker. Ayrıca geriye yayılım(backpropogation) sırasında gradyan yok olması problemi de yaşamaktadır. Gradyan değerleri çok fazla küçüldüğünde RNN öğrenme problemi yaşamaya başlar.

Temel RNN problemlerinden dolayı daha sonra varyantları olan LSTM ve GRU(Geçitlenmiş Özyinelemeli Birimler) gibi çeşitli ağlar önerilmiştir. LSTM, bilgileri daha iyi depolayarak, standart RNN'in kısa vadeli bellek problemini ortadan kaldırmak için tasarlanmıştır.

Tüm RNN'ler aşağıdaki şekilde görüldüğü gibi zincir formunda tekrar eden hücrelerden oluşur. Yazı boyunca hücreler olarak bahsettiğim şekiller ağın her bir zaman adımıdır(time step). Standart bir RNN tek bir tanh katmanı içerirken, LSTM'ler iletişim halinde olan 4 farklı katman içerirler.



Şekil 3.7 RNN bloğu



Şekil 3.8 RNN Bloğuna LSTM eklenmiş hali

LSTM'in temel konsepti Cell State ve kullandığı çeşitli kapılardır(gates). Cell State, tahmin yapmak için anlamlı bilgileri hücreler boyunca taşıyan bir iletişim hattı ve ağın hafızası olarak açıklanabilir. Bu sayede kısa süreli bellek(short-term memory) problemi çözülerek eski veriler ağ zinciri boyunca taşınabilir. Cell State'in bu yolculuğu boyunca taşıması gereken bilgiler ise kapılar aracılığı ile belirlenir. Bu kapılar hangi bilginin gerekli veya gereksiz olduğunu belirleyebilir.Kapılar ise verileri 0 ile 1 aralığına sıkıştıran sigmoid aktivasyon fonksiyonunu kullanır. Sigmoid aktivasyonu sonucunda 0 olan bilgiler unutulur ve 1 olan bilgiler ise Cell State ile ilerlemeye devam eder.

Unutma Kapısı(Forget Gate): Hangi bilgilerin unutulacağı veya tutulacağı kararına varan kapıdır. Önceki hücreden gelen bilgi(ht) ve mevcut bilgi(xt) sigmoid aktivasyon fonksiyonuna sokulur ve sonucuna göre karar verilir. 0 olan bilgiler unutulur ve 1 olan bilgiler Cell State ile taşınmaya devam eder.

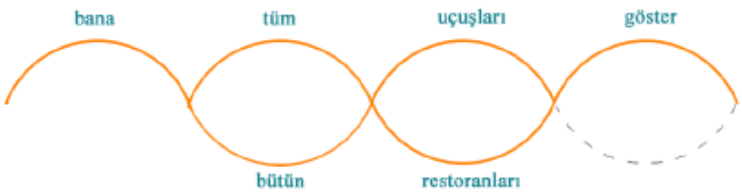
Giriş Kapısı(Input Gate): Cell State güncellemesi yapar. Önceki ve mevcut bilginin sigmoid işlemi sonucuna göre güncelleme yapıp yapılmayacağı kararına varılır. 0 olan bilgi önemsiz ve 1 olan bilgi önemli olarak kabul edilir. Ayrıca ağı düzenleme(regulate) işlemi için veriyi -1 ve 1 aralığına sıkıştıran tanh aktivasyon

fonksiyonu da kullanılır. Daha sonrasında sigmoid ve tanh fonksiyonu çıkışları çarpılır ve hangi bilginin güncelleneceği kararına varılır.

Çıkış Kapısı(Output Gate): Çıkış kapısı ise bir sonraki hücrenin girişini(ht+1) belirler. Ayrıca tahmin yapmak için de kullanılır. Öncelikle önceki bilgiyi ve mevcut girişin bilgisini sigmoid fonksiyondan geçiririz. Daha sonra Cell State üzerinde var olan bilgiyi tanh fonksiyonundan geçiririz. Son olarak iki sonucu çarparak hangi bilgilerin bir sonraki hücre için giriş(ht+1) olacağına karar veririz. Mevcut hücre için kapı işlemleri tamamlandığında bir sonraki hücreye gidecek olan Cell State ve hücrenin giriş bilgisi olarak tanımladığım Hidden State(ht) bilgilerine karar verilmiş olur.[3]

3.4. Dil Modeli Oluşturma Yöntemleri

Dil modelinin oluşturulması ve kullanılması ses tanıma sistemlerinin ana basamaklarından biridir. Ses tanıma sistemleri, ses girdisi işlenerek ardışık kelime gruplarının olasılığı en yüksek olanını seçmek üzerine kurulmaktadır. Bu tahmin aşamasında dil modelleri kullanılmaktadır. Dil modelleri içerdikleri verileri kullanarak ardışık kelimelerin veya kelime gruplarının olasılıksal tahminlerini oluşturmaktadır. Ses tanıma sistemleri ile beraber kullanılan birçok dil modeli bulunmaktadır. Bu dil modellerinden Ngram ve türevleri, kullandığı veriler ile kelime olasılıklarını hazırlaması sebebiyle daha çok veride daha doğru sonuçlar vermesi, kalıp sözcükleri daha kolay öğrenebilmesi ve matematiksel olarak kolay ifade edilerek sistemlere kolay adapte olabilmesi ile ön plana çıkmaktadır. Şekil 3.7.'de dil modelinin karar adımları örnek üzerinde gösterilmiştir.



Şekil 3.9 Dil modeli karar basamakları

3.4.1 N-gram Dil Modeli

N-gram dil modeli, ses verisinden elde edilen kelimenin veya kelime gruplarının istatistiksel olarak değerlendirilerek sonrasında gelecek kelimenin tahmin edilmesi yöntemidir. Modelin matematiksel gösterimi Denklem (3.8) üzerinden incelenebilir.

$$P(W_1, W_2, \dots, W_m) \approx \prod_{i=1}^{m} P(W_i \mid W_{i-n+1}, \dots, W_{i-1})$$

Takip eden kelimelerin tahmini için bir önceki kelimeyi kullanmak yanlış sonuçlar doğurabilmektedir. Bu hatalı sonuçların en aza indirilebilmesi için tahmin edilecek kelimeden önce birden daha fazla kelime grubunu incelemek gerekmektedir. N-gram modelde N değeri bu tahmin için kullanılması gereken geçmiş adım sayısını ifade etmektedir.

3.4.2 Fonem Tabanlı Yaklaşım

Ses tanıma sistemlerinde karakter tabanlı dil modeli harflerin okunuşlarını gelen veriler ile eşleştirerek karakter tahmini yapan bir yöntemdir. Bu yöntem ile tespit edilen karakterler yan yana dizilerek kelimeleri ve kelimeler de cümleleri oluşturacak şekilde düzenlenmektedir. Konuşmalardan karakterleri ayıklamak zorlayıcı bir görev olsa da bu yöntemle hafıza değişkenlerinden bağımsız bir şekilde elde edilen veriler ile dil bilgisine ait temel kurallar takip edilerek doğru sonuçlar elde edilebilmektedir. (Amodei vd., 2016)[4]

3.5 Coqui(Speak to Text)

Coqui speak to text ve text to speak olarak offline çalışabilen motordur. Açık kaynak kodlu ve bir çok dil desteği mevcuttur. Yazılım dillerinden python, C/C++, .NET Framework , java dillerine desteği mevcuttur.

Coqui STT(speak to text) nin çalışma yapısı , STT, bir ses akışını girdi olarak alır ve bu ses akışını belirlenen alfabedeki bir karakter dizisine dönüştürür. Bu dönüştürme iki temel adımla mümkün olur: İlk olarak ses, alfabedeki karakterler üzerinden bir olasılıklar dizisine dönüştürülür. İkincisi, bu olasılık dizisi bir karakter dizisine dönüştürülür.

Derin sinir ağı, ses ve karşılık gelen metin transkriptleri üzerinde eğitilir ve N-gram dil modeli, bir metin külliyatı üzerinde eğitilir (bu, genellikle sesin metin transkriptlerinden farklıdır). Nöral model, konuşmadan metni tahmin etmek için eğitilir ve dil modeli, önceki metinden metni tahmin etmek için eğitilir. Çok yüksek bir seviyede, ilk kısmı (akustik model) fonetik bir kopyalayıcı olarak ve ikinci kısmı (dil modeli) bir yazım ve dilbilgisi denetleyicisi olarak düşünebiliriz.

Projemizi gerçekleştirirken önceden kaydettiğimiz ses (wav.)dosyalarımızı proje dosyasına ekleyip motorumuzu çalıştıracağız.

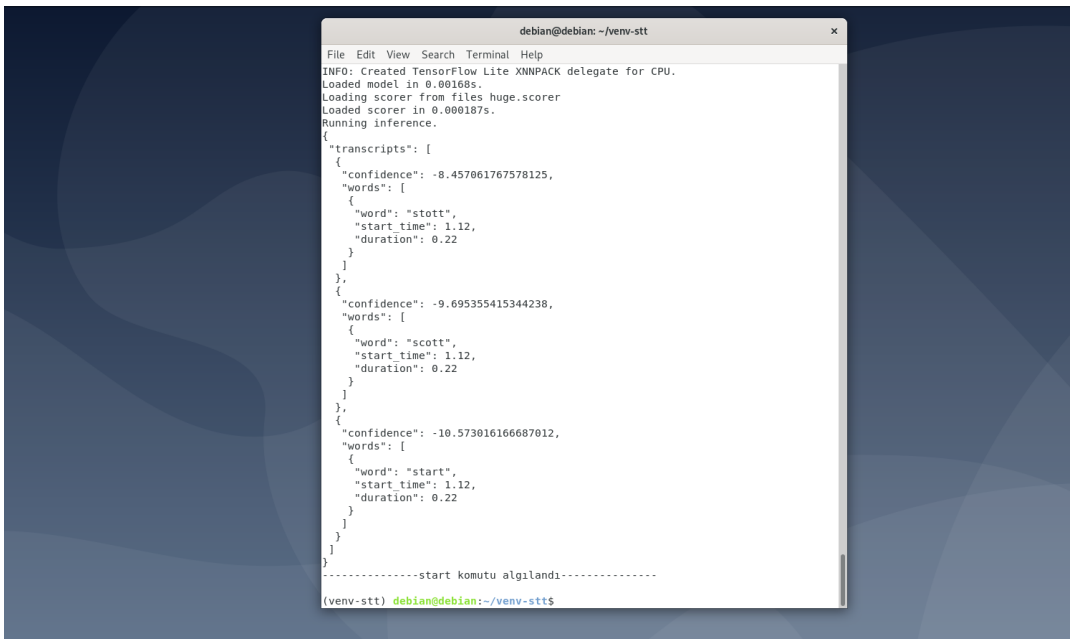
Coqui STT motorunun çalışabilmesi için linux işletim sistemi üzerine python ile kodlama yapacağız. Oluşturmuş olduğumuz motorumuz aracılığı ile wav formatta ingilizce komut ses dosyalarını işleteceğiz.

İlk etapda virtualbox sanallaştırma uygulaması kurulacaktır. Daha sonra üzerine linux kurulumu yapılacaktır. Linux kurulumundan sonra python kurulumları yapılıp python üzerinden coqui STT motorumuzu çalıştırıp ses dosyalarımızı işleyip sonuçlarını json formatında ekrana yazdıracağız. Aynı zamanda önceden tanımlamış olduğumuz komutların sesde bulunan komuta uyup uymadığını kontrol edip ekrana yazdıracağız.

Uygulama kısmında komutlarımız sırası ile start,stop,left,right,up,down dur.

1.Start

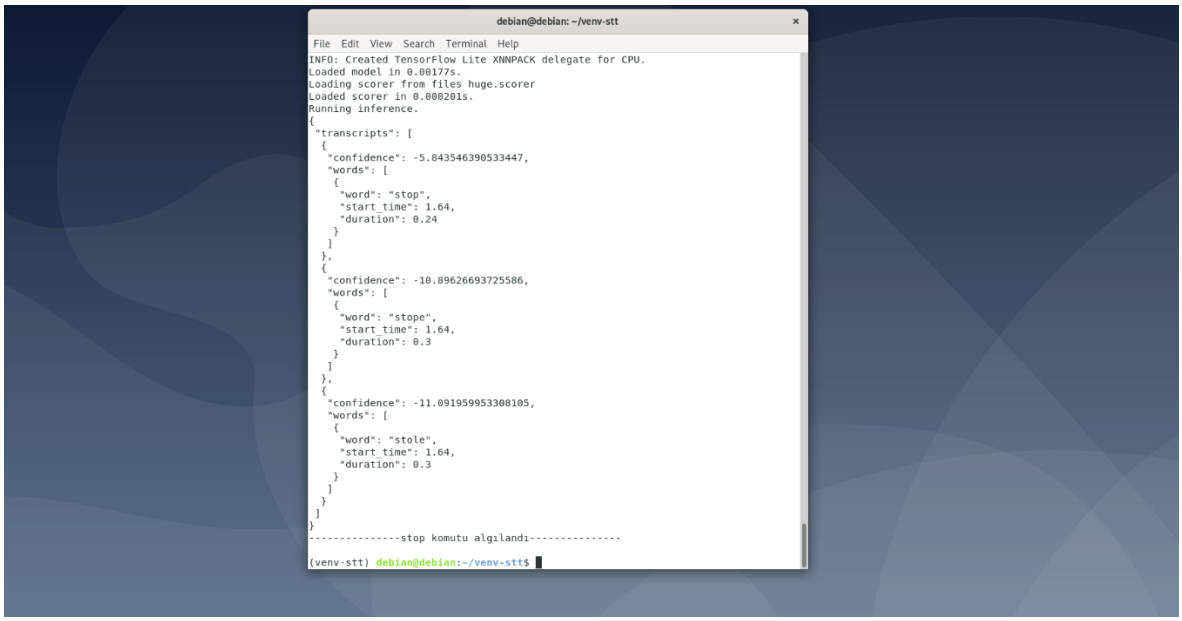
Start.wav dosyamızı import edip kodumuzu çalıştırdığımızda start kelimesinin en yakın tahminlerini json formatta göstermektedir ve json listesinden start kelimesi ile eşleşen veriyi tarayıp ekranda start komutunun algılandığı bilgisi verilmektedir.



Şekil 3.10 Start komutunun tespiti

2.Stop

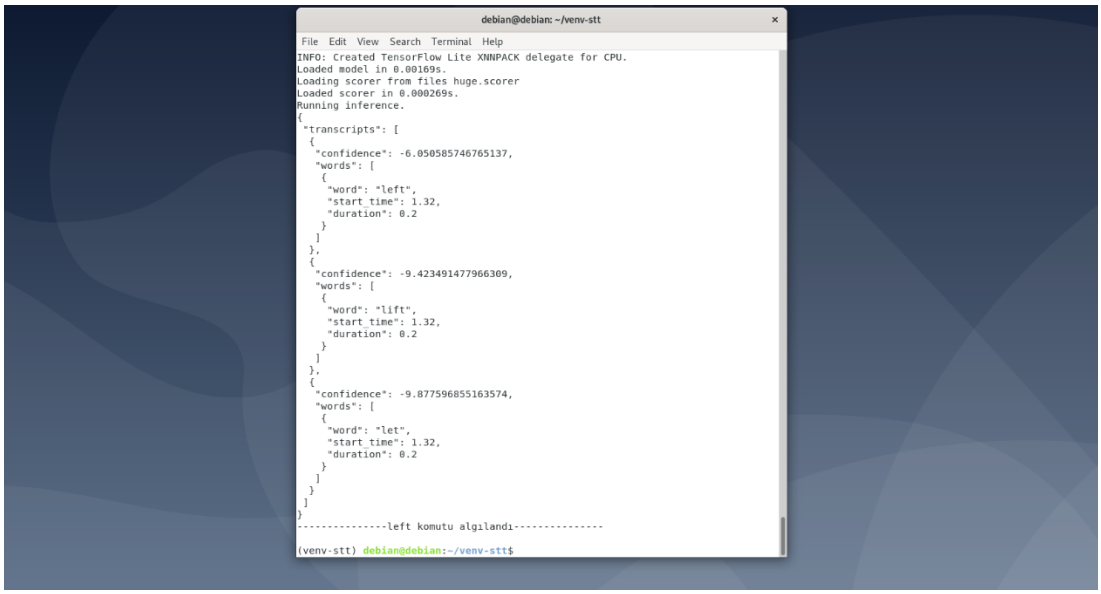
Stop.wav dosyamızı import edip kodumuzu çalıştırdığımızda stop kelimesinin en yakın tahminlerini json formatta göstermektedir ve json listesinden start kelimesi ile eşleşen veriyi tarayıp ekranda stop komutunun algılandığı bilgisi verilmektedir.



Şekil 3.11 Stop komutunun tespiti

3.Left

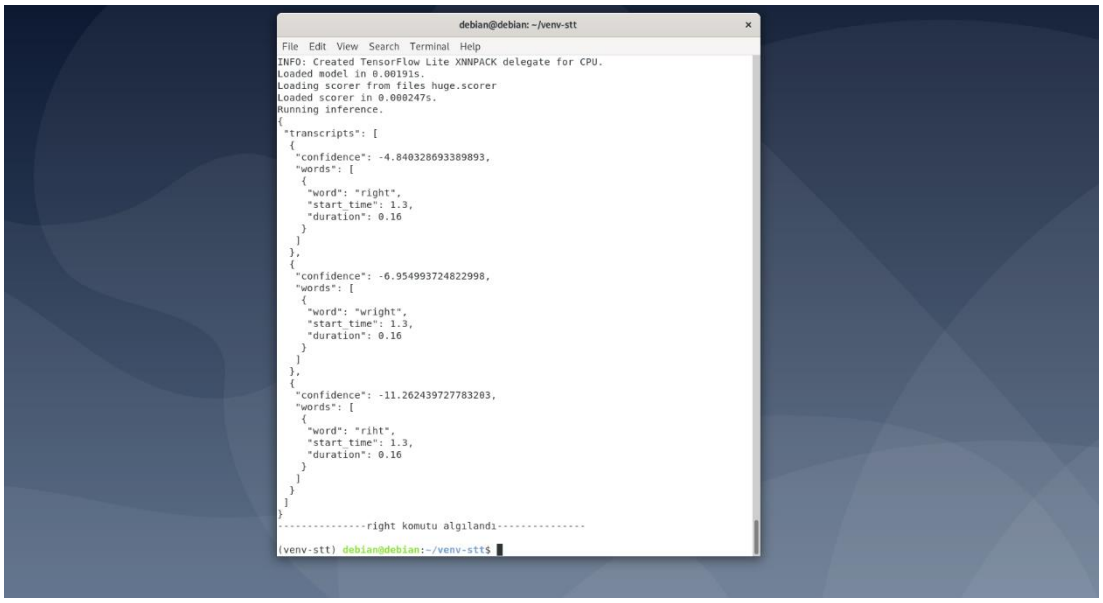
Left.wav dosyamızı import edip kodumuzu çalıştırdığımızda left kelimesinin en yakın tahminlerini json formatta göstermektedir ve json listesinden left kelimesi ile eşleşen veriyi tarayıp ekranda left komutunun algılandığı bilgisi verilmektedir.



Şekil 3.12 Left komutunun tespiti

4.Right

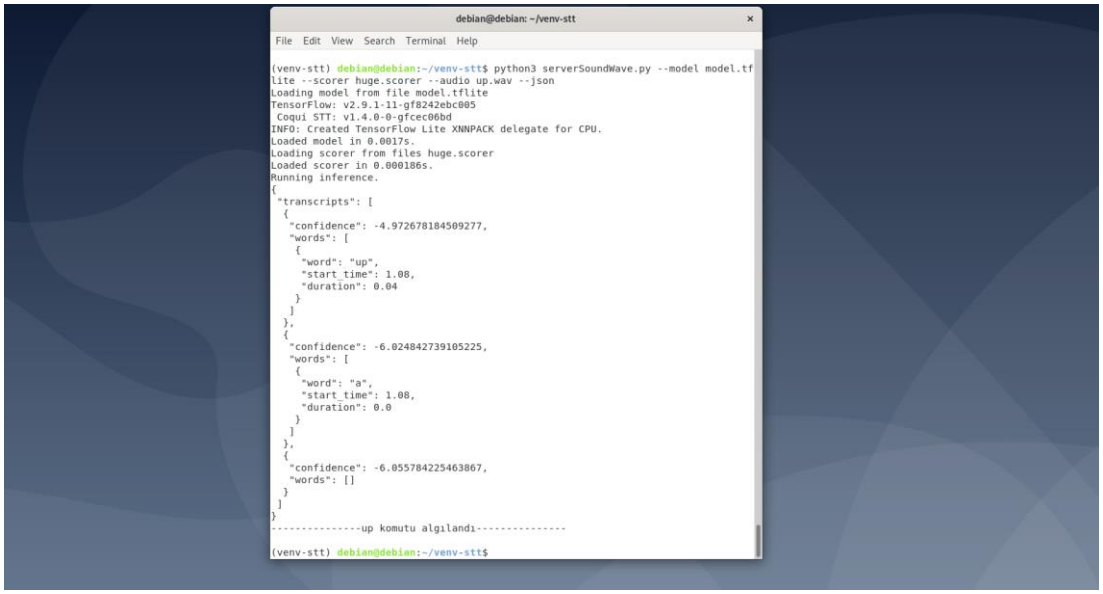
Right.wav dosyamızı import edip kodumuzu çalıştırdığımızda right kelimesinin en yakın tahminlerini json formatta göstermektedir ve json listesinden right kelimesi ile eşleşen veriyi tarayıp ekranda right komutunun algılandığı bilgisi verilmektedir.



Şekil 3.13 Right komutunun tespiti

5.Up

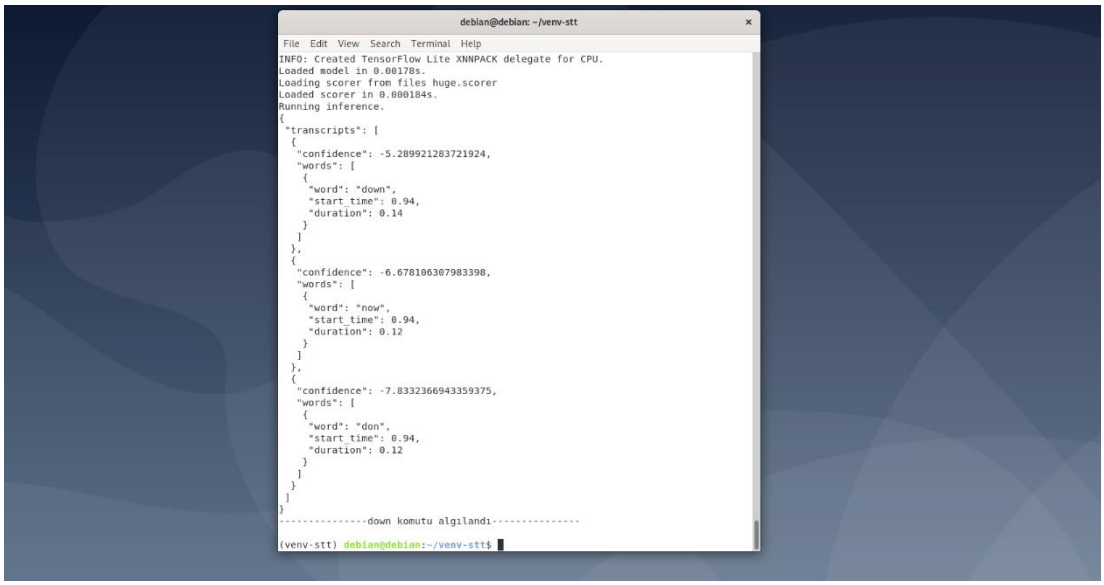
Up.wav dosyamızı import edip kodumuzu çalıştırdığımızda up kelimesinin en yakın tahminlerini json formatta göstermektedir ve json listesinden up kelimesi ile eşleşen veriyi tarayıp ekranda up komutunun algılandığı bilgisi verilmektedir.



Şekil 3.14 Up komutunun tespiti

6.Down

Down.wav dosyamızı import edip kodumuzu çalıştırdığımızda down kelimesinin en yakın tahminlerini json formatta göstermektedir ve json listesinden down kelimesi ile eşleşen veriyi tarayıp ekranda down komutunun algılandığı bilgisi verilmektedir.



Şekil 3.15 Down komutunun tespiti

Sonuç olarak coqui STT motorumuz ile ses dosyasındaki sesin ingilizce dilinde geliştirilmiş model yardımı ve yapay sinir ağları ile tahminde bulunup sonuca ulaşmış olmaktayız.Proje internet gerektirmeden yani bir sunucuya bağlantı talep etmeden offline olarak tahminde bulunabilmesi sebebi ile ve açık kaynak kod yapısı sayesinde geliştirmeye çok açıktır.Bu nedenle gömülü sistemler,mobile sistemler dahil birçok platform için kullanım sağlanabilmektedir.

3.6 Kaynaklar

[1] Amodei, D., Ananthanarayanan, S., Anubhai, R., Bai, J., Battenberg, E., 2016, Deep speech 2 : end-to-end speech recognition in English and Mandarin, Proceedings of Machine Learning Research, 48, p.173-182.

[2] Olah, C., 2015, Understanding LSTM Networks, http://colah.github.io/posts/2015-08-Understanding-LSTMs/, erişim tarihi: 10.08.2019.

[3] Onur Akköse, 2020 ,https://medium.com/deep-learning-turkiye/uzun-k%C4%B1sa-vadeli-bellek-lstm-b018c07174a3

[4] Emre Ateş, Derin öğrenme ile sesli komut tanıma ,2019, https://tez.yok.gov.tr/UlusalTezMerkezi/tezDetay.jsp?id=4SlfV3WTEnBmpTVf3i4AEQ&no=gG6UuRTGT_7HlGEavFCPHw

[5] Can Çetin, Gömülü sistemlerde sesli komut tanıma,2020

[6] https://stt.readthedocs.io/en/latest/, Coqui STT

Özgeçmiş

Adı Soyadı: Görkem Şen

E-mail: gorkemsen@windowslive.com

Eğitim:

2010–2013 9 Eylül Üniversitesi (MYO Elektronik Teknolojisi) (Ön lisans)
2013–2016 Anadolu Üniversitesi İşletme Fakültesi (Lisans)

İş Deneyimi:

2009 – 2010 Pınar ET A.Ş

2013 – 2014 Neta Norm A.Ş

2014 – 2015 Altekma(PLC Otomasyon)

2017 – 2020 Ceyber Eğitim Teknolojileri(Arge Yöneticisi)

2021 – Halen Baylan Water Meters (Gömülü Sistemler Mühendisi)