t.c.

Balıkesir ünİversİtesİ

Fen bİlİmlerİ enstİtüsü

Bilgisayar mühendisliği bölümü

A picture containing text, circle, logo, emblem

Description automatically generated

**GERÇEK ZAMANLI SES VERİLERİNDEN**

**CİNSİYET TESPİTİ**

**FARUK TAYYİB AYDOĞAN**

**BİTİRME PROJESİ TEZİ**

**Danışman : Dr. Öğr. Üyesi FATİH AYDIN**

**BALIKESİR, HAZİRAN - 2023**

ÖZET

Projem, gerçek zamanlı ses verilerinden cinsiyet tespiti yapmaktır. Bu projeyi Matlab programını kullanarak gerçekleştirdim.

Bu projede, önceden eğitilmiş bir model kullanarak ses verilerinden gerçek zamanlı cinsiyet tespiti yapmamızı sağlayan bir Matlab programı geliştiriyoruz . Proje; ses işleme, spektrogram oluşturma ve yapay sinir ağı kullanark cinsiyet tespiti problemini çözmeyi hedefliyor. Kullanılan matlab kodu; ses kaydetme, veri işleme ve CNN (Evrişimli Sinir Ağı) modeli eğitimi adımlarını içerir.

**İÇİNDEKİLER**

Sayfa

ÖZET 1

İÇİNDEKİLER 2

1. Projeye Giriş 4

2. CNN Modeli Kullanımı 5

3. Adım Adım Projemizi Oluşturalım 6

3.1 Ses Verileri 6

3.2 Spektrogram 7

3.3 Spektoframımızı Modelimiz ile Birleştiriyoruz 8

3.4 Cinsiyeti Tahmin Ediyoruz 9

4. Projemdeki Kod Blokları 10

5. Confusion Matrix 17

6. Kaynakça 19

# Projeye Giriş

İlk olarak projeye genel bir bakış açısıyla bakalım, önceden eğitilmiş bir CNN modelini yüklüyoruz. Bu model, erkek ve kadın seslerini ayırt etmeyi öğrenmiş bir yapay sinir ağıdır.

Ses verilerini kaydetmek için mikrofonu kullanarak gerçek zamanlı bir ses kaydı başlatıyoruz. Ses kaydı, 1 saniye arayla her 3 saniyelik ses kaydını işliyor. Kaydedilen ses verilerini audioData değişkeninde saklıyoruz.

Ses verilerini spektrograma dönüştürüyoruz. Spektrogram, sesin frekans bileşenlerini zamanla ilişkilendirilmiş bir şekilde görselleştiren bir grafik türüdür. Bu, sesin frekans özelliklerini modelin anlayabileceği bir formata çevirmemizi sağlar.

Ses verilerinin spektrogramını, önceden eğitilmiş modelin girdi formatına uygun hale getiriyoruz. Bunun için spektrogramı yeniden boyutlandırıyor ve 64x64 piksel boyutuna getiriyoruz. Elde edilen görüntüyü img değişkeninde saklıyoruz.

Önceden eğitilmiş modeli kullanarak tahmin yapma adımına geçiyoruz. Model, img verisini alarak cinsiyet tahminini gerçekleştirir ve sonucu YPred değişkeninde saklar.

Yapılan tahmin sonucunu YPred değerine göre erkek, kadın olarak ekrana yazdırıyoruz.



S.Türüdü, Artist, *Gerçek Zamanlı Ses Spektrumu Oluşturma.* [Art]. 2021.

# CNN Modeli Kullanımı

# 

CNN (Evrişimli Sinir Ağı), derin öğrenme alanında yaygın olarak kullanılan bir yapay sinir ağı modelidir. Özellikle görüntü işleme ve desen tanıma gibi görsel veri analizi görevlerinde etkili bir şekilde kullanılır. CNN, verilerin içerdiği özelliklerin hiyerarşik olarak çıkarılmasını sağlayan evrişim (convolution) ve havuzlama (pooling) katmanlarını içerir.

CNN modeli, bir önceki çalışmada genellikle büyük veri setlerinde uzun süreler boyunca eğitilen bir CNN modelidir. Bu model, genellikle geniş bir veri kümesi üzerinde eğitilmiş ve genel bir özellik çıkarımı yeteneğine sahiptir. Örneğin, geniş bir görüntü veri seti üzerinde eğitilen bir CNN modeli, görüntülerin belirli nesneleri tanıma yeteneğine sahip olabilir.

CNN modeli, genellikle bir dizi evrişim katmanı, havuzlama katmanı ve tam bağlantılı katmanlardan oluşur. Evrişim katmanları, görüntülerdeki özellikleri çıkarmak için filtreler kullanır. Havuzlama katmanları, özellik haritalarını boyutunu azaltarak ve önemli bilgileri koruyarak sıkıştırır. Tam bağlantılı katmanlar ise, özellik haritalarını sınıflandırma için kullanılır.

CNN modeli, genellikle birçok örnekleme adımından geçerek, büyük bir veri kümesindeki örnekleri sınıflandırmak için eğitilir. Eğitim süreci boyunca, model, giriş verileriyle çıktı arasındaki ilişkiyi öğrenir ve bu bilgiyi içeren ağırlıkları günceller. Bu sayede, model veri setindeki örüntüleri ve özellikleri algılamayı ve sınıflandırmayı öğrenir.

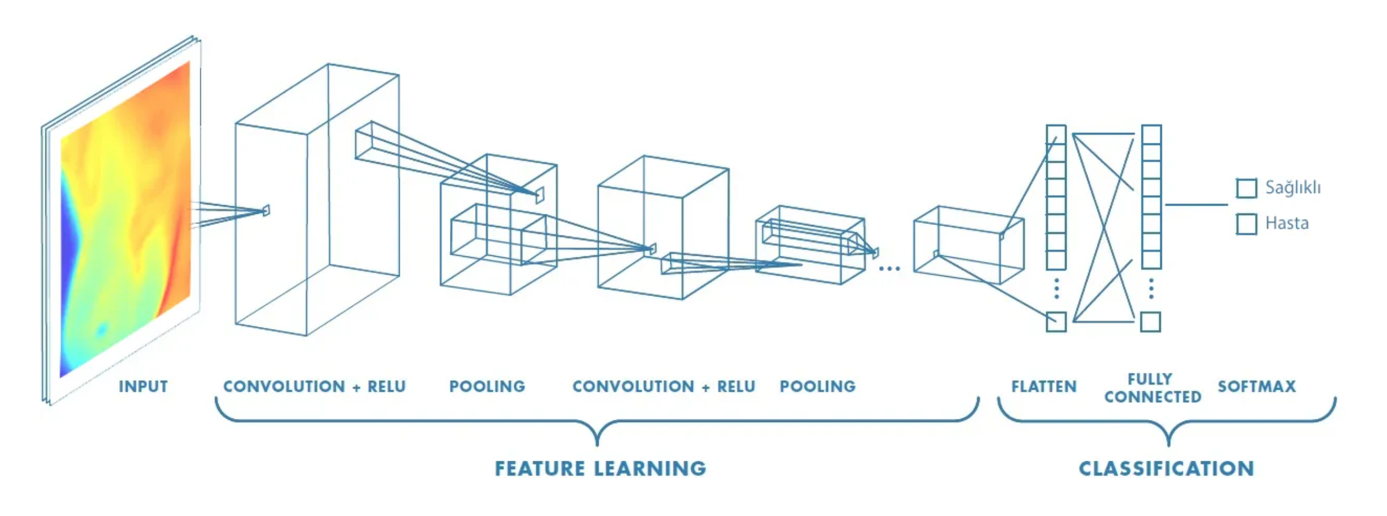
CNN modeli kullanmanın avantajlarından biri, büyük ve karmaşık veri setlerinde uzun süreler boyunca eğitilmiş olmasıdır. Bu, modelin genel özellikleri ve örüntüleri anlamak için geniş bir veri yelpazesine maruz kaldığı anlamına gelir. Özellikle görüntü verileriyle çalışırken, önceden eğitilmiş bir CNN modeli, nesne tanıma, yüz tanıma, duygusal analiz gibi çeşitli görsel analiz görevlerinde etkili sonuçlar elde etmek için kullanılabilir.

CNN modeli, genellikle popüler derin öğrenme kütüphanelerinde (örneğin, TensorFlow, PyTorch, Keras) yaygın olarak bulunur. Bu kütüphanelerde, önceden eğitilmiş modellerin ağırlıkları ve yapıları önceden tanımlanmış olarak sağlanır. Bu sayede, kullanıcılar bu modelleri indirip kullanabilir ve kendi verilerine uyarlayabilir.

CNN modeli kullanmanın bir diğer avantajı, veri toplama ve model eğitimi için harcanacak zamanı azaltmasıdır. Önceden eğitilmiş bir model, birçok örnekleme adımını ve uzun süreler boyunca süren eğitim sürecini içerir. Dolayısıyla, kendi veri kümenizi toplamadan veya modeli eğitmeden önce bu modeli kullanarak hızlı sonuçlar elde edebiliriz.

Ancak, önceden eğitilmiş bir CNN modeli kullanmanın bazı sınırlamaları da vardır. Önceden eğitilmiş modelin genel özellik çıkarımı yetenekli olsa da, belirli bir uygulama veya veri kümesi için özelleştirilmiş olmayabilir. Bu nedenle, önceden eğitilmiş modeli kendi verilerinize uyarlamak ve hedef göreviniz için daha iyi sonuçlar elde etmek için bazı ince ayar (fine-tuning) adımları uygulamanız gerekebilir.

Özetlemek gerekirse, önceden eğitilmiş bir CNN modeli, derin öğrenme ve görüntü işleme alanında yaygın olarak kullanılan bir yapay sinir ağı modelidir. Geniş bir veri kümesinde eğitilmiş olup genel özellik çıkarımı yeteneğine sahiptir. Bu modelleri kullanarak, hızlı sonuçlar elde edebilir ve belirli bir görev için özelleştirmek için ince ayar adımları uygulayabiliriz.



*Tuncer Ergin, Artist, Convolutional Neural Network (ConvNet yada CNN) nedir, nasıl çalışır?). [Art]. 2018.*

# Adım Adım Projemizi Oluşturalım

## Ses Verileri

İlk olarak, MATLAB'da ses kaydetmek için mikrofonun kullanılabilmesi için audiorecorder nesnesini oluşturmanız gerekmektedir. Bu nesne, ses kaydı işlemlerini gerçekleştirmenizi sağlar.

Mikrofonun örnekleme oranını belirlememiz gerekmektedir. Örnekleme oranı, saniyede alınan ses örnek sayısını ifade eder. Örnekleme oranı, sesin kalitesini ve hassasiyetini belirler. Örneğin, orneklem\_orani = 16000; şeklinde bir değer belirleyebiliriz.

Ses kaydı için bir süre belirlemeliyiz.1 saniye arayla ses kaydı yapıp bunu grafik haline getirmek için recordblocking fonksiyonunu kullanıldı. Örneğin, recordblocking(recorder, 1); komutunu kullanarak 1 saniyelik bir kayıt yapabiliriz.

Kaydedilen ses verilerine erişmek için getaudiodata fonksiyonunu kullanılmıştır. Bu fonksiyon, ses kayıt nesnesinden kaydedilen ses verilerini alır ve bir dizi olarak döndürür. Bu ses verilerini audioData değişkeninde saklıyoruz. Örneğin, audioData = getaudiodata(recorder);

Bu adımları takip ederek, MATLAB'da mikrofonu kullanarak gerçek zamanlı bir ses kaydı yapılmıştır. Kaydedilen ses verilerini audioData değişkeninde saklayarak, daha sonra işleme veya analiz için kullanılmıştır.

Dikkat etmeniz gereken noktalar:

Ses kaydedebilmek için bilgisayarınızda uygun bir mikrofonun takılı olması gerekmektedir.

Mikrofonun örnekleme oranı, kaydetmek istediğiniz sesin frekans aralığına uygun olarak seçilmelidir.

Kayıt süresi, projenin gereksinimlerine bağlı olarak ayarlanmalıdır.

## Spektrogram

Spektrogram, sesin frekans bileşenlerini zamanla ilişkilendirilmiş bir şekilde görselleştiren bir grafik türüdür. Bir ses sinyali, zaman boyunca değişen frekanslardaki ses bileşenlerini içerir. Spektrogram, bu frekans bileşenlerini zaman ekseniyle ilişkilendirerek sesin frekans özelliklerini görsel bir formata dönüştürür.

Spektrogram oluşturmak için, ses verileri öncelikle belirli bir süre boyunca küçük parçalara bölünür. Bu süre, analiz edilen sesin zaman çözünürlüğünü belirler. Her parça daha sonra Fourier dönüşümü gibi bir frekans analizi yöntemi kullanılarak frekans bileşenlerine dönüştürülür.

Bu frekans bileşenleri, genellikle güç veya yoğunluk olarak ifade edilir ve renk kodlamasıyla görselleştirilir. Yüksek güçlü frekans bileşenleri daha parlak renklerle gösterilirken, düşük güçlü frekans bileşenleri daha koyu renklerle gösterilir.

Spektrogram, ses verilerinin frekans özelliklerini analiz etmek ve belirli frekans bileşenlerini belirlemek için yaygın olarak kullanılır. Örneğin, konuşma sesi analizinde, Spektrogram, belirli frekans aralıklarında yoğunlaşmaları görselleştirerek konuşmacının vokal seslerini ve farklı kelimeleri tanımlamaya yardımcı olur.

Projede, ses verilerini Spektrograma dönüştürerek, sesin frekans bileşenlerini zamanla ilişkilendirilmiş bir şekilde görselleştiriyorsunuz. Bu, sesin frekans özelliklerini modelin anlayabileceği bir formata çevirmenizi sağlar. Spektrogram, ses verilerinin görsel bir temsilini sağlayarak, özellik çıkarımı ve sınıflandırma gibi işlemleri gerçekleştirmek için kullanılan bir araçtır.

Unutmayın, spektrogram oluştururken kullanılan parametreler (örneğin, zaman dilimi, frekans çözünürlüğü) spektrogramın elde ettiği sonucu etkileyebilir. Bu nedenle, projenin gereksinimlerine göre bu parametreleri ayarlamak önemlidir.

## Spektrogramımızı Modelimiz ile Birleştiriyoruz

Spektrogram, ses verilerinin frekans bileşenlerini zamanla ilişkilendirilmiş bir şekilde görselleştiren bir grafik türüdür. Bu spektrogramı, önceden eğitilmiş bir modelin girdi formatına uygun hale getirmemiz gerekmektedir.

Bu dönüşüm adımında, spektrogamı öncelikle yeniden boyutlandırmanız gerekmektedir. Genellikle, görüntü işleme ve derin öğrenme modelleriyle çalışırken, giriş verilerinin belirli bir boyuta sahip olması beklenir. Bu durumda, spektrogramın 64x64 piksel boyutuna getiriyoruz Bu boyut, modelin işlemesi ve analiz etmesi için uygun bir boyuttur.

Yeniden boyutlandırma işlemi, spektrogramın boyutunu değiştirmek ve piksellerin oranını korumak için bir yeniden örnekleme veya yeniden ölçeklendirme işlemi kullanır. Örneğin, MATLAB'da imresize fonksiyonunu kullanarak spektrogramı 64x64 boyutuna yeniden boyutlandırabiliriz.

Bu işlem sonucunda elde edilen görüntüyü img değişkeninde saklıyorsunuz. Bu img görüntüsü, spektrogramın yeniden boyutlandırılmış halini temsil eder ve önceden eğitilmiş modelin girdi formatına uygun hale getirilmiştir.

Bu dönüşüm işlemi, spektrogramı modelin anlayabileceği bir formata çevirerek, özellik çıkarımı ve sınıflandırma gibi işlemleri gerçekleştirmek için kullanılan bir ara adımdır. eğitilmiş model, bu yeniden boyutlandırılmış spektrogramı alarak cinsiyet tahmini yapacaktır.

Bu şekilde, spektrogramın önceden eğitilmiş modelin girdi formatına uygun hale getirilmesi adımı gerçekleştirilir. Bu, spektrogramı modelin anlayabileceği bir boyuta yeniden boyutlandırarak, verileri modelin işleyebileceği bir formata dönüştürmeyi sağlar.

## Cinsiyeti Tahmin Ediyoruz

Önceden eğitilmiş bir modeli kullanarak tahmin yapma adımı, modelin giriş verisi olan img'yi alarak cinsiyet tahminini gerçekleştirmesini sağlar. Bu adımda, önceden eğitilmiş modelin içindeki ağırlıklar ve öğrenilen özellikler kullanılarak tahmin yapılır.

classify fonksiyonunu kullanarak, önceden eğitilmiş modeli img verisiyle besleyerek tahmin yapabiliriz. Bu fonksiyon, girdi verisini alır ve modele uygular, ardından tahmini sonucu döndürür. Tahmin sonucu, YPred değişkeninde saklanır. Bu değişken, önceden eğitilmiş modelin tahmin ettiği cinsiyeti temsil eder.

Son olarak, YPred değerine göre yapılan tahmin sonucunu ekrana yazdırırız. Örneğin, if ve else ifadeleri kullanarak, YPred değerini kontrol eder ve ekrana "Erkek", "Kadın" olarak yazdırırız. Bu şekilde, modelin tahmin sonucunu kullanarak cinsiyeti belirleyebiliriz.

Bu adımlar, eğitilmiş modeli kullanarak veriye dayalı bir tahmin yapma sürecini temsil eder. Model, önceden eğitilen özelliklerini kullanarak giriş verisini analiz eder ve tahmin sonucunu üretir. Bu tahmin sonucunu daha sonra isteğe bağlı olarak kullanabilir, ekrana yazdırabilir veya başka bir şekilde işleyebiliriz.

A screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated with medium confidence

# Projemdeki Kod Blokları

Projemde 4 ana dosya bulunmaktadır ve her biri farklı bir görevi gerçekleştirmektedir. Bu bölümde hem bu dosyaları inceleyeceğiz hem de içerdikleri yapılara değineceğiz.

İlk olarak Matlab programındaki ‘micToRealtimeGender.m’ dosyasını ele alalım, burada ses kaydetme, spektrogram oluşturma ve önceden eğitilmiş bir model kullanarak sesin cinsiyetini tahmin etme işlemleri gerçekleştirilmiştir. Bu işlemler sırasıyla şu şekilde yapılmaktadır ;

Önceden eğitilmiş bir modeli yüklüyoruz. load fonksiyonunu kullanarak lastTrainingModel.mat dosyasını yükleyerek net değişkenine atıyoruz. Bu dosya, modelin MATLAB'da kaydedilmiş hali olmalıdır.

orneklem\_orani değişkeniyle, ses örnekleme oranını belirliyoruz. Bu değer, ses kaydetme işlemi sırasında kaç örnek alınacağını ifade eder. Örnekleme oranı genellikle hertz (Hz) biriminde belirtilir.

audiorecorder fonksiyonunu kullanarak bir ses kayıt nesnesi olan recorder'ı oluşturuyoruz. Bu nesne, sesi kaydetmek için kullanılacak mikrofon ayarlarını ve örnekleme oranını içerir.

disp fonksiyonu kullanılarak "Ses Kaydı Başladı" mesajı ekrana yazdırılıyor.

recordblocking fonksiyonu, belirli bir süre boyunca kayıt yapmamızı sağlar. Burada 5 saniye boyunca kayıt yapılması için recordblocking(recorder, 5); kullanılıyor.

getaudiodata fonksiyonuyla, kaydedilen ses verilerini audioData değişkenine atıyoruz.

Ses verilerini spektrograma dönüştürmek için bazı parametreler belirliyoruz. win, pencere boyutunu, hop, pencere aralığını, nfft, FFT (Hızlı Fourier Dönüşümü) nokta sayısını temsil eder. Bu parametreler, spektrogramın ne kadar ayrıntılı olacağını belirler.

spectrogram fonksiyonunu kullanarak ses verilerini spektrograma dönüştürüyoruz. Bu işlem sonucunda, s değişkenine spektrogram verisi, ~ değişkenine frekanslar ve ~ değişkenine zaman noktaları atanır.

Spektrogram verisini mutlak değere dönüştürmek için abs fonksiyonunu kullanıyoruz ve sonuç S değişkenine atanıyor.

imresize fonksiyonunu kullanarak spektrogramı 64x64 boyutuna yeniden boyutlandırıyoruz ve img değişkenine atıyoruz. Bu, spektrogram verisini CNN modeline girdi olarak kullanmak için hazırlıyor.

Önceden yüklenmiş modele classify fonksiyonunu kullanarak img verisini veriyoruz ve sonucu label değişkeninde saklıyoruz. Bu adımda, model sesin cinsiyetini tahmin ediyor.

label değişkenini, tahmin edilen cinsiyeti ifade eden bir kategoriye dönüştürüyoruz. Eğer tahmin "0" ise, label değişkenine atanan değeri "Erkek" olarak değerlendiriyoruz. Eğer tahmin "1" ise, label değişkenine atanan değeri "Kadın" olarak değerlendiriyoruz

disp fonksiyonunu kullanarak, tahmin edilen cinsiyeti ekrana yazdırıyoruz.

Sesin spektrogramını çizdirmek için spectrogram fonksiyonunu kullanıyoruz. Bu, kaydedilen ses verilerinin frekans bileşenlerini zamanla ilişkilendirilmiş olarak görselleştirir.

Son olarak, tahmin edilen cinsiyeti başlığa (title) ekleyerek çizim güncellemesini (drawnow) sağlıyoruz.

Yorum satırı olarak eklenen % Sonraki 2 saniyelik kayıt için hazırlık kısmı, daha fazla kayıt yapmak için gerekli olan işaretlemeyi içerir. Bu satırın etkinleştirilmesi durumunda, bir sonraki 2 saniye kaydedilir.

İkinci olarak Matlab programında ‘modelTrain..m’ dosyasını ele alalım, erkek ve kadın seslerinden spektrogram verileri oluşturarak bir CNN modelini eğitiyor ve elde edilen modeli kaydediyor. Ardından, test verilerini kullanarak modelin performansını değerlendiriyor ve bir confusion matrix ile sonuçları görselleştiriyor. Bu işlemler sırasıyla şu şekilde yapılmaktadır ;

İlk olarak, erkek ses dosyalarının bulunduğu klasörü erkekFolder olarak, kadın ses dosyalarının bulunduğu klasörü kadinFolder olarak belirtiyoruz.

dir ve fullfile fonksiyonlarını kullanarak her iki klasördeki ses dosyalarının isimlerini alıyoruz ve erkekFiles ve kadinFiles değişkenlerine atıyoruz.

Spektrogram oluşturma için gerekli olan parametreleri belirliyoruz: win pencere boyutunu, hop pencere aralığını ve nfft FFT nokta sayısını temsil eder.

Toplam dosya sayısını belirlemek için numel fonksiyonunu kullanarak erkek ve kadın dosya sayılarını topluyoruz.

Eğitim verisi (XTrain ve YTrain dizileri) için yer ayırıyoruz. Bu diziler, spektrogram verilerini ve etiketleri içerecektir.

Erkek ses dosyalarını işlemek için bir döngü kullanıyoruz. Her bir dosyayı sırayla işlemleyerek spektrograma dönüştürüyoruz.

audioread fonksiyonunu kullanarak ses dosyasını okuyoruz ve ses verilerini ve örnekleme oranını (orneklem\_orani) elde ediyoruz.

Ses verilerini spektrograma dönüştürmek için spectrogram fonksiyonunu kullanıyoruz. Elde edilen spektrogram verisini S değişkenine atıyoruz.

imresize fonksiyonunu kullanarak spektrogramı 64x64 boyutuna yeniden boyutlandırıyoruz ve img değişkenine atıyoruz.

Elde edilen görüntüyü ve etiketi (YTrain) XTrain ve YTrain dizilerine ekliyoruz.

Kadın ses dosyalarını işlemek için bir döngü kullanıyoruz. Benzer şekilde, her bir dosyayı sırayla işlemleyerek spektrograma dönüştürüyoruz.

Eğitim verilerini (XTrain ve YTrain) karıştırma ve bölme işlemleri için cvpartition nesnesini kullanıyoruz. totalFiles sayısına göre verileri bölüyoruz ve eğitim ve test veri setlerini (XTrain\_final, YTrain\_final, XTest, YTest) elde ediyoruz.

CNN modeli için gerekli olan katmanları (layers) oluşturuyoruz. Bu katmanlar, görüntü işleme ve sınıflandırma ayrımlarını gerçekleştirecek olan katmanları içerir. İlk olarak, giriş katmanı (imageInputLayer) görüntülerin boyutunu belirtir. Ardından, çeşitli konvolüsyon (convolution2dLayer) ve havuzlama (maxPooling2dLayer) katmanları ile öznitelik çıkarımı yapılır. Son olarak, tamamen bağlı katmanlar (fullyConnectedLayer) ve çıktı katmanı (softmaxLayer ve classificationLayer) kullanılarak sınıflandırma işlemi gerçekleştirilir.

1. **imageInputLayer([64 64 1]):** Bu, ağın giriş katmanıdır. Burada, modelin alacağı veri türünü ve boyutunu belirtiyoruz. Bu durumda, 64x64 boyutunda ve tek kanallı (gri tonlamalı) bir görüntü veya spektrogram olarak düşünülebilir.
2. **convolution2dLayer(3,8,'Padding','same'):** Bu katman, konvolüsyon işlemi uygulayarak görüntüde özellikleri algılar. Çekirdek boyutu 3 ve 8 filtre kullanılıyor. 'Padding' parametresi 'same' olarak ayarlanmış, bu da girdi ve çıktının aynı boyutta olmasını sağlar.
3. **batchNormalizationLayer:** Bu katman, ağın öğrenme sürecini hızlandırır ve daha kararlı hale getirir. Her katmanda çıktıları normalleştirir.
4. **reluLayer:** Bu, bir aktivasyon katmanıdır ve çıktının doğrusal olmayan hale gelmesini sağlar. Bu, modelin daha karmaşık desenleri öğrenebilmesini sağlar.
5. **maxPooling2dLayer(2,'Stride',2):** Bu katman, girdi özellik haritasının boyutunu azaltır ve böylece modelin hesaplama maliyetini düşürür ve aşırı uyumu azaltır. Burada, her 2x2 piksel alanının maksimum değeri alınır ve her 2 pikselde bir adım atılır.

Bu döngü, 2 ek konvolüsyon ve max pooling işlemi için tekrarlanır, bu sefer daha fazla filtre kullanılır (sırasıyla 16 ve 32). Daha fazla filtre kullanmak, modelin daha karmaşık özellikleri öğrenebilmesini sağlar.

Son olarak, modelde aşağıdaki katmanlar yer alır:

1. **fullyConnectedLayer(numClasses):** Bu katman, önceki katmandan gelen tüm özelliklerin, sınıflandırma işlemi için tamamen bağlandığı yerdir. 'numClasses' parametresi, sınıf sayısını belirtir.
2. **softmaxLayer:** Bu katman, her bir sınıf için olasılıkları hesaplar.
3. **classificationLayer:** Bu katman, çıktıları belirtilen sınıflara göre etiketler. Yani, her bir girdinin hangi sınıfa ait olduğunu belirler.

Eğitim seçeneklerini (trainingOptions) belirliyoruz. Bu seçenekler, eğitim sürecinin nasıl yapılacağını ve modelin nasıl optimize edileceğini belirler.

trainNetwork fonksiyonunu kullanarak CNN modelini eğitiyoruz. Eğitim verilerini (XTrain\_final, YTrain\_final), oluşturulan katmanları (layers) ve eğitim seçeneklerini (options) parametre olarak kullanıyoruz.

Tahmin yapmak için eğitilen modeli kullanıyoruz. classify fonksiyonunu kullanarak XTest verilerini sınıflandırıyoruz ve tahmin sonuçlarını YPred değişkenine atıyoruz.

Doğruluk oranını (accuracy) hesaplıyoruz. Yapılan doğru tahminlerin toplam veri sayısına oranını temsil eder.

Confusion matrix'i (confusionMatrix) hesaplıyoruz. Bu, tahmin edilen sınıfların gerçek sınıflarla karşılaştırılmasını gösteren bir tablodur.

Confusion matrix'i görselleştirmek için confusionchart fonksiyonunu kullanıyoruz.

Modeli kaydediyoruz. save fonksiyonunu kullanarak eğitilen modeli belirtilen yola (path) kaydediyoruz.

Üçüncü olarak Matlab programında ‘predictManuelM4a.m’ dosyasını ele alalım, önceden eğitilmiş modeli kullanarak belirtilen ses dosyasının cinsiyetini tahmin eder. Tahmin sonucunu ekrana yazdırarak ses dosyasının erkek, kadın veya bilinmeyen olduğunu belirtir. Bu işlemler sırasıyla şu şekilde yapılmaktadır ;

İlk satırda, önceden eğitilmiş modeli yüklemek için load fonksiyonunu kullanıyoruz. lastTrainingModel.mat dosyasını yükleyerek net değişkenine atıyoruz.

Tahmin yapmak istediğimiz ses dosyasını audioread fonksiyonuyla okuyoruz. Ses dosyasının yolu audioData değişkenine atılır ve örnekleme oranı (orneklem\_orani) elde edilir.

Ses dosyasını spektrograma dönüştürmek için gerekli parametreleri belirliyoruz: win pencere boyutu, hop pencere aralığı ve nfft FFT nokta sayısı.

spectrogram fonksiyonunu kullanarak ses verilerini spektrograma dönüştürüyoruz. Elde edilen spektrogram verisini S değişkenine atıyoruz.

Spektrogram verisini 64x64 boyutuna yeniden boyutlandırıyoruz ve img değişkenine atıyoruz.

Önceden eğitilmiş modele (net) tahmin yapmak için classify fonksiyonunu kullanıyoruz ve sonucu YPred değişkenine atıyoruz.

Yapılan tahmini kontrol ederek, sonucu erkek (0), kadın (1) veya bilinmeyen olarak (Bilinmeyen) ekrana yazdırıyoruz.

A screenshot of a computer

Description automatically generated with medium confidence

*Faruk Aydoğan, Artist,Gender Detection On Realtime Voice [Art]. 2023.*

# Confusion Matrix

Confusion matrix, bir sınıflandırma modelinin performansını değerlendirmek için kullanılan bir araçtır. Bu matris, modelin tahmin ettiği sınıflar ile gerçek sınıflar arasındaki ilişkiyi gösterir. Confusion matrix, sınıflandırma probleminde "true positive" (gerçek pozitif), "false positive" (yanlış pozitif), "false negative" (yanlış negatif) ve "true negative" (gerçek negatif) değerlerini içeren bir tablo şeklindedir. Bu dört temel terim şu şekildedir :

* True Positive (TP): Gerçek pozitif değerlerin doğru bir şekilde tahmin edildiği durumu ifade eder. Yani modelin doğru bir şekilde "pozitif" olarak sınıflandırdığı örnek sayısı.
* True Negative (TN): Gerçek negatif değerlerin doğru bir şekilde tahmin edildiği durumu ifade eder. Yani modelin doğru bir şekilde "negatif" olarak sınıflandırdığı örnek sayısı.
* False Positive (FP): Gerçek negatif değerlerin yanlış bir şekilde "pozitif" olarak tahmin edildiği durumu ifade eder. Modelin "pozitif" olarak sınıflandırdığı ancak gerçekte "negatif" olan örnek sayısı.
* False Negative (FN): Gerçek pozitif değerlerin yanlış bir şekilde "negatif" olarak tahmin edildiği durumu ifade eder. Modelin "negatif" olarak sınıflandırdığı ancak gerçekte "pozitif" olan örnek sayısı.

Confusion Matrix, bu dört terimi kullanarak sınıflandırma sonuçlarını tablo şeklinde gösterir. Tipik olarak, satırlar gerçek sınıfları temsil ederken, sütunlar modelin tahminlerini temsil eder. Her bir hücre, ilgili sınıfın tahminlerinin sayısını gösterir.

Confusion matrix, modelin doğruluğunu, precision'ını, recall'ını ve F1 skoru gibi diğer performans metriklerini hesaplamaya yardımcı olabilir.

MATLAB'da confusion matrix oluşturmak için "confusionmat" fonksiyonunu kullanabiliriz.

Üst bölümlerde de belirttiğim gibi projemde ‘confusion matrix’ kullandım ve aşağıdaki şekilde sonuç aldım ;

A screenshot of a computer

Description automatically generated

*Faruk Aydoğan, Artist,Gender Detection On Realtime Voice [Art]. 2023.*

# KAYNAKLAR

|  |  |
| --- | --- |
| [1] | B. B. Monson, "Gender and vocal production mode discrimination using the high frequencies for speech and singing," 30 October 2014. [Online]. Available: https://www.frontiersin.org/articles/10.3389/fpsyg.2014.01239/full. |
| [2] | V.Tümen, "Sinaptik Etkinlik Fonksiyon Tabanlı Sızdıran Entegre ve Ateşleme Nöron Modelini Kullanarak İnsan Ses Sinyallerinde Cinsiyet Tespiti," Bitlis Eren Üniversitesi Fen Bilimleri Dergisi*,* pp. 469-477, 2022. |
| [3] | M. Y. Çakır, "GERÇEK ZAMANLI YÜKSEK KALİTEDE SES TANIMA  ," M. Y. Çakır, 2017. |
| [4] | S. Türüdü, " MATLAB ile Gerçek Zamanlı Ses Spektrogramı Oluşturma," 2021. [Online]. Available: https://www.sonerturudu.com/matlab-ile-gercek-zamanli-ses-spektrogrami-olusturma/. |
| [5] | T.Ergin, "Convolutional Neural Network (ConvNet yada CNN) nedir, nasıl çalışır?" ,  2 October 2018 |