TODO LIST

CENTRO UNIVERSITÁRIO DA FEI

ARIEL GRAÇA FERREIRA

ESTIMATIVA DE FAIXA ETÁRIA PELA VOZ COM REDES NEURAIS CONVOLUCIONAIS

SÃO BERNARDO DO CAMPO

2021

Ariel Graça Ferreira

Estimativa de Faixa Etária Pela Voz com Redes Neurais Convolucionais

Qualificação apresentada como requisito parcial para obtenção do título de Mestre em Engenharia Elétrica, pelo Programa de Pós-Graduação em Engenharia Elétrica do Centro Universitário da FEI.

Orientador: Prof. Dr. Ivandro Sanches

São Bernardo do Campo

2021

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

MFCC	Mel Frequecy Cepstral Coeficients
GMM	Gaussian Mixture Models
RNC	Rede Neural Convolucional
API	Application Programming Interface
URA	Unidade de Resposta Audível
IVR	Interactive Voice Response
ASR	Automatic Speech Recognition

SUMÁRIO

1 INTRODUÇÃO

A estimação de faixa etária pela voz, possui diversas aplicações que visam criar uma interface homem-máquina para utilização em áreas como transporte, segurança e medicina.

Um exemplo bastante presente no cotidiano das pessoas atualmente, seriam as Inteligências Artificiais como a *Alexa*, desenvolvida pela Amazon, e o *Google Assistant*, criado pelo próprio Google. São dispositivos espalhados em milhares de casas ao redor do mundo, e que recebem os mais diversos comandos de voz para a execução de várias tarefas, que podem ser desde tocar uma música, até realizar uma complexa rotina de acionamento de outros elementos e/ou equipamentos inteligentes, interconectados e espalhados pela casa de um indivíduo, ou fora dela. Para citar apenas uma aplicação, a estimação de faixa etária poderia funcionar como uma validação de segurança, evitando assim acesso ou acionamento indevido por alguma criança que tenha acesso ao dispositivo mas não possa ter acesso a todas as funcionalidades disponíveis.

Tomando como base esse cenário de crescimento tecnológico e expansão das aplicações de *Internet of Things* e Inteligência Artificial, este trabalho tem por finalidade estudar os sistemas de estimação de faixa etária por sinais de voz que utilizem técnicas de aprendizado profundo, como Redes Neurais Convolucionais (RNC). Busca-se entender e mostrar evidências do desempenho desse tipo de sistema, considerando a utilização de tais técnicas para classificação de dados.

Através do sinal da voz de um indivíduo, é possível obter muitas informações sobre quem produz o som, como gênero, emoções e idade, porém essa tarefa também traz desafios. A incidência de distorções na fonte do sinal, geradas pelo próprio indivíduo (estado de saúde, emoção), ou geradas pelo ambiente (ruído externo), deformam o sinal de tal forma que a etapa de extração das características acústicas (features) fica prejudicada e, consequentemente, a etapa seguinte, a classificação do dado, também vai sofrer com a degradação.

Nesse sentido, as técnicas de aprendizado de máquina, mais especificamente aprendizado profundo, com base em pesquisas realizadas com a literatura atual, mostram-se mais robustas para implementação desse tipo de sistemas que trabalham com sinais acústicos. Por tal razão, este trabalho busca estudar e avaliar a utilização de redes neurais convolucionais no problema de estimação etária.

Para efeitos de comparação e posterior análise dos resultados, será utilizado como referência a pesquisa FAPESP desenvolvida pelo Prof. Dr. Ivandro Sanches (2019), que utiliza técnicas mais tradicionais de aprendizado de máquina, como o Gaussian Mixture Models (GMM's) e i-vector. Visto que o trabalho do Prof. Ivandro faz uso do mesmo banco de dados que será utilizado no desenvolvimento do trabalho com redes neurais convolucionais, haverá a oportunidade de comparar e discutir os resultados obtidos com ambos os sistemas.

Em relação ao banco de dados, será adotado um banco com sinais de chamadas telefônicas que pertence a *Deutsche Telekom Laboratories*, de Berlim na Alemanha.

1.1 Objetivos

- Avaliar o desempenho de um sistema de estimação automático de faixa etária, o qual utiliza técnicas de aprendizado profundo para classificação de dados. Dentre tais técnicas, as Redes Neurais Convolucionais serão um dos principais objetos de estudo do trabalho visto que suas características são propícias para o processamento dos sinais de voz.

- Comparar o desempenho do sistema de estimação de faixa etária que utiliza RNC com o sistema que utiliza GMM e i-vector, desenvolvido pelo Prof. Ivandro Sanches (2019), visto que ambos os trabalhos executam simulações com a mesma base de dados.

- Durante as pesquisas, ensaios e simulações, deve-se avaliar eventuais limitações dos elementos que compõem o sistema de estimação de faixa etária com RNC. Com base na identificação desses limitadores, deve-se buscar e pesquisar sobre formas de mitigar tais fatores, e assim propor otimizações para este tipo de sistema.

2 REVISÃO BIBLIOGRÁFICA

A pesquisa do Prof. Ivandro, Sanches (2019), aborda o problema de estimação de faixa etária utilizando técnicas de Aprendizagem de Máquina como Gaussian Mixture Models (GMM) e i-vectors. Neste trabalho foi desenvolvido ainda uma técnica de estimação de pitch utilizando o algoritmo de Viterbi, e assim obtendo uma taxa de acerto considerando 3 grupos (criança, homem, mulher) de 87.8%. As simulações foram realizadas utilizando mel-frequency cepstral coefficients (MFCC) para caracterizar as amostras de áudio, sendo essa a técnica de extração de features mais consagrada para processamento desse tipo de sinal.

3 METODOLOGIA

3.1 Base de Dados

Corpus aGender da *Deutsche Telekom AG Laboratories*, criado por Burkhardt et al. (2010).

A base de dados escolhida para este trabalho foi utilizada tanto em Sanches (2019), como também em outros trabalhos pesquisado durante a revisão bibliográfica. Trata-se de um banco composto por 65364 arquivos de áudio, que correspondem a um total de 47 horas de gravações telefônicas (somatória de arquivos que possuem, em média, 2.58 segundos de duração), realizadas por 954 indivíduos, com uma distribuição igualitária de gênero dentro de quatro grupos etários: crianças, jovens, adultos e sêniores/idosos. Com base nessa composição, o banco possui 7 classes, divididas conforme tabela abaixo:

Classe	Idade	Gênero
1	7 - 14	f, m
2	15 - 24	f
3	15 - 24	m
4	25 - 54	f
5	25 - 54	m
6	55 - 80	f
7	55 - 80	m

Tabela 1 – Classes

f: feminino / m: masculino

Para elaboração desse banco, cada participante selecionado teve que realizar seis chamadas telefônicas utilizando dispositivos móveis, alternando entre locais internos (*indoor*) e externos (*outdoor*) para obtenção de diferentes níveis de ruídos de ambiente. A cada ligação, o participante interagia com uma Unidade de Resposta Audível (URA, em inglês conhecida como *Interactive Voice Response* - IVR) e tinha que ler as declarações/sentenças pré definidas e providenciadas antecipadamente pela equipe do laboratório. Havia um intervalo de um dia entre uma ligação e outra, para que fosse possível obter maior variação nas vozes dos indivíduos.

Todas as gravações são disponibilizadas em extensão .raw, armazenadas com 8 bits, frequência de amostragem de 8KHz e codificação PCMA.

3.1.1 Base de treinamento

Da base total, foram separados 53076 arquivos, que correspondem a 770 indivíduos e 38.16 horas de gravação. Esse conjunto foi definido como base para treinamentos, onde as

gravações são disponibilizadas já com a indicação de a qual classe cada uma pertence. A tabela abaixo, mostra a divisão dos locutores e gravações (o artigo oficial indicado nas referências desse trabalho, se refere as gravações como *utterances*, traduzido como *declarações*) por classe:

Classe	**1**	**2**	**3**	**4**	**5**	**6**	**7**
#locutores	106	99	88	113	107	123	134
#gravações	6804	7360	6189	7844	6911	8575	9575

Tabela 2 – Distribuições no banco de treinamento

3.2 Sistema de Estimação Automática de Faixa Etária com Aprendizado Profundo

Para a execução das simulações e ensaios com base em aprendizado profundo, será utilizado uma RNC derivada do programa criado por Badine (2020) e publicada na página Web do Keras, o qual é uma API, em linguagem de programação Python, desenvolvida para abstração do framework Tensorflow, criado pelo Google Brain Team, e que possui vasta aplicação na área de Ciência de Dados.

A RNC original do programa em questão é treinada para classificar arquivos de áudio em 5 classes, cada uma correspondendo a um locutor diferente. Trata-se, por tanto, de um sistema de Automatic Speech Recognition (ASR), ou seja, Reconhecimento Automático de Fala, com o objetivo de identificar o locutor a partir de uma trecho de fala. A análise do sinal é realizada no domínio da frequência, e por isso utiliza-se FFT para obtenção dos espectros. Ponto importante que deve ser observado no artigo, seria o fato de ter-se acrescido ruído artificialmente ao sinal de voz para que assim a rede tivesse mais dados para processar. Esse é um ponto de relevância que a pesquisa deve se aprofundar, a fim de investigar a influência do tamanho do banco de dados no desempenho do sistema.

Visto que a finalidade da RNC original seria a de reconhecimento de locutor, o sistema apresenta precisão de, aproximadamente, 98%, porém deve-se levar em conta que as etapas de treinamento e validação foram realizadas com a mesma base de dados, e que a precisão foi obtida sobre os resultados da validação. Em outras simulações, com diferentes dados, essa taxa de precisão deve cair.

Tomando então a mesma RNC como base, bem como o princípio de realizar a análise do sinal de voz a partir do seu espectro em frequência, foram realizadas algumas alterações no código para criar um sistema de estimação de faixa etária capaz de trabalhar com a mesma base aGender da Deutsche Telekom Laboratories.

É importante lembrar que o corpus aGender é dividido em 7 classes: crianças (sem separação de gênero), jovem/masculino, jovem/feminino, adulto/masculino, adulto/feminino, sênior/masculino, sênior/feminino.

Para que fosse possível realizar ensaios e simulações iniciais de forma mais simples, o programa criado, e utilizado para o início da pesquisa, considera apenas 6 classes, portanto os resultados disponibilizados nessa dissertação, não consideram a classe "1"*crianças*. Vale ressaltar no entanto, que no decorrer dos trabalhos de pesquisa o código será melhorado para que as 7 classes sejam processadas pela RNC.

Na sequência, são apresentados alguns trechos do programa criado.

Figura 1 – Código em Python: declaração das bibliotecas e variáveis relevantes

```
import os
import pandas as pd
import numpy as np
import tensorflow as tf
from tensorflow import keras


DATASET_ROOT = os.path.join(os.path.expanduser("~"), '''Documents/Mestrado/tra
                            balho/redeNeural/agender_distribut
                            ion/''')
VALID_SPLIT = 0.1  # Percentage of samples to use for validation
SAMPLING_RATE = 16000
SHUFFLE_SEED = 43
BATCH_SIZE = 128
EPOCHS = 100
```

Trecho do código que mostra todas as bibliotecas utilizadas no código atual, e parâmetros importantes para a modelagem e treinamento da RNC.

Figura 2 – Código em Python: função audio_to_fft

```
def audio_to_fft(audio):
    # Since tf.signal.fft applies FFT on the innermost dimension,
    # we need to squeeze the dimensions and then expand them again
    # after FFT
    audio = tf.squeeze(audio, axis=-1)
    fft = tf.signal.fft(tf.cast(tf.complex(real=audio, imag=tf.zeros_like(audio
                                                                          )
                                           ), tf.complex64))
    fft = tf.expand_dims(fft, axis=-1)
    # Return the absolute value of the first half of the FFT
    # which represents the positive frequencies
    return tf.math.abs(fft[:, : (audio.shape[1] // 2), :])
```

Função para obtenção da FFT de cada arquivo de áudio a ser submetido à RNC.

Figura 3 – Código em Python: preparação dos dados para treinamento do modelo

```
# Transform audio wave to the frequency domain using `audio_to_fft`
train_ds = train_ds.map(
    lambda x, y: (audio_to_fft(x), y),
    num_parallel_calls=tf.data.experimental.AUTOTUNE)
valid_ds = valid_ds.map(
    lambda x, y: (audio_to_fft(x), y),
    num_parallel_calls=tf.data.experimental.AUTOTUNE)
train_ds = train_ds.prefetch(tf.data.experimental.AUTOTUNE)
valid_ds = valid_ds.prefetch(tf.data.experimental.AUTOTUNE)
```

Trecho do código onde os dados de treinamento e validação são transformados de áudio para frequência.

Figura 4 – Código em Python: definição do modelo

```
# MODEL DEFINITION


def residual_block(x, filters, conv_num=3, activation="relu"):
    # Shortcut
    s = keras.layers.Conv1D(filters, 1, padding="same")(x)
    for i in range(conv_num - 1):
        x = keras.layers.Conv1D(filters, 3, padding="same")(x)
        x = keras.layers.Activation(activation)(x)
    x = keras.layers.Conv1D(filters, 3, padding="same")(x)
    x = keras.layers.Add()([x, s])
    x = keras.layers.Activation(activation)(x)
    return keras.layers.MaxPool1D(pool_size=2, strides=2)(x)


def build_model(input_shape, num_classes):
    inputs = keras.layers.Input(shape=input_shape, name="input")

    x = residual_block(inputs, 16, 2)
    x = residual_block(x, 32, 2)
    x = residual_block(x, 64, 3)
    x = residual_block(x, 128, 3)
    x = residual_block(x, 128, 3)

    x = keras.layers.AveragePooling1D(pool_size=3, strides=3)(x)
    x = keras.layers.Flatten()(x)
    x = keras.layers.Dense(256, activation="relu")(x)
    x = keras.layers.Dense(128, activation="relu")(x)

    outputs = keras.layers.Dense(num_classes,
                                 activation="softmax", name="output")(x)

    return keras.models.Model(inputs=inputs, outputs=outputs)


model = build_model((SAMPLING_RATE // 2, 1), len(class_labels))

model.summary()

# Compile the model using Adam's default learning rate
model.compile(
    optimizer="Adam", loss="sparse_categorical_crossentropy",
    metrics=["accuracy"])
```

Funções e trechos do código que modelam a rede neural e seus parâmetros.

Figura 5 – Código em Python: treinamento e precisão do modelo

```
# TRAINING

history = model.fit(
    train_ds,
    epochs=EPOCHS,
    validation_data=valid_ds,
    callbacks=[earlystopping_cb, mdlcheckpoint_cb],
)


# Model 's precision
print(model.evaluate(valid_ds))
```

Trecho do código que executa o treinamento e mede a precisão do modelo.

É necessário falar mais detalhadamente sobre os parâmetros da rede, como as funções softmax, relu, optimer/Adam, loss/crossentropy?

Para treinamento do modelo foram utilizados 28120 arquivos de áudio da base de treino do corpus aGender, convertidos para o formato .wav. Além disso, conforme mencionado anteriormente, removeu-se ainda uma classe (crianças) dos dados para realizar as simulações iniciais. Dessa base, 90% dos dados foram utilizados para treinamento do modelo e 10% para validação. Utilizando um computador com as especificações indicadas na sequência, o modelo leva 4 horas para ser treinado. Considerando que está sendo utilizada a linguagem de programação Python, que não é compilada mas baseia-se em um interpretador, poderíamos esperar que esse tempo de treinamento diminuísse caso fosse utilizada outra linguagem como C, por exemplo. Mais pesquisas devem ser realizadas para avaliar se esse tempo obtido com o programa atual, pode ser considerado dentro dos padrões pela literatura (considerando a quantidade de dados e outros fatores).

- Processador: Intel(R) Core(TM) i7-8565U | CPU @ 1.80GHz
- Memória RAM: 8GB (DDR3)
- Armazenamento: 256GB (SSD)
- Placa gráfica: nVidia GM108M [GeForce MX110] | 0.97 - 0.99 GHz (velocidade CPU) / 1800 MHz (velocidade memória DDR3)

Finalizada a etapa de treinamento, é possível obter a precisão do modelo, que neste caso foi de 59.82%.

O planejamento para a sequência do trabalho, é estudar todos os aspectos e parâmetros desse sistema de forma a buscar formas de otimizar o desempenho do mesmo.

Figura 6 – Precisão obtida

```
print(model.evaluate(valid_ds))

88/88 [==============================] - 24s 258ms/step - loss: 1.0513 - accuracy: 0.5982
[1.0513083934783936, 0.5981507897377014]
```

Precisão obtida ao fim da etapa de treinamento da rede.

3.3 Sistema de Estimação Automática de Faixa Etária com GMM

O trabalho elaborado pelo Prof. Ivandro Sanches (2019) (FAPESP: 2016/18700-7), utiliza GMM e i-vector como técnicas de aprendizado de máquina. Outra diferença em relação ao trabalho proposto nessa dissertação, é que o sistema foi implementado em linguagem C, diferentemente do Python que está sendo utilizado na nova pesquisa.

Utiliza-se ainda a estimação de *pitch*, através do algoritmo de Viterbi, para conferir maior robustez ao sistema. O pitch é a frequência de vibração das cordas vocais durante a produção da voz, e funciona como parâmetro biométrico pois seu valor médio é função, por exemplo, do gênero e idade do indivíduo. Ao agregar o valor do pitch ao vetor de dados a ser processado, tem-se dados mais robustos para reforçar o padrão que precisa ser definido.

Nesse trabalho, foram realizados três tipos de experimentos conforme descrito a seguir:

- Exp 01: todo o sinal de voz de cada arquivo de áudio será processado e usado nos processos de treino, adaptação e teste
- Exp 02: apenas os quadros em que ocorre vibração das cordas vocais de cada arquivo de áudio serão processados.
- Exp 03: apenas os quadros em que ocorre vibração das cordas vocais de cada arquivo de áudio serão processados e, adicionalmente, o valor estimado de pitch será adicionado ao vetor de padrões do quadro correspondente.

A seguir, serão compartilhados alguns dos resultados obtidos. Os valores indicados, se referem a ensaios realizados com base de treinamento e teste distintas, que ao todo somam **20492** arquivos de áudio.

	GMM	i-vector
Exp 01	41.4	46.4
Exp 02	44.5	46.4
Exp 03	48.0	47.8

Tabela 3 – Resultados globais em porcentagem de acerto.

		estimado						
		1	2	3	4	5	6	7
	1	**1354**	467	155	192	29	122	69
	2	510	**1398**	24	544	10	303	14
	3	4	12	**949**	33	583	65	454
real	4	155	944	48	**1263**	19	907	13
	5	2	4	681	21	**901**	62	895
	6	221	559	32	947	17	**1663**	37
	7	7	4	400	42	920	120	**2317**

Tabela 4 – Matriz de confusão referente ao Exp 03, considerando GMM.

Considerando apenas os quatro grupos etários (crianças, jovens, adultos, sêniores), a taxa de acerto obtida foi de 49.2%.

		estimado			
		criança	jovem	adulto	sênior
real	criança	**1354**	622	221	191
	jovem	514	**2383**	1170	836
	adulto	157	1677	**2204**	1877
	sênior	228	995	1926	**4137**

Tabela 5 – Matriz de confusão referente ao Exp 03, considerando GMM e quatro grupos etários.

Somatória dos valores da diagonal principal: 10078.

$$\frac{10078}{20492} x 100\% = 49.2\%.$$

Considerando apenas três grupos (crianças, homens e mulheres), a taxa de acerto obtida foi de 87.8%.

		estimado		
		crianças	mulheres	homens
real	crianças	**1354**	78	253
	mulheres	886	**8528**	214
	homens	13	363	**8100**

Tabela 6 – Matriz de confusão referente ao Exp 03, considerando GMM e três grupos.

Somatória dos valores da diagonal principal: 17982.

$$\frac{17982}{20492} x 100\% = 87.8\%.$$

Pretende-se realizar ensaios similares e adicionais com o sistema utilizando a RNC e então comparar os resultados para posterior discussão.

3.4 Métricas

Para avaliação de desempenho do sistema, deverão ser utilizados os parâmetros *accuracy*, *precision*, *recall* e *F1-score*. Esses parâmetros podem ser calculados através da quantidade de positivos verdadeiros (TP, true positives), negativos verdadeiros (TN, true negatives), negativos falsos (FN, false negatives) e positivos falsos (FP, false positives). De qualquer forma, as API's e bibliotecas Python que serão utilizadas possuem ferramentas que possibilitam a obtenção dessas métricas de forma imediata.

3.5 Desenvolvimento do Trabalho

3.5.1 Desenvolvimento do código (A1)

O código atual da rede neural, deve ser desenvolvido para trabalhar com as sete classes disponíveis no corpus aGender. Feito isso, ensaios e simulações serão executados para que se possa comparar com os resultados obtidos através da técnica de GMM. Nessa etapa, as simulações com a rede devem considerar arquivos distintos nas bases de treino e teste, diferentemente dos ensaios que foram realizados até o momento.

3.5.2 Adição de ruído externo (A2)

A etapa seguinte, consistirá em adicionar artificialmente ruído aos sinais de áudio para que a rede tenha mais dados para processar. Nesse ponto, espera-se iniciar uma investigação quanto a influência do tamanho do volume de dados no desempenho desse tipo de técnica de aprendizado profundo com sinais acústicos.

3.5.3 Busca por otimizações (A3)

Conforme progresso das atividades e obtenção de dados mais concretos quanto a capacidade da RNC para esse tipo de problema (estimação de faixa etária), planeja-se definir as limitações que por ventura tenham sido identificadas no sistema, além de pesquisar possíveis otimizações que possam ser indicadas para a aplicação que está sendo abordada nesse trabalho. Essa etapa deve ocorrer em paralelo com as demais atividades , após a banca de qualificação, e durante todo o trabalho de pesquisa.

4 CRONOGRAMA

A Tabela 7 apresenta o cronograma de execução das atividades desta proposta.

Tabela 7 – Cronograma das atividades previstas

Etapa	Meses											
	jan	fev	mar	abr	mai	jun	jul	ago	set	out	nov	dez
Revisão bibliográfica	■	■	■	■	■	■	■	■	■	■	■	■
A0		■	■	■	■							
Banca de qualificação						■						
A1							■					
A2								■				
A3							■	■	■	■	■	
A4												■

A0: Implementação do programa inicial e realização das primeiras simulações

A1: Desenvolvimento do código com melhorias.

A2: Adição de ruído externo.

A3: Busca por otimizações.

A4: Elaboração da dissertação, análise e discussão dos resultados.

REFERÊNCIAS

BADINE, F. *Speaker Recognition*. [S.l.], 2020. Disponível em: <https://keras.io/examples/audio/speaker_recognition_using_cnn/>. Citado na página 10.

BURKHARDT, F. et al. A database of age and gender annotated telephone speech. In: *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*. Valletta, Malta: European Language Resources Association (ELRA), 2010. Disponível em: <http://www.lrec-conf.org/proceedings/lrec2010/pdf/262_Paper.pdf>. Citado na página 9.

SANCHES, I. *Estimação automática de faixa etária pelo processamento do sinal de voz*. São Bernardo do Campo, 2019. 14 p. Processo FAPESP 2016/18700-7. Citado 5 vezes nas páginas 6, 7, 8, 9 e 14.