Age and Gender Classification from Speech: Exploring a Convolutional Neural Network with Multi-Attention Module with Mobile App-Based Evaluation

Luis Miguel Sanchez Pinilla

Data Science Project

May 18, 2024

Abstract

This report replicates the work of a Convolutional Neural Network with a Multi-Attention Module for speaker age and gender recognition through speech recordings. The Multi-Attention Module allows the model to focus on critical speech features like pitch and timing variations, leading to improved speaker characteristic differentiation. The model achieved some success in differentiating speaker characteristics, with general accuracy reaching around 66%. However, the model was over-fitting and it performed best in classifying older age groups from fifties and above, regardless of gender. The accuracy dropped significantly for younger demographics like teens, twenties, and thirties, where the model performs best on women than men. An app was developed to test the model's performance in real-world settings.

Keywords: convolutional neural network, multi-attention module, age and gender recognition, speech spectrogram, mobile application.

1 Introduction

Human speech is a powerful tool for communication, conveying not only the content of our words but also information about the speaker's identity, emotions, age, and even gender. This information is becoming increasingly valuable in the field of Human-Computer Interaction, where speech signals are used as primary input for various applications. These applications include Automatic Speech Recognition (Park et al. 2020), Speech Emotion Recognition (Anvarjon, Mustaqeem, and Kwon 2020), and age and gender estimation (Ghahremani et al. 2018) (Sánchez-Hevia et al. 2019). Recently, there has been a surge of interest in automatically extracting this speaker information from speech signals.

Accurately and efficiently identifying a speaker's age and gender from their voice unlocks a range of potential applications. For instance, targeted advertising could be delivered based on a customer's age and gender. Call centers could pair callers with agents more suited to their needs based on the caller's identity. Additionally, recognizing a speaker's gender can improve the performance of speaker recognition systems by reducing the search space within databases. Age recognition allows voice-controlled systems to adapt to users and provide a more natural human-machine interaction experience.

Despite ongoing research, accurately recognizing age and gender from speech signals remains a challenge. Two key hurdles exist: selecting the most informative features from speech data and designing effective classification models. Traditional machine learning algorithms can struggle with high-dimensional feature sets, leading to decreased accuracy. Additionally, existing methods like i-vectors and x-vectors, which rely on acoustic features, may not provide optimal results, especially for age classification (Lortie et al. 2015) (Landge, Deshmukh, and Shrishrimal 2015). While alternative approaches exist, achieving high performance often requires substantial domain expertise and can be computationally demanding. This highlights the need for a Data Science approach that explores various feature engineering techniques and classification algorithms to optimize age and gender recognition from speech signals.

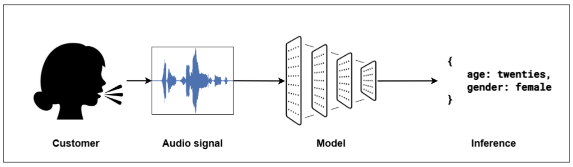


Figure 1: Algorithms to classify speakers into age and gender groups may be applied in call centres with Interactive Voice Response systems

Existing Interactive Voice Response systems typically leverage a two-step approach for automatic caller intent classification. The first step involves continuous speech recognition, which transcribes spoken language into text format (Bhat et al. 2013). Following this, semantic text classification techniques categorize the transcribed text into predefined call intents. Additionally, these systems can extract biometric and emotional features from the speech signal as in Figure 1. This combined approach, leveraging both lexical and paralinguistic information, aims to create a more natural and human-like experience during human-machine interaction.

2 Problem Formulation

Tursunov et al. 2021 proposes a novel approach to speaker age and gender classification based on speech audio signals by introducing a Convolutional Neural Network with a specially designed Multi-Attention Module. The model utilizes separate attention mechanisms for time and frequency, allowing it to capture temporal cues and relevant spectral information. The authors tested their approach on two datasets, achieving high accuracy scores (up to 97%) for both datasets in tasks like gender recognition and age-gender classification.

2.1 Inputs and Outputs

The age and gender recognition system hinges on audio recordings of human voices captured by a standard audio recorder. These recordings, containing the raw vocal data, serve as the system's input for analysis.

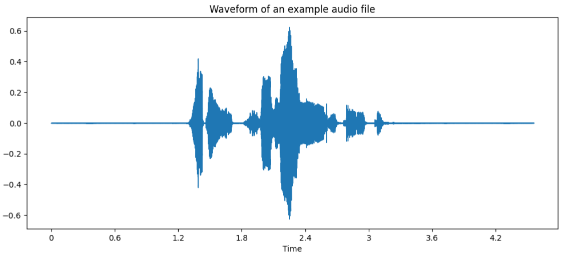


Figure 2: A waveform is like a sound's heartbeat, with ups and downs showing how pressure changes over time. The bigger the wave, the louder the sound..

The speech spectrogram contains significant information regarding speaker age, gender, speaking style, emotional content, etc. One of the reasons for the importance of the speech spectrograms is that a human also processes the sounds in the form of different frequencies over time in the ear (Carlyon 2004).

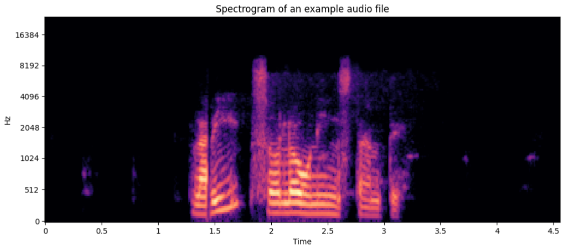


Figure 3: A spectrogram is like a fingerprint for sound, showing how frequency changes over time with color representing loudness.

As seen in Figure 3, for spectrograms, time is on the horizontal axis, while the vertical axis represents the frequency, and the yellow color of each point in the graph corresponds to an amplitude of a certain frequency at a particular time.

The ultimate goal of the age and gender recognition system is to assign labels to the analyzed voices. These labels categorize the speaker's gender, such as male or female, and their age group. An example of a label would be the tuple:

$$gender, age = (female, twenties) \tag{1}$$

2.2 Data acquisition process

This project utilizes the publicly available Common Voice Corpus 17.0 from the Mozilla Foundation. It provides a vast and continuously expanding collection, currently exceeding 20,408 validated hours of recordings across 124 languages. The project welcomes user contributions of their voices and allows for language requests through the dedicated Languages page. By leveraging this publicly available resource, the study ensures transparency and ethical data acquisition for the task of age and gender recognition from speech signals.

2.3 Samples description

The dataset consists of unique MP3 audio files paired with corresponding text files. Significantly, over 60% of the dataset's 31,175 recorded hours include valuable demographic metadata such as age, sex, and speaker accent. Specifically, the Spanish recordings within Common Voice Corpus 17.0 consist of 336,846 samples.

Each sample in the dataset is represented as a dictionary containing features such as: locale, and segment. The labels would be the age and gender columns of the dataset.

locale	audio	age	gender
es	{array, sr}	twenties	female
en	{array, sr}	thirties	
it	{array, sr}	forties	male

Table 1: Samples from the dataset. The column audio is an dictionary containing *array*, the amplitudes of the audio signal, and *sr*, the sample rate of the recording.

2.4 Problem characterization

The initial step in this project involves characterizing the machine learning task at hand. The goal is to predict speaker characteristics, namely age group and gender, based on audio recordings. This task falls under the category of classification.

Classification problems are designed to predict discrete labels or categories for data points. In this case, the data points are the audio recordings, and the labels we're interested in are the speaker's age group and gender.

The rationale behind using classification lies in the nature of the predicted labels. Age group and gender are not continuous values, but rather represent distinct categories with predefined boundaries. The model won't be estimating a specific age in years or a degree of masculinity/femininity on a spectrum.

Instead, the model will leverage information extracted from the audio signal. This information is typically visualized as a spectrogram, which offers a detailed representation of the sound's frequency and intensity over time. By analyzing the spectrogram, the model can extract relevant features such as pitch, resonance patterns, and speech rhythm that might be indicative of speaker characteristics. These features then serve as input to the classification model, which predicts the most likely categories for age group and gender.

3 Solution Approach

3.1 Data processing

In preparing the data for training, the speaker identification, spoken text, and any other irrelevant information were removed first. This initial cleaning ensured the model focused on the relevant to age and gender prediction, the audio signal, so that the remaining columns for each sample would be $(audio, age, gender)$.

Next, the labels for age and gender underwent a validation process. Any entries that didn't conform to the categories defined in the equation 2 and equation 3 were excluded to maintain data integrity.

$$genders = \{female, male\} \tag{2}$$

$$\begin{aligned} age\ groups = \{&teens, twenties, thirties, \\ &fourties, fifties, sixties\} \end{aligned} \tag{3}$$

The data then went through a transformation stage. Here, the age and gender columns were combined into a single label, which results in the 12 labels present in equation 4.

$$\begin{aligned} labels = \{&F\_teens, F\_twenties, F\_thirties, \\ &F\_fourties, F\_fifties, F\_sixties, \\ &M\_teens', M\_twenties, M\_thirties, \\ &M\_fourties, M\_fifties, M\_sixties\} \end{aligned} \tag{4}$$

Following this, the audio data itself was processed. The recordings were filtered out so that any recordings exceeding 10 seconds were excluded. This ensures the model trains on audio snippets that fall within a manageable timeframe.

The filtered samples then are shuffled and 500 samples are selected randomly for every label. This ensures the model is trained on a representative sample of the population. The result is a dataset of $6,000$ samples. This is significantly lower than the original $300,000$ samples, however, this dataset size is chosen because of memory constraints. This dataset is furthered split into training, validation, and testing sets with a proportion of $(60\%, 20\%, 20\%)$.

The column *audio* is transformed into a column *spectrogram*, containing an RGB image of the audio signal as a 3D-array of dimensions $(240 \times 240 \times 3)$. The Short-time Fourier transform is applied to a discrete signal to generate a spectrogram of the speech signal. The speech signal is divided into short overlapping segments of equal length, and then a fast Fourier transform is applied to each frame to compute its spectrum.

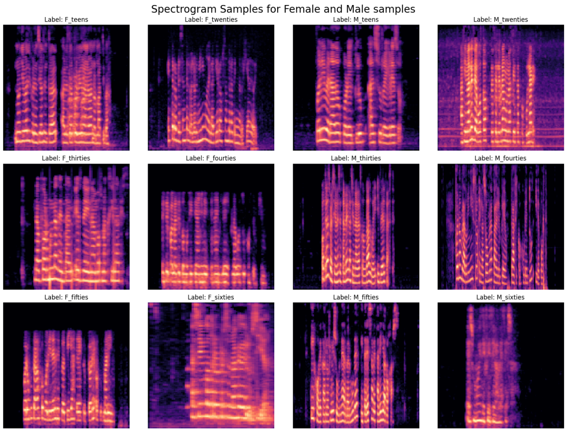


Figure 4: A sample and its corresponding spectrogram is displayed for each existing label.

For a given spectrogram S, the strength of a given frequency attribute f at a given time t is represented by the darkness or color of the corresponding point $S(t, f)$ as shown in Figure 4. The end result are samples with the form of $(spectrogram, label)$.

3.2 Models selection

Current state-of-the-art results in the field of computer vision are achieved by utilizing the power of Convolutional Neural Networks for different tasks, such as image classification (Tan and Le 2019), automatic speech recognition (Passricha and Aggarwal 2019), and speech emotion recognition (Kwon et al. 2021). The evidence presented in Tursunov et al. 2021 demonstrate that Convolutional Neural Networks perform really well at the task of age and gender recognition on audio recordings.

The selected model is a Convolutional Neural Network with Multi-Attention Module that consists of three blocks: Feature Learning Block, Multi-Attention Module, and a Fully Connected Network as shown in Figure 5. It is important to consider that since spectrogram of the speech signal represents temporal and spatial information, it becomes necessary to capture all the valuable features of the samples to achieve high performance. However, capturing temporal cues using exclusively convolutional neural networks is difficult.

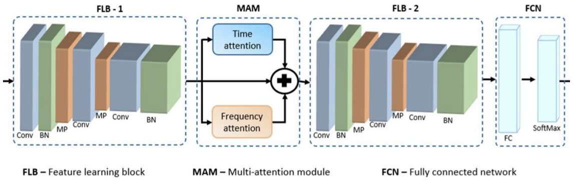


Figure 5: CNN with Multi-Attention Module architecture diagram extracted from Tursunov et al. 2021.

The utilization of attention mechanisms in deep learning models has demonstrated its strength in model performance and robustness in solving several different tasks, such as sound event detection (Miyazaki et al. 2020) and speech emotion recognition (Kwon et al. 2021). The working mechanism of attention is to select essential features to the target by focusing on the extracted features. Feature maps generated using convolutional neural networks from speech spectrograms contain information regarding particular regions of the input data.

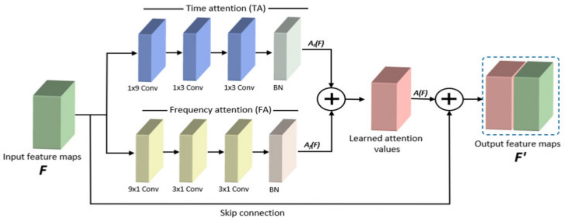


Figure 6: Multi-Attention Module architecture diagram extracted from Tursunov et al. 2021.

The attention module proposed in Tursunov et al. 2021, displayed in Figure 6, uses a two-stream, time and frequency attention mechanisms separately. This Multi-Attention Module focuses on extracting key features from both the time and frequency domains within the speech data.

4 Experimentation

The experiments were conducted on a server ran Ubuntu 22.04 and Python 3.9. Tensorflow version 2.1.6 handled the computations. To accelerate these computations, the server leveraged a NVIDIA GTX 1050ti graphics card with CUDA support. The system was further equipped with an Intel i5 4770k processor and 16 GB of RAM. Speech spectrograms were generated using the librosa Python library for audio and music analysis.

The training, validation, and test sets are split into batches of 32 samples. The samples are shuffled again only on the training and validation sets.

All of the models tested have an initial layer of preprocessing that transforms the RGB image into a grey scale image, resizes it to a 64 pixels square, and normalizes the pixels values into a $[0, 1]$ range. The final result is a features column with the size of $(64 \times 64 \times 1)$.

The model was instantiated with two different configurations. The first configuration is shown on Table 2, and was trained on 120 epochs with steps 112 each, as well as 37 steps of validation.

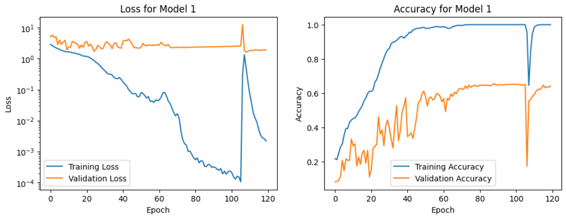


Figure 7: Performance of the first model in terms of its accuracy and loss in both validation and training. This configuration presents over-fitting.

The Figure 7 indicates that the first model configuration presents over-fitting. It finishes with an accuracy of 100% in training, but holds an accuracy of 64.13% in validation. Around the first 8 epochs the loss in validation starts to converge as the loss in training keeps decreasing. After the first 70 epochs, the accuracy on validation and training converges. This results indicate that the accuracy will not improve significantly in further epochs.

For the second configuration there were two considerations: remove convolutional layers in order to reduce the complexity of the model and increase batch size. While there were some convolutional layers removed on the second configuration as seen in Table 3, increasing the batch size resulted in the computer running out of memory. Therefore the batch size of 32 was kept. This configuration was trained on 120 epochs with steps 112 each, as well as 37 steps of validation.

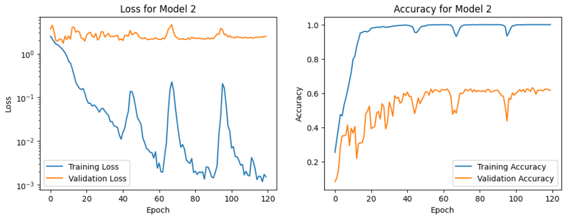


Figure 8: Just like the first configuration, the second model configuration also presents over-fitting as well as its accuracy seems to have already converged.

The Figure 8 indicates that the second model configuration also presents over-fitting. It finishes with an accuracy of 100% in training, but holds an accuracy of 61.73% in validation. After the first 40 epochs the loss in validation starts to converge as the loss in training keeps decreasing. In contrast to the first configuration, the accuracy on training converges after the first 10 epochs, but it takes until the first 60 epochs to converge on validation. This results indicate that the accuracy will not improve significantly in further epochs.

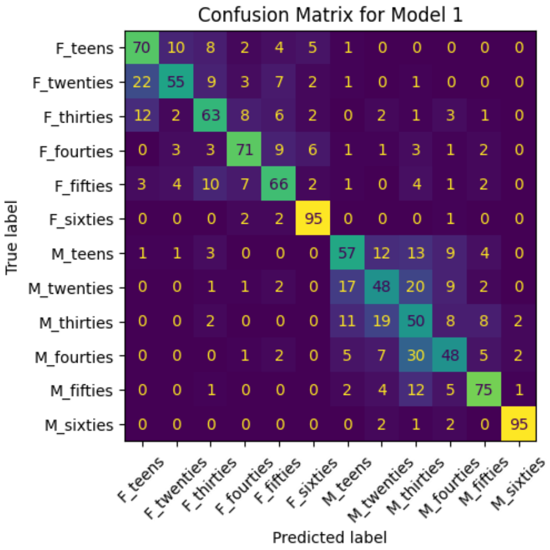
Table 2: Model Configuration 1

Layer Name	**Output Tensor**	**Kernel Size**	**Stride**	**Activation**	**Parameters**
Preprocessing					
ToGreyscale	$240 \times 240 \times 1$	-	-	-	0
Resizing	$64 \times 64 \times 1$	-	-	-	0
Rescaling	$64 \times 64 \times 1$	-	-	-	0
FLB					
Conv2D	$32 \times 32 \times 120$	9×9	2×2	ReLU	$9,840$
BatchNormalization	$32 \times 32 \times 120$	-	-	-	480
MaxPooling2D	$16 \times 16 \times 120$	2×2	2×2	-	0
Conv2D	$16 \times 16 \times 256$	5×5	1×1	ReLU	$768,256$
MaxPooling2D	$8 \times 8 \times 256$	2×2	1×1	-	0
Conv2D	$8 \times 8 \times 384$	3×3	1×1	ReLU	$885,120$
BatchNormalization	$8 \times 8 \times 384$	-	-	-	$1,536$
MAM					
Conv2D	$8 \times 8 \times 64$	1×9	1×1	ReLU	$221,248$
Conv2D	$8 \times 8 \times 64$	1×3	1×1	ReLU	$12,352$
Conv2D	$8 \times 8 \times 64$	1×3	1×1	ReLU	$12,352$
BatchNormalization	$8 \times 8 \times 64$	-	-	-	256
Conv2D	$8 \times 8 \times 64$	9×1	1×1	ReLU	$221,248$
Conv2D	$8 \times 8 \times 64$	3×1	1×1	ReLU	$12,352$
Conv2D	$8 \times 8 \times 64$	3×1	1×1	ReLU	$12,352$
BatchNormalization	$8 \times 8 \times 64$	-	-	-	256
Concatenate	$8 \times 8 \times 128$	-	-	-	0
BatchNormalization	$8 \times 8 \times 128$	-	-	-	512
Concatenate	$8 \times 8 \times 512$	-	-	-	0
FCN					
Flatten	384	-	-	-	0
Dense	80	-	-	ReLU	$30,800$
BatchNormalization	80	-	-	-	320
Dropout	80	-	-	-	0
Dense	12	-	-	SoftMax	972
Total params = $8,822,404$					
Total trainable params = $8,819,716$					

Table 3: Model Configuration 2 (only blocks that changed)

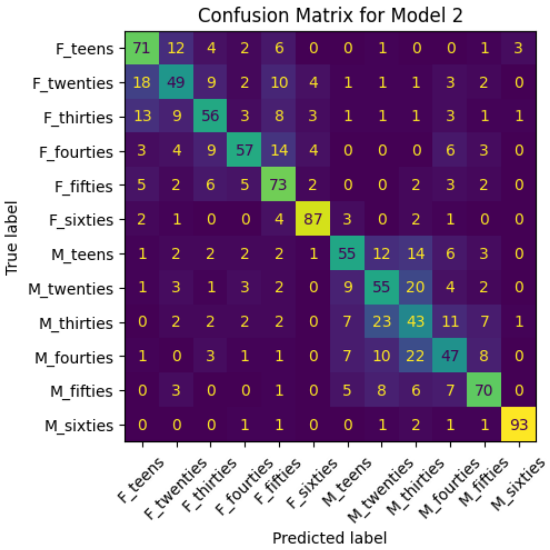
Layer Name	**Output Tensor**	**Kernel Size**	**Stride**	**Activation**	**Parameters**
FLB					
Conv2D	$32 \times 32 \times 120$	9×9	2×2	ReLU	$9,840$
BatchNormalization	$32 \times 32 \times 120$	-	-	-	480
MaxPooling2D	$16 \times 16 \times 120$	2×2	2×2	-	0
MAM					
Conv2D	$8 \times 8 \times 64$	1×9	1×1	ReLU	$69,184$
Conv2D	$8 \times 8 \times 64$	1×3	1×1	ReLU	$12,352$
BatchNormalization	$8 \times 8 \times 64$	-	-	-	256
Conv2D	$8 \times 8 \times 64$	9×1	1×1	ReLU	$69,184$
Conv2D	$8 \times 8 \times 64$	3×1	1×1	ReLU	$12,352$
BatchNormalization	$8 \times 8 \times 64$	-	-	-	256
Concatenate	$8 \times 8 \times 128$	-	-	-	0
BatchNormalization	$8 \times 8 \times 128$	-	-	-	512
Concatenate	$8 \times 8 \times 512$	-	-	-	0
Total params = $2,740,548$					
Total trainable params = $2,739,396$					

5 Results and Applications



	F_teens	F_twenties	F_thirties	F_fourties	F_fifties	F_sixties	M_teens	M_twenties	M_thirties	M_fourties	M_fifties	M_sixties
F_teens	70	10	8	2	4	5	1	0	0	0	0	0
F_twenties	22	55	9	3	7	2	1	0	1	0	0	0
F_thirties	12	2	63	8	6	2	0	2	1	3	1	0
F_fourties	0	3	3	71	9	6	1	1	3	1	2	0
F_fifties	3	4	10	7	66	2	1	0	4	1	2	0
F_sixties	0	0	0	2	2	95	0	0	0	1	0	0
M_teens	1	1	3	0	0	0	57	12	13	9	4	0
M_twenties	0	0	1	1	2	0	17	48	20	9	2	0
M_thirties	0	0	2	0	0	0	11	19	50	8	8	2
M_fourties	0	0	0	1	2	0	5	7	30	48	5	2
M_fifties	0	0	1	0	0	0	2	4	12	5	75	1
M_sixties	0	0	0	0	0	0	0	2	1	2	0	95

Figure 9: Confusion Matrix for Model 1 evaluated with the testing dataset. This model does better on average classifying women.



	F_teens	F_twenties	F_thirties	F_fourties	F_fifties	F_sixties	M_teens	M_twenties	M_thirties	M_fourties	M_fifties	M_sixties
F_teens	71	12	4	2	6	0	0	1	0	0	1	3
F_twenties	18	49	9	2	10	4	1	1	1	3	2	0
F_thirties	13	9	56	3	8	3	1	1	1	3	1	1
F_fourties	3	4	9	57	14	4	0	0	0	6	3	0
F_fifties	5	2	6	5	73	2	0	0	2	3	2	0
F_sixties	2	1	0	0	4	87	3	0	2	1	0	0
M_teens	1	2	2	2	2	1	55	12	14	6	3	0
M_twenties	1	3	1	3	2	0	9	55	20	4	2	0
M_thirties	0	2	2	2	2	0	7	23	43	11	7	1
M_fourties	1	0	3	1	1	0	7	10	22	47	8	0
M_fifties	0	3	0	0	1	0	5	8	6	7	70	0
M_sixties	0	0	0	1	1	0	0	1	2	1	1	93

Figure 10: Confusion Matrix for Model 2 evaluated with the testing dataset. Similar performance to that of Model 1.

The first model configuration holds an accuracy of the 66.08% and a loss of the 178% in testing, so this configuration did slightly better in testing that validation while presenting over-fitting. In the Table 4, the macro average precision and recall, which represent the unweighted mean across all classes, are both 66%. This indicates that the model performs reasonably well on average. However, there's a clear distinction in performance between age groups. The model excels at classifying the older age groups *F_sixties* and *M_sixties*, achieving a precision above 85% and a recall close to 95%. In contrast, the younger age groups *F_teens*, *M_teens*, *F_twenties*, *M_twenties*, *M_thirties* show more balanced but less outstanding metrics.

Table 4: Classification Report for Model 1

Class	Precision	Recall	F1-Score
F_teens	0.65	0.70	0.67
F_twenties	0.73	0.55	0.63
F_thirties	0.63	0.63	0.63
F_fourties	0.75	0.71	0.73
F_fifties	0.67	0.66	0.67
F_sixties	0.85	0.95	0.90
M_teens	0.59	0.57	0.58
M_twenties	0.51	0.48	0.49
M_thirties	0.37	0.50	0.43
M_fourties	0.55	0.48	0.51
M_fifties	0.76	0.75	0.75
M_sixties	0.95	0.95	0.95
accuracy			0.66
macro avg	0.67	0.66	0.66
weighted avg	0.67	0.66	0.66

Table 5: Classification Report for Model 2

Class	Precision	Recall	F1-Score
F_teens	0.62	0.71	0.66
F_twenties	0.56	0.49	0.52
F_thirties	0.61	0.56	0.58
F_fourties	0.73	0.57	0.64
F_fifties	0.59	0.73	0.65
F_sixties	0.86	0.87	0.87
M_teens	0.62	0.55	0.59
M_twenties	0.49	0.55	0.52
M_thirties	0.38	0.43	0.40
M_fourties	0.51	0.47	0.49
M_fifties	0.70	0.70	0.70
M_sixties	0.95	0.93	0.94
accuracy			0.63
macro avg	0.64	0.63	0.63
weighted avg	0.64	0.63	0.63

There's a trade-off between precision, correctly identifying a high proportion of predictions, and recall, identifying most of the actual cases, in these younger classes. This suggests the model is making some false positive classifications on the younger classes.

The confusion matrix in the Figure 9 reveals the model might be struggling to distinguish between younger age groups *F_teens*, *F_twenties*, *M_teens*, *M_twenties*. There are scattered patterns throughout these rows, with many misclassified instances with their neighbors. This confirms the lower precision and recall observed in the classification report for these classes. It can also be noted that the model has a worse performance identifying the younger male labels than with the younger female labels. Although this bad performance is not present when classifying the labels *M_fifties* and *M_sixties*, where it can actually do slightly better than with the *F_fifties* and *F_sixties*.

Overall the second configuration has similar performance of that of the first configuration. The second model configuration holds an accuracy of the 63% and a loss of the 255.64% in testing. The macro average precision and recall are both 63% which means the performance on both configurations is very similar and both perform reasonably well on average.

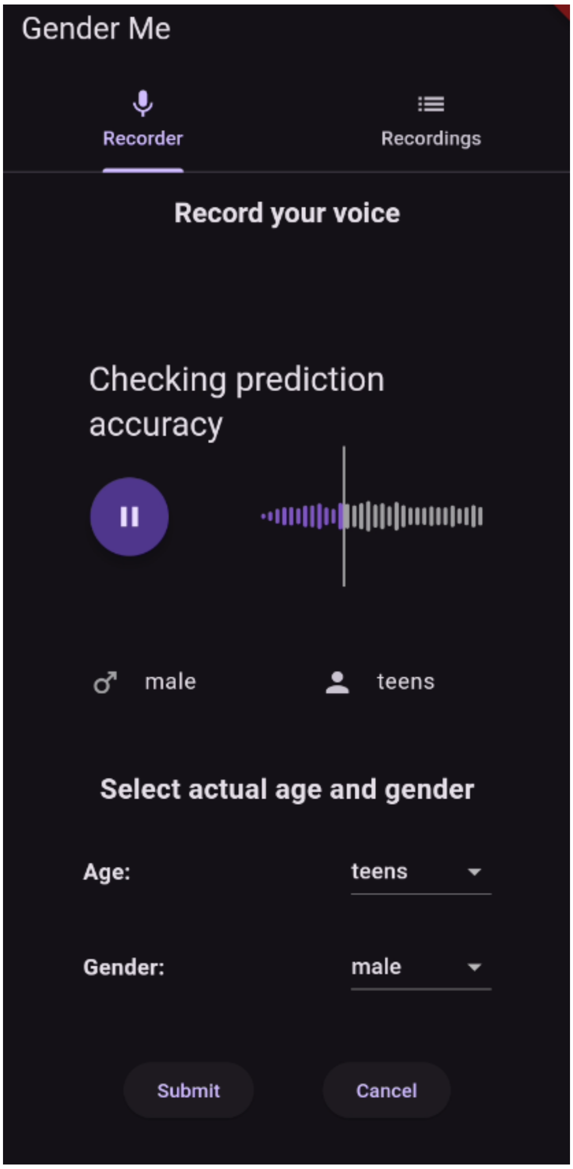
There's a clear distinction in performance between age groups, and just like the first configuration, both models excel at classifying the older age groups $F\_sixties$ and $M\_sixties$, the second one achieving a precision above 87% and a recall close to 93% as shown in Table 5. Also, just like in the first configuration, there's a trade-off between precision and recall in these younger classes, suggesting that the second configuration is making some false positive classifications on the younger classes.

The confusion matrix in the Figure 10 reveals the second configuration is struggling to distinguish between younger age groups F_teens, $F\_twenties$, M_teens, $M\_twenties$, just like on the first configuration. It also has a worse performance identifying the younger male labels than with the younger female labels, except with the labels $M\_fifties$ and $M\_sixties$, where it can actually do slightly better than with the $F\_fifties$ and $F\_sixties$.

5.1 Real life applications

This models have various practical applications. In customer service, it can tailor interactions by automatically adjusting responses to suit different age groups or genders, thereby enhancing user experience. In security and authentication systems, it can add an extra layer of verification by matching voice characteristics with expected demographic profiles. Additionally, in healthcare, such a model can assist in diagnosing age-related conditions or monitoring vocal changes in patients. Marketing and user research can also benefit by analyzing the demographic profiles of callers to better understand and target audience segments.

In order to put this model to the test, a mobile application was developed for the Android using the Flutter framework. The app embeds the trained model from the first configuration directly within the app. Users can leverage this app to record their own voice or the voice of someone else. Once a recording is captured, the embedded model computes the inference on the audio data and generates a prediction about the speaker's gender and age group, see Figure 11. The app then presents this prediction to the user. The user is prompted to select the actual gender and age group with the goal of keeping a record of the model performance. This information is all stored within a local database on the user's phone as pictured in Figure ??.



Record your voice

Checking prediction accuracy

♂ male

teens

Select actual age and gender

Age: teens

Gender: male

Figure 11: The app makes a prediction and checks the accuracy.

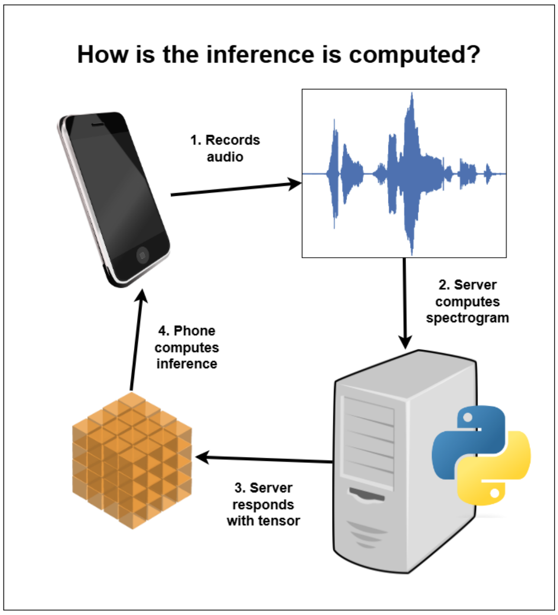

How is the inference is computed?
1. Records audio
2. Server computes spectrogram
3. Server responds with tensor
4. Phone computes inference

Figure 12: A server offloads tasks from the app.

In this setup, we can leverage a Python server built with Flask to act as an intermediary between the phone and the classification model, as pictured in Figure 12.

The the audio recording is transmitted to the server, and then uses libraries like librosa to compute a spectrogram, essentially converting the audio into a visual representation suitable for the model. Finally, the server transforms the spectrogram into a tensor. The phone then receives this tensor and feeds it into the pre-trained classifier model for analysis and classification of age and gender. This way, the server efficiently prepares the audio data for the model, while the phone handles the actual classification task.

This speaker classification app focuses on a core function: analyzing audio recordings and predicting the speaker's age and gender. This basic functionality, however, could be a valuable building block for various applications. Developers can integrate this app as a component within a larger system to achieve the benefits we previously discussed. For instance, the app's predictions could be fed into another program that personalizes user experiences or enhances security measures. By providing this age and gender analysis, the app empowers developers to build more sophisticated systems with targeted functionalities.

6 Conclusions

The research explored replicating a Convolutional Neural Network with a special Multi-Attention Module proposed by Tursunov et al. 2021 to classify speaker age and gender from speech recordings. The model configurations proposed were not able to replicate the results exposed in the paper. In terms of accuracy, the models implemented did better for older age groups compared to younger ones. There are indications that the model is over-fitting and may not perform as well on completely new data.

An app was developed to provide a user-friendly interface for testing the model in a real-world setting. This app allows users to record their voice samples directly and give instant feedback on the model's age and gender classification attempts. The app serves as a crucial tool for researchers so they can gather new samples to train the model on.

While the previous research showed promise, there's room for improvement. Techniques to address over-fitting and using more diverse datasets are areas for further exploration. Additionally, investigating alternative neural network architectures might lead to even better performance.

References

Anvarjon, Tursunov, Mustaqeem, and Soonil Kwon (2020). "Deep-net: A lightweight CNN-based speech emotion recognition system using deep frequency features". In: *Sensors* 20.18, p. 5212.

Bhat, Chitralekha et al. (2013). "Deploying usable speech enabled ivr systems for mass use". In: *2013 International Conference on Human Computer Interactions (ICHCI)*. IEEE, pp. 1–5.

Carlyon, Robert P (2004). "How the brain separates sounds". In: *Trends in cognitive sciences* 8.10, pp. 465–471.

Ghahremani, Pegah et al. (2018). "End-to-end Deep Neural Network Age Estimation." In: *Interspeech*. Vol. 2018, pp. 277–281.

Kwon, Soonil et al. (2021). "Att-Net: Enhanced emotion recognition system using lightweight self-attention module". In: *Applied Soft Computing* 102, p. 107101.

Landge, Maheshkumar B, RR Deshmukh, and PP Shrishrimal (2015). "Analysis of variations in speech in different age groups using prosody technique". In: *International Journal of Computer Applications* 126.1.

Lortie, Catherine L et al. (2015). "Effects of age on the amplitude, frequency and perceived quality of voice". In: *Age* 37, pp. 1–24.

Miyazaki, Koichi et al. (2020). "Weakly-supervised sound event detection with self-attention". In: *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, pp. 66–70.

Park, Daniel S et al. (2020). "Improved noisy student training for automatic speech recognition". In: *arXiv preprint arXiv:2005.09629*.

Passricha, Vishal and Rajesh Kumar Aggarwal (2019). "A hybrid of deep CNN and bidirectional LSTM for automatic speech recognition". In: *Journal of Intelligent Systems* 29.1, pp. 1261–1274.

Sánchez-Hevia, Héctor A et al. (2019). "Convolutional-recurrent neural network for age and gender prediction from speech". In: *2019 Signal Processing Symposium (SPSympo)*. IEEE, pp. 242–245.

Tan, Mingxing and Quoc Le (2019). "Efficientnet: Rethinking model scaling for convolutional neural networks". In: *International conference on machine learning*. PMLR, pp. 6105–6114.

Tursunov, Anvarjon et al. (2021). "Age and gender recognition using a convolutional neural network with a specially designed multi-attention module through speech spectrograms". In: *Sensors* 21.17, p. 5892.