TALLINN UNIVERSITY OF TECHNOLOGY
Faculty of Information Technology

Leo Kristopher Piel 166770

SPEECH-BASED IDENTIFICATION OF CHILDREN'S GENDER AND AGE WITH NEURAL NETWORKS

Master's thesis

Supervisor: Tanel Alumäe
Senior Researcher

Tallinn 2018

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Leo Kristopher Piel 166770

KÕNEPÕHINE LASTE VANUSE JA SOO TUVASTAMINE NÄRVIVÕRKUDE ABIL

Magistritöö

Juhendaja: Tanel Alumäe
Vanemteadur

Tallinn 2018

Author's declaration of originality

I hereby certify that I am the sole author of this thesis. All the used materials, references to the literature and the work of others have been referred to. This thesis has not been presented for examination anywhere else.

Author: Leo Kristopher Piel

06.05.2018

Abstract

Identifying children’s age and gender based on speech can be used in different kinds of applications to improve child-computer interaction, adapt content, protect children or to improve the performance of other speech related tasks such as speech or speaker recognition. The fact that the voices of children evolve differently and males’ puberty voice mutation can happen at different times of age, makes it a challenging task. Different approaches, including machine learning methods, have been used to challenge the stated problem, but there are only few works, where neural networks have taken i-vectors or mel-frequency cepstral coefficients as input. The goal of this thesis was to analyse the performance of different types of neural networks on children’s age and gender identification based on speech and compare the results to a chosen i-vector baseline system. To this end, feedforward deep neural networks and recurrent neural networks with convolutional layers were built. A simple feedforward neural network worked best for gender identification by achieving accuracy of 92.8% on test data. For age group identification, a combination of the built models achieved the highest accuracy of 76.3%. The built models outperformed humans on both tasks by nearly 9 percentage points. Compared to the baseline method, the built ensemble model achieved 4.6 percentage points higher accuracy on age group identification on test data. A feedforward deep neural network achieved 2.6 percentage points higher accuracy on gender identification.

This thesis is written in English and is 85 pages long, including 8 chapters, 49 figures and 18 tables.

Keywords: age identification, gender identification, children, speech, deep neural networks, recurrent neural networks, convolutional neural networks, i-vectors, mel-frequency cepstral coefficients

Annotatsioon

Kõnepõhine laste vanuse ja soo tuvastamine närvivõrkude abil

Lapse hääle põhjal vanuse ja soo määramist saab kasutada näiteks lapse ja arvuti vahelise suhtluse või teiste kõnega seotud ülesannete nagu kõne- või kõnelejatuvastuse parendamiseks. suunatud sisu kuvamiseks ning ka laste kaitsmiseks. Ülesande teeb keerukaks see, et lapsed arenevad erineva kiirusega ja puberteedieas toimuv häälemurre ei toimu meestel alati samas vanuses. Antud probleemi lahendamiseks on varem kasutatud erinevaid meetodeid, muuhulgas ka masinõpet, kuid enamik neist ei ole käsitlenud närvivõrke, mis võtaksid sisendiks i-vektoreid või mel-sageduse kepstri kordajaid. Käesoleva lõputöö eesmärgiks oli analüüsida erinevat tüüpi närvivõrkude efektiivsust soo ja vanuse määramisel ning võrrelda neid aluseks võetud i-vektorite süsteemiga. Lõputöö käigus loodi pärilevi närvivõrke ja rekurrentseid närvivõrke konvolutsiooniliste kihtidega. Soo määramiseks toimis kõige paremini lihtne pärilevi närvivõrk, saavutades täpsuseks 92,8%. Kõige kõrgem täpsus 76,3% vanusegrupi määramiseks saavutati erineva struktuuriga loodud mudeleid kombineerides. Närvivõrgud saavutasid inimestest mõlemal ülesandel ligi 9 protsendipunkti parema tulemuse. Aluseks võetud i-vektoreid kasutavast meetodist saavutas vanusegrupi määramiseks kombineeritud närvivõrk testandmestikul 4.6 protsendipunkti parema tulemuse ning soo määramiseks mõeldud pärilevi närvivõrk 2.6 protsendipunkti parema tulemuse.

Lõputöö on kirjutatud inglise keeles ning sisaldab teksti 85 leheküljel, 8 peatükki, 49 joonist, 18 tabelit.

Võtmesõnad: vanuse määramine, soo määramine, lapsed, kõne, sügavad närvivõrgud, rekurrentsed närvivõrgud, konvolutsioonilised närvivõrgud, i-vektorid, mel-sageduse kepstri kordajad

List of abbreviations and terms

ANN	Artificial Neural Network
DNN	Deep Neural Network
LSTM	Long Short-Term Memory
GMM	Gaussian Mixture Model
MFCC	Mel-Frequency Cepstral Coefficient
ReLU	Rectified Linear Units
RNN	Recurrent Neural Network
UBM	Universal Background Model

Table of contents

List of figures

List of tables

1 Introduction

In addition to linguistic information such as the used words, human speech also includes paralinguistic information that considers the tone and pitch of the voice that can be used to identify the gender and age of a speaker. The goal of this research is to analyse the performance of different kinds of neural networks that could identify the gender and age of a child based on the information in human speech.

1.1 Problem Statement

Already implemented solutions for identifying the speaker's gender and age are using, for instance, Gaussian mixture models, support vector machines, multi-layer perceptron networks with sigmoid function and hidden Markov models [1], [2], [3]. In contrast, this research aims to build neural networks using i-vectors as well as mel-frequency cepstral coefficients (MFCC-s) as input and experiment with feedforward neural networks, recurrent neural networks and convolutional networks with different structure and parameters for children's age group and gender identification. Most of the similar works done with neural networks have focused on speaker and speech recognition [4], [5], [6].

This kind of automatic identification of speaker age and gender could be used in different child-computer interaction systems, educational applications and to improve child protection systems. For example, it could be used to make applications to adapt content based on the age of the user, restrict children's access to certain webpages and protect children from conversations with adults acting as children over the Internet. In addition, it can be used to improve the performance of other speech related tasks, such as speech and speaker recognition.

To train and test the created neural networks, a speech recordings corpus of native Estonian subjects in the age range from 9 to 19 years is going to be used [7]. As baseline, a method using i-vectors, which represent audio recordings in low-dimensional space, is used. The baseline method determines the age group and gender by calculating the angle between the unit normalized linear discriminant analysis test i-vector and the mean of projected and length normalized i-vectors for each group. This method achieved accuracy rate of 85.8% for identifying the age group of a child on OGI speech corpus [8], which is bigger than the manual determination rate of 67.5% achieved in the same experiment [2].

For validation, the results of this research will be compared with the performance of the baseline method, which has the highest accuracy of the investigated methods for age group identification. In order to compare the achieved accuracy rates with human performance on the same task, an experiment on manual human determination of children's age and gender will be conducted.

1.2 Organization of the Thesis

The following parts of the research are divided into 7 sections. First part of the research gives overview of neural networks and related terminology as well as brings out related work. It is followed by the description of the input data. The conducted experiments are described in chapter 4 and the results are analysed in chapter 5. In the 6-th chapter, the results of the conducted survey are brought out and analysed. Finally, chapter 7 brings out possible usage fields for the built neural networks together with future work that could be done. The thesis is summarized in chapter 8.

2 Background

The following paragraphs give an overview of the techniques and patterns used in this research to build deep neural networks (DNN) and recurrent neural networks (RNN) for speech-based identification of children's gender and age group. In addition, related terminology is explained.

2.1 Gaussian Mixture Model (GMM)

Mixture modelling is a widely used technique for representing a dataset when it can be divided into segments, each following a certain distribution. GMM is a probabilistic model that describes a dataset, where subpopulations follow normal distribution: each mixture component in GMM can be represented through normal distribution and the whole dataset can then be described by a mixture of those independent distributions. Figure 1 shows a Gaussian mixture model (black line), that is created based on a dataset, in which values follow 2 separate normal distributions.

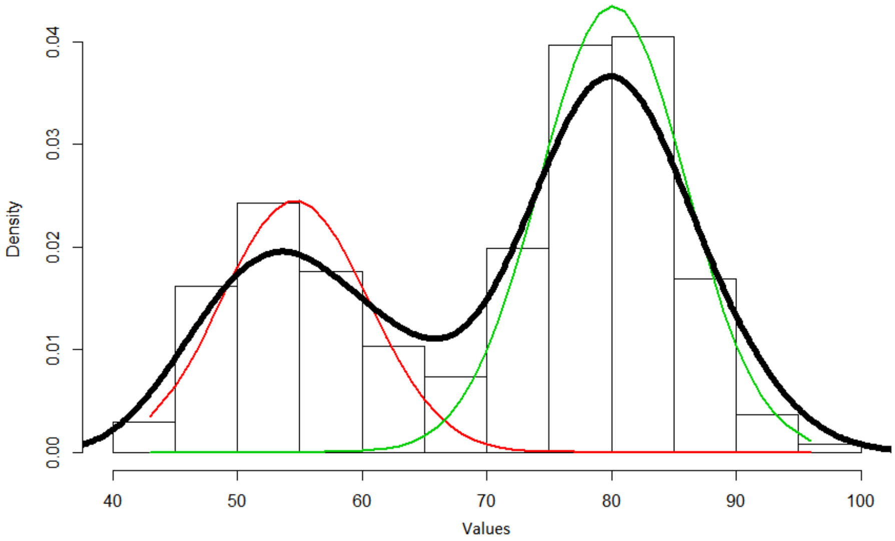


Figure 1 A Gaussian mixture model of two normal distributions.

2.2 Universal Background Model (UBM)

UBM is a model that represents general, person-independent feature characteristics that can be compared against a model of person-specific feature characteristics. For example, in speaker recognition systems, UBM can be used in a likelihood ratio tests to calculate the likelihood ratio statistic. This is the ratio of the probability of an utterance belonging to a specific person to the probability of an utterance not belonging to the person. UBM is used to calculate the probability of the utterance not belonging to the person as it is trained on a large portion of different voice utterances, possibly belonging to different people. The speaker-specific model can be a GMM trained on voice utterances from an enrolled speaker. It means that the ratio can be defined as the probability of the utterance belonging to a specific person to the probability of the given utterance belonging to the UBM [9].

2.3 I-Vectors

I-vector can be defined as a fixed-length representation of a speech utterance in a low-dimensional space that preserves total variability of a signal with speaker-specific information [10], [11], [12]. The i-vector approach is used in state-of-art speaker recognition systems [13].

Given an audio recording of a particular speaker, the locations of speech are first discovered to extract acoustic information from the recording that convey information about the speaker. Typically, this results in 100 feature vectors per second that describe a short-term power spectrum of a sound. In other words, each of those vectors describe a short period of the recording [14].

After that, the sequence of feature vectors is represented by their distribution relative to UBM, which in speech related tasks can be a speaker-independent GMM that describes general speech characteristics. Total variability matrix is then used to turn the parameters of the distribution into a low dimensional (typically 500-dimensional) i-vector. To this end, GMM supervector (μ) is modelled as follows:

$$\mu = \mathrm{m} + \mathrm{Tw} \tag{1}$$

In the formula (1), m represents the mean supervector of the UBM, T is a low-dimensional total variability matrix, which can be estimated through factor analysis, and w is an i-vector having a standardized normal distribution [2], [14].

The i-vectors are used as an input for the feedforward deep neural networks that are going to be built in this research.

2.4 Artificial Neural Networks (ANN)

Artificial neural network, inspired by a human brain, is a set of algorithms that enable computers to learn. It is a tool used for clustering and classification. ANN consists of neurons. Given a set of inputs, a neuron can calculate an output. To this end, neuron needs to have a weight for each of its inputs, which is a real number that expresses the importance of the corresponding input towards the output, and a bias. By using a set of techniques and relevant labelled data, it is possible to solve any mathematical function with artificial neural networks. During the process of building a neural network, weights and biases are first chosen randomly, considering the scale of inputs and outputs. The process of recalculating the weights and biases to solve a given problem is called training the neural network [15].

As a linear combination of multiple linear functions can be replicated through a single linear function, a non-linear activation function is added to each neuron. Otherwise each neural network could be represented through a single layer neural network. Adding the non-linear layer extends significantly the kinds of problems that can be solved with a neural network. Figure 2 shows a neuron that takes two inputs x_1 and x_2, has a set of weights ($w_1 = 2$ and $w_2 = 3$) and a bias ($b = 5$):

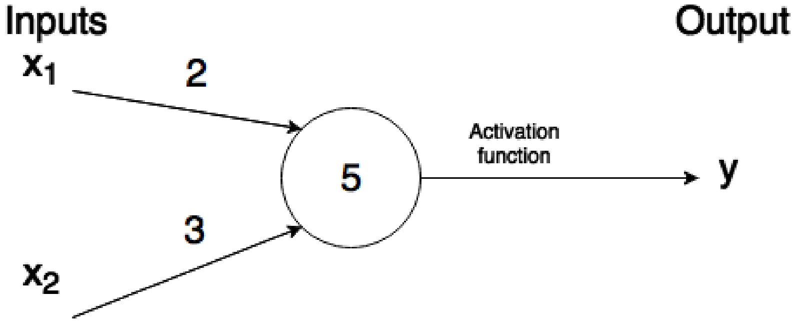


Figure 2 A neuron with two inputs.

Output y of a neuron can be calculated by finding the weighted sum $\sum_j x_j w_j$, adding the bias and finally applying the activation function [15].

An artificial neural network can consist of many neurons combined into hidden and output layers. Layer of a neural network can take as input the output of a previous layer. A layer is called a dense layer if every neuron in the layer is connected to every neuron of the next layer and a sparse layer in case it is not. Figure 3 shows an architecture of an ANN that accepts 2 inputs, has 4 neurons in the hidden dense layer and 1 neuron in its output layer.

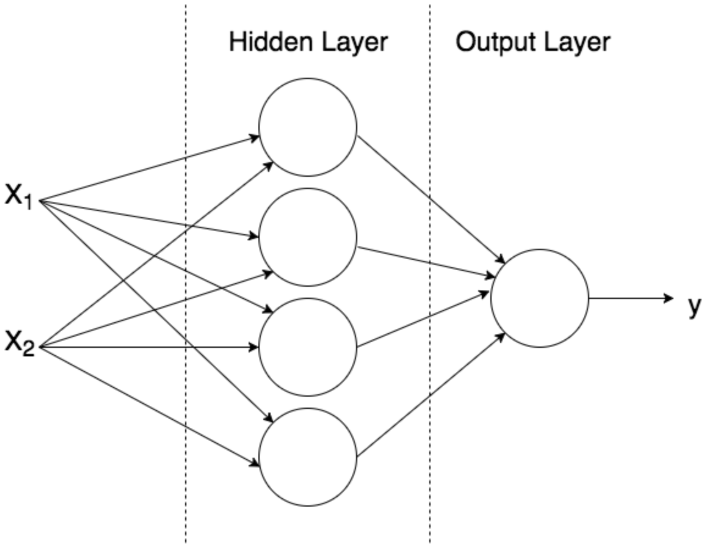


Figure 3 An ANN architecture.

Neural networks with more than 2 hidden layers are called deep neural networks. By combining the features calculated in previous layer, each added layer can find new levels of abstractions and solve more complex functions [16].

2.4.1 Cost Function

Cost function is an indicator that shows how well a neural network is doing. The goal of training the network is to lower the cost function. One of the simplest cost functions is quadratic cost function (2) also known as mean squared error. It measures average of the squares of the errors, between neural network output ($\vec{Y}$) and desired outcome ($\vec{L}$).

$$MSE = \frac{\sum_{i=0}^{n}(\vec{Y}-\vec{L})^2}{n} \tag{2}$$

A model is doing better if the cost function is closer to 0. Choosing the cost function depends on the problem that is being solved. The reason why we cannot use the number of correctly classified examples to evaluate the neural network in classification tasks is that changing weights and biases only a little may not change the number of correct classifications, but it affects the cost function. The number of correctly classified items is an example of discrete data, while the cost function is usually a continuous function [15].

2.4.2 Feedforward and Recurrent Neural Networks

Neural networks can be classified as feedforward and recurrent neural networks. In feedforward networks, the output is calculated based on the current input it is exposed to. It is trained on labelled inputs until it minimizes the cost function and passes information only one way from input towards the output of the neural network. On the other hand, recurrent neural networks are able to consider previously perceived information. They are used, for instance, in speech recognition, because the order of the words in a sentence is important: when recognizing a word, it is important to consider former words in the same sentence. Instead of taking as input only a current example, this kind of neural network can take as input consider with past examples as well. Figure 4 shows a recurrent neural network with two inputs x1 and x2, that is considering the previous output y’ and produces an output y.

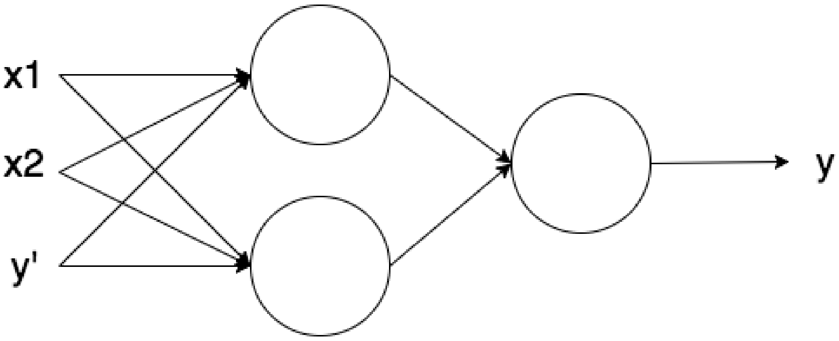
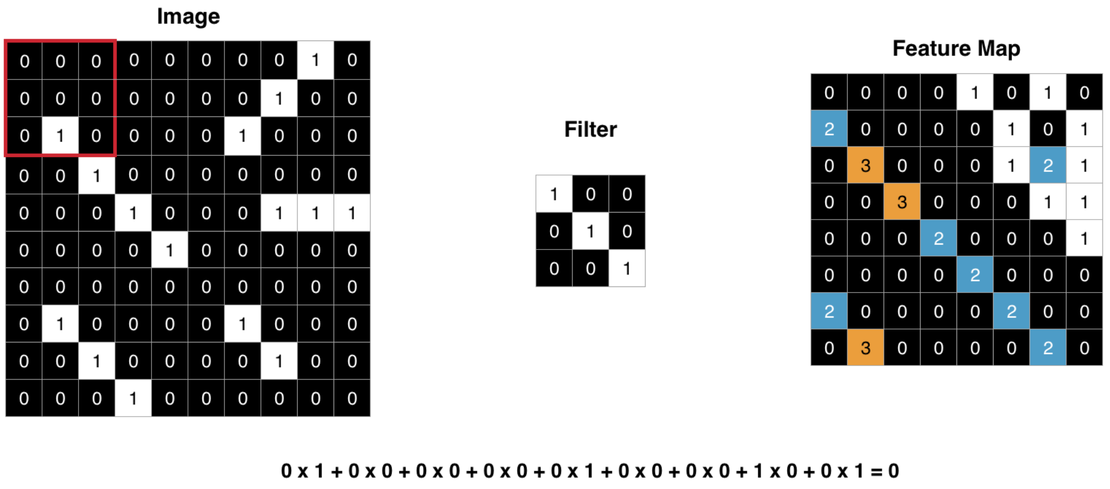


Figure 4 A recurrent neural network

2.4.3 Convolutional Layer

A convolutional layer can be thought of as a filter window that tries to extract useful information based on contiguous features. For images, it could mean that it is a 3x3 filter that is moved over the image pixel grid to find lines from the picture. In case of speech, convolutional filter can capture speech specific information such as the change in the tone of a voice. As convolutional layers make calculation based on multiple inputs instead of considering just one pixel or one array that describes a small specific part of speech, it fastens the overall training process. By adding multiple convolutional layers to the network, more complex structures can be identified. In other words, if first convolutional layer could identify lines and corners on a picture through the analysis of the pixels, the following convolutional layer could recognize squares by combining the outputs of the first layer. Figure 5 shows how filter window is used to calculate feature map based on a picture pixel grid.



Image

0	0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	1	0	0
0	1	0	0	0	0	1	0	0	0
0	0	1	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1	1	1
0	0	0	0	1	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	1	0	0	0
0	0	1	0	0	0	0	1	0	0
0	0	0	1	0	0	0	0	0	0

Filter

1	0	0
0	1	0
0	0	1

Feature Map

0	0	0	0	1	0	1	0
2	0	0	0	0	1	0	1
0	3	0	0	0	1	2	1
0	0	3	0	0	0	1	1
0	0	0	2	0	0	0	1
0	0	0	0	2	0	0	0
2	0	0	0	0	2	0	0
0	3	0	0	0	0	2	0

0 x 1 + 0 x 0 + 0 x 0 + 0 x 0 + 0 x 1 + 0 x 0 + 0 x 0 + 1 x 0 + 0 x 1 = 0

Figure 5 Extracting features of an image with convolutional filter window (from [17]).

2.4.4 Long Short-Term Memory (LSTM)

Long short-term memory recurrent neural network is a kind of neural network that can handle long term dependencies between the current and previous outputs. It was first proposed by Sepp Hochreiter and Jürgen Schmidhuber in 1997 [18].

The idea is about holding relevant past information in the cells of a memory block that is connected to 3 different types of gates: an input gate, an output gate and a forget gate that was introduced in 2000 [19]. Instead of using the whole memory block, LSTM decides based on the current input which part of the block to use. The relevant part of the whole memory block is called working block or hidden state, while the whole block is sometimes referred to as cell state.

To avoid information overload, the forget gate (3) decides before every input, which pieces of LSTM should be kept and which ones discarded. The input gate (4) of LSTM block determines, whether the new input should be stored in a memory cell and the output gate (5) decides, which information should be read from the memory. The gates are vectors that on a given moment t can be calculated with neurons that receive input x, previous output y_{t-1}, have weight W, bias b and a sigmoid activation function σ (6) as follows:

$$f_t = \sigma(W_f \cdot [y_{t-1}, x_t] + b_f) \tag{3}$$

$$i_t = \sigma(W_i \cdot [y_{t-1}, x_t] + b_i) \tag{4}$$

$$o_t = \sigma(W_o \cdot [y_{t-1}, x_t] + b_o) \tag{5}$$

$$\sigma = \frac{1}{1+e^{-y_t}} \tag{6}$$

Figure 6 illustrates the structure of the LSTM layer in recurrent neural networks. The mentioned three gates are shown with vertical connections between the horizontal lines, from which the top one represents cell state.

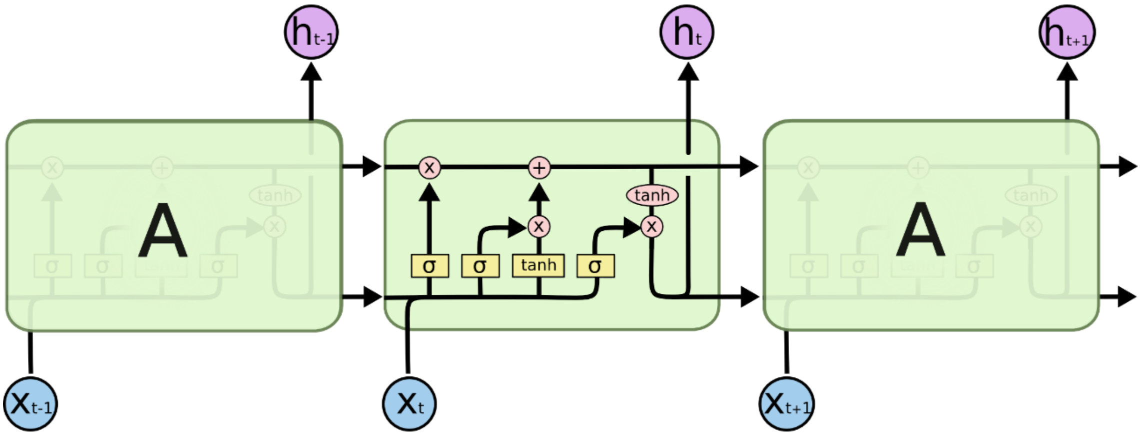


Figure 6 Structure of LSTM layer in recurrent neural networks. (from [20])

2.4.5 Rectified Linear Activation Function

Rectified linear function is an example of non-linear activation function (7).

$$f(x) = \max(0, x) \tag{7}$$

It gives 0 as an output if the input is less or equal to zero. Otherwise it outputs the given input. Blue line on Figure 7 shows the plot of rectified linear function.

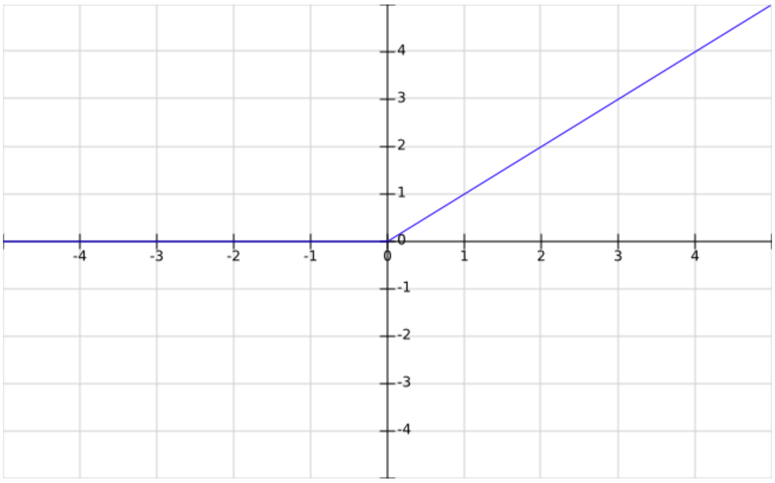


Figure 7 Rectified linear function.

Rectified linear function is computationally less expensive compared to other popular activation functions like sigmoid function (6) or tanh function (8) that use exponential calculations [21].

$$f(x) = \frac{e^x}{1+e^x} \tag{8}$$

Neuron that uses rectified linear function as activation functions is called rectified linear unit (ReLU). As weights and biases of a neural network are first randomly signed, then there is a high probability that many of the neurons with rectified linear activation function will have an output of 0. This creates sparse layers and as a result the training process is quickened. Considering the available computing power in this research, rectified linear function is mainly used as the activation function for the neurons in the built deep neural networks.

2.4.6 Softmax Function

Softmax function (9) is an activation function that is often used in the output layer of a neural network. It gives an output through statistical probabilities.

$$f(X_j) = \frac{e^{X_j}}{\sum_{j=0}^{k} e^{X_j}} \tag{9}$$

In the equation (9) X_j represents the j-th output of a neural network and k is the number of outputs. Softmax function gives the probability distribution of the outcome, with the probabilities ranging from 0 to 1 and adding up to 1. If the neural network is doing a classification task, then softmax function can produce probabilities of the inputs belonging to a certain class. This kind of output is intuitive, because it gives an easily interpretable probability for each possible outcome that shows how sure the neural network is on classifying a given input to a certain class. The softmax function is going to be used as the activation function of the output layer in the deep neural networks built in this research.

2.4.7 Gradient Descent

Gradient descent is a standard algorithm that is used to train neural networks. The goal of training a neural network is to generalize the neural network beyond the training data, by minimizing the cost function. Gradient descent takes advantage of multivariable calculus: the negative of the gradient of a function gives the direction that decreases it most quickly. The negative gradient of a cost function is a vector of partial derivatives of the function that tells in which way and how much each weight and bias of the neural network should be changed in order to make the cost function decrease most rapidly. As the weights and biases are selected randomly in the beginning, then applying gradient repeatedly can move the result of the cost function towards the local minimum. Before finding how much the weight should be changed with gradient descent, a learning rate is determined. Learning rate is a real number that tells the algorithm, how strongly the weight should be changed at once. If the learning rate is very small, then the training takes longer. On the other hand, if the learning rate is too big, the weight can be changed too much and instead of going towards local minimum, the cost might go over it. The change to j-th weight (δw_j) of the neural network can be calculated by multiplying learning rate with the partial derivative of the cost function with respect to the j-th weight (10).

$$\delta w_j = -\eta \frac{\partial C}{\partial w_j} \tag{10}$$

The change to each bias can be similarly calculated by replacing the j-th weight with j-th bias.

To fasten the process of training, instead of considering all the training inputs separately to calculate the gradient, the training data is usually divided into mini-batches, which can give a good approximation of the gradient. Average of those gradients is then taken to find the final gradient, that is used to change weights and biases. The method of using mini-batches for calculating the gradient is called stochastic gradient descent.

2.5 Related Work

The following paragraphs bring out few examples of previous work that have challenged age group or gender identification. Most of the work done with speech, i-vectors or mel-frequency cepstral coefficients are about speech or speaker recognition. There are some examples, where neural networks have been used for age and gender identification based on speech, but only few, where only children have been considered. Most of them are not using RNN-s and only some are using i-vectors as input [22], [23], [24], [25], [26], [27].

2.5.1 Age-Group Identification from Children's Speech

The research conducted in the University of Birmingham in United Kingdom in 2014 used GMM-UBM, GMM-SVM and i-vector systems for children's age-group on the OGI Kids Speech corpus [8], [2]. The GMM-UBM and GMM-SVM systems were used a year before for children's gender identification in another research conducted by the same authors [28].

For age group identification, data was divided into training and test set containing 334 and 766 speakers respectively. Three age groups were created including children between 5 and 16 years of age. The first age group considered children from kindergarten to 3rd grade, the second contained children from 4th to 8th grade and the last one from 8th to 10th grade [2].

Instead of applying neural network to i-vectors, angles between unit normalized linear discriminant analysis test i-vector and the mean of projected and length normalized i-vectors for each class were analysed. Out of the used methods, i-vector system proved to be the most accurate one with an accuracy rate of 85.77% for age-group identification based on band-limited speech to 5.5kHz. The i-vector system was followed by GMM-UBM system with an accuracy rate of 84.07% and GMM-SVM system with an accuracy rate of 79.77%. The results of the research suggest that removing the higher end of the sound spectrum improves the age-group identification accuracy. Analysis of the i-vector system showed that it had difficulties with children who are from 4th and 7th class which are both near the boundaries of the second age group [2]. For gender identification, an age independent GMM-SVM achieved higher overall accuracy of 77.4% compared to the GMM-UBM's accuracy of 67.4%

Both of the researches also conducted an experiment with 20 human listeners on the same dataset. For age group identification, each participant listened to 38 utterances on average. Human listeners achieved accuracy rate of 67.54% over all age groups [2]. For gender identification, each participant listened to 34 utterances on average and the average accuracy was 67%.

The proposed i-vector system is used as baseline method in this research for validating the built deep neural network as it had one of the highest accuracies of identifying age group out of the investigated methods and is similar in the sense that only children are considered. Gender and age group identification accuracies of the proposed i-vector system will be measured on the dataset used in this research.

2.5.2 Language Identification with Recurrent Neural Network

The research done by Google Inc. in New York in co-operation with ATVS-Biometric Recognition Group from Autonomous University of Madrid used long short-term memory recurrent neural network for automatic language identification [29].

The training data used in the research was gathered using an automatic acquisition system from “Voice of America” news. 8 languages were selected for which up to 200 hours of material was available. Average cost was used to evaluate the capabilities of one versus all language detection and equal error rate was used to measure the performance when considering only score for one specific language. The research contained implementing two different baseline systems. The first one was based on i-vectors using linear discriminant analysis followed by cosine distance. It followed the standard procedure described in another research conducted by MIT and School of Computer Science & Advanced Techniques in Paris [11]. Secondly, three deep neural network systems with different number of hidden layers containing units with rectified linear activation functions were built. All of the built neural networks used softmax output layer and cross-entropy cost function. In addition, a LSTM RNN with special units called memory blocks was built [29].

Instead of taking i-vectors as inputs, the built neural networks took frames using perceptual linear prediction (PLP) features from utterances as input. For DNNs, a frame with its 10 neighbour frames from both sides are given as input. In contrast, as

RNN is aware of the context, it does not take extra 20 frames as input. Finally, the language is identified by considering the mean of the results for each frame [29].

The results show that LSTM RNNs outperforms feed forward neural networks on short test utterances [29]. The recurrent neural networks for children’s gender and age identification in this research are inspired by the LSTM RNN built in the described research.

3 Input Data

The following paragraphs describe the data that is used as input for the neural networks built in this research. The data is provided by Tallinn University of Technology and contains speech recordings of native Estonian subjects with related metadata [7].

3.1 Data Collection

The dataset was gathered between January 2012 and December 2014 by Einar Meister and Lya Meister, who both worked at Tallinn University of Technology during that period. The speakers are from capital area and from three other dialectal areas of Estonia. Table 1 shows the number of speakers by the following areas: Harjumaa, North-East Estonia, South Estonia and Saaremaa [7].

Table 1 Number of speakers from different areas

Area	**Saaremaa**	**North-East Estonia**	**Harjumaa**	**South Estonia**
Number of speakers	37	42	189	41

Most of the subjects are from schools that are located in Harjumaa, while there are recordings from around 40 speakers from each of the other areas.

3.2 The Content of the Speech Corpus

The data contains speech from 309 subjects out of which 133 are men and 176 are women. In total the corpus contains 21628 voice files including 70 different recordings from each subject apart from two, with whom 69 recordings were made. The recordings were made on topics that can be divided into 11 logical categories, with the intention of getting linguistically diverse material. For example, the corpus includes speech about numbers, person and institution names, IT terms and description of pictures. To ensure that the speech covers linguistic diversity, several sentences were derived from Estonian Babel Corpus [30], which contains material that includes all Estonian vowels, consonants as well as frequently used diphthongs [7].

3.3 Metadata

Each of the subjects has related metadata that contains information about the speaker and the subject id that can be used to map voice files to the speakers. The metadata contains general information such as the recording date and recording place, as well as the following attributes for every subject: age, gender, school, grade, native language, language at home, place of living, former place of living, spoken foreign languages and indication whether the subject has lived somewhere abroad with the specification of where exactly.

3.4 Statistics

To get overview of how the human voice changes with age, the mean, median, minimum and maximum of the fundamental frequency (F0) were calculated for each of the voice files during the creation of the speech corpus. The fundamental frequency depends on vocal cords length and voice tract length, which tend to be smaller for children and lengthen in the process of becoming an adult [31], [32]. Figure 8 and Figure 9 show how F0 statistics of male and female speakers depend on the subject's age. The F0 statistic of the only 19 years old male speaker in the dataset is shown together with 18 years old speakers.

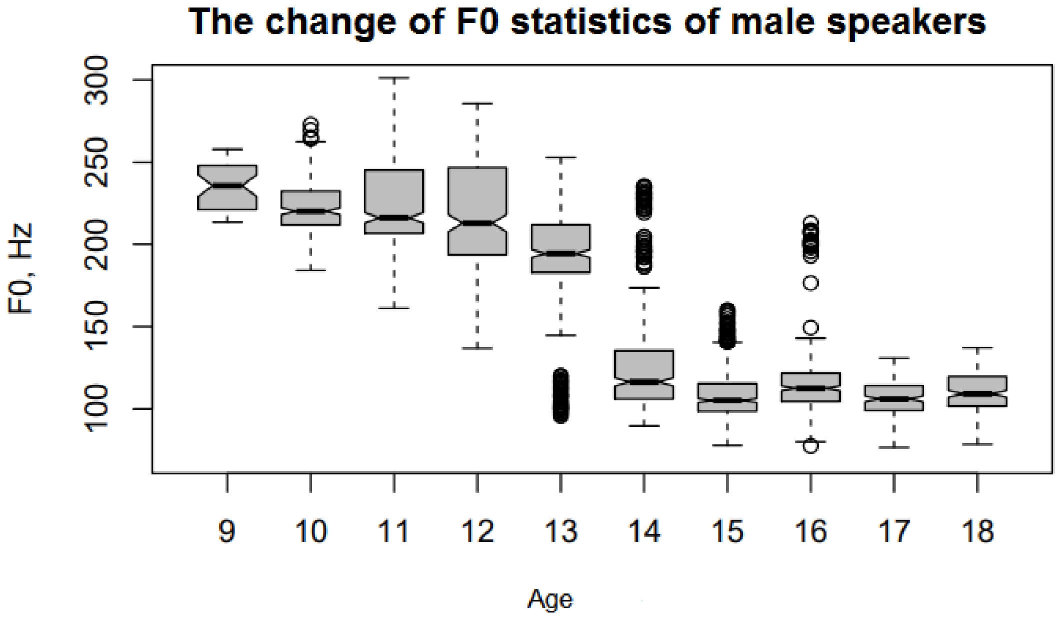


Figure 8 Change of F0 statistics relative to age for male speakers (from [32]).

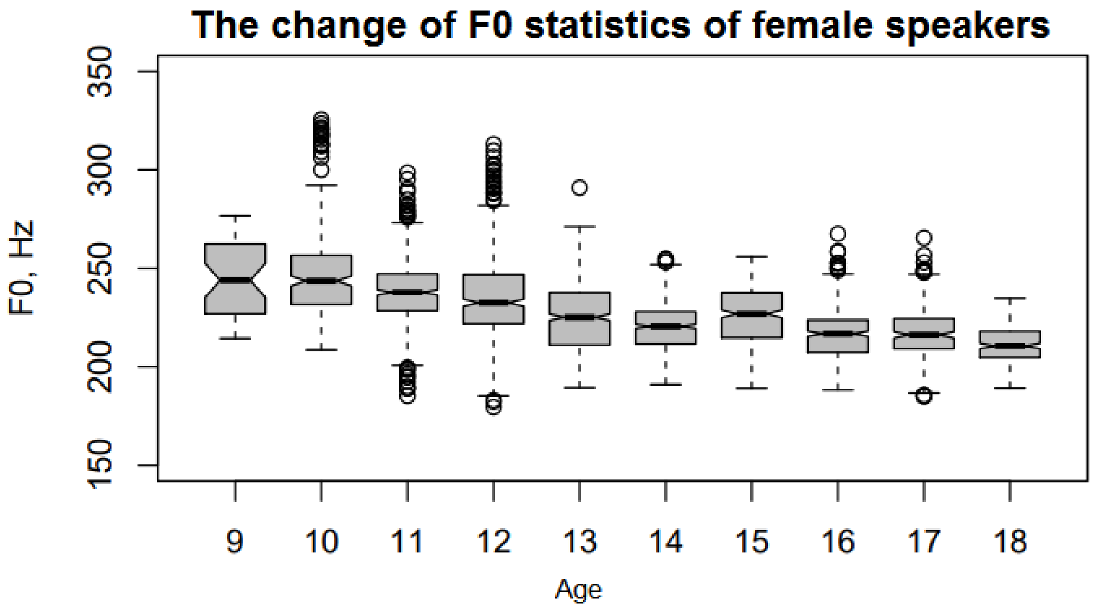


Figure 9 Change of F0 statistics relative to age for female speakers (from [32]).

For women, F0 median lowers almost gradually from 241 Hz to 209 Hz without any remarkable drop-down and is relatively similar for speakers between 14 and 18 years of age. In contrast, for men, F0 median gradually lowers from 236 Hz for 9 years old to 215 Hz for 12 years old speakers. During those years, the F0 median for male and female speakers in the given dataset is relatively similar. There is a little bit bigger drop of 30 Hz in the median of F0 for male speakers between the ages 12 and 13. Between the ages 13 and 14, there is a significant drop-down of 60 Hz in the median of the statistics due to the puberty voice mutation. After that the F0 statistics stabilizes at around 110 Hz [32].

As can be seen from the figures, the number of outliers for female speakers is bigger than for male speakers. Almost all of the outliers of male speakers are aged between 13 – 16 years. This indicates that, although for most of the male speakers the puberty voice mutation happens between ages 13 and 14, there are some males in the given dataset, whose voice deepening happens during other years of age [32]. As the extraction of F0 statistics from speech is not fully automatic, they are not used as input for the neural networks built in this research.

3.5 Data Pre-processing

3.5.1 Age Groups

Based on the F0 median changes for male speakers, the data was divided into three age groups. The first age group considers speakers, who are aged between 9-12. This age group characterizes prepubertal children. The second age group involves children, who are aged between 13-14 and should contain voice that is characteristic to children in puberty. The last age group characterizes the voice of postpubertal children aged between 15-19. One-hot encoding was used to present the age group and gender of each speaker for neural networks that were not built to identify the exact age of the speaker.

3.5.2 Input for Neural Networks

To use neural networks for the speaker's age group and gender identification, numerical representation of each of the voice files were created. To this end, Kaldi [33], an open source speech recognition system was used.

First of all, voice files were signalled into short frames and spectral density was calculated by limiting the upper bandwidth to 3700 Hz for each frame. After that, 23 mel filter banks where applied to the power spectra and the energy was summed in each filter. In the end Discrete Cosine Transform was taken from the logarithms of the filter bank energies, which results in 23 cepstral coefficients. Only coefficients from 2-21 where kept and this resulted in 20 mel-frequency cepstral coefficients that describe short-term power spectrum of a sound. One group of these 20 coefficients is created after each 10 milliseconds of sound and each of them form a vector that characterizes 20 milliseconds of speech. For example, a voice file with length of 2 second is described by a sequence that contains two hundred of the above described 20 dimensional arrays. As the length of the input for the recurrent neural network must be fixed and the available computing power in this research is limited, 80-th percentile of the lengths of the vectors was determined. 80% of the arrays contained less than 1107 of 20 MFCC arrays. All of the longer arrays were cut and 20 dimensional arrays of zeros were added to the end of the shorter arrays. This resulted in a 1107 dimensional array of 20 MFCC-s for each of the voice files.

In addition, i-vectors, that are used as input for deep neural networks were calculated based on the sequences of 20 MFCC-s from previous steps by following the method described in section 2.3 of this research. The main difference between the two different kind of inputs is that while i-vectors are fixed length arrays then the original length of the 20 MFCC depends on the length of the voice file and they preserve the sequentiality of the speech. Both of these inputs are widely used in the field of speech processing [22], [23], [24].

3.5.3 Training, Validation and Test set

Besides dividing data into age groups, it was also randomly divided into training set, validation set and test set, in a way, that training set would approximately contain 80%, while validation and test set would both contain around 10% of the total number of speakers. To avoid data leakage between each of the data groups, speech from a single subject was kept in the same group. Figures 10-12 show how many speakers with specific age and gender are in each of the datasets after the division.

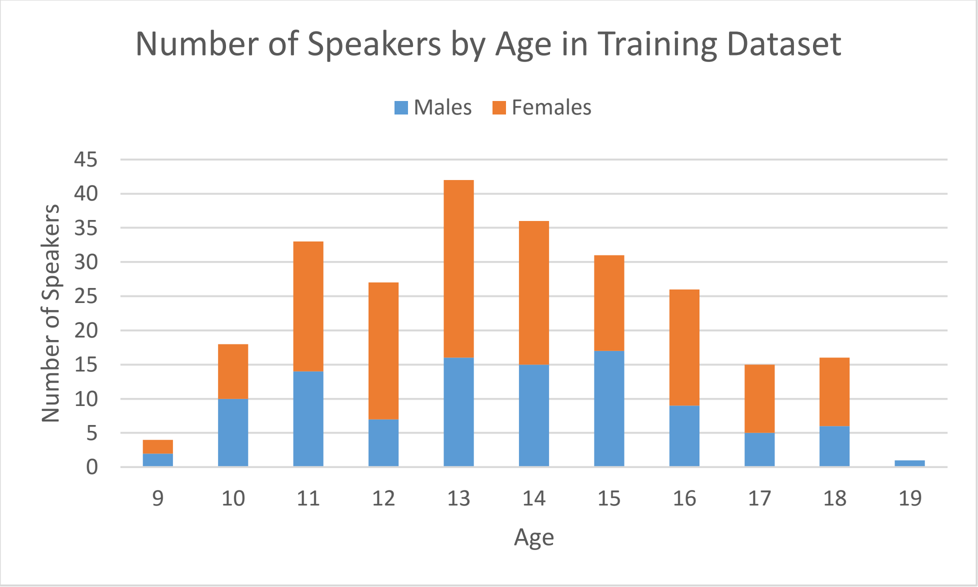


Figure 10 Number of speakers in the training set by age.

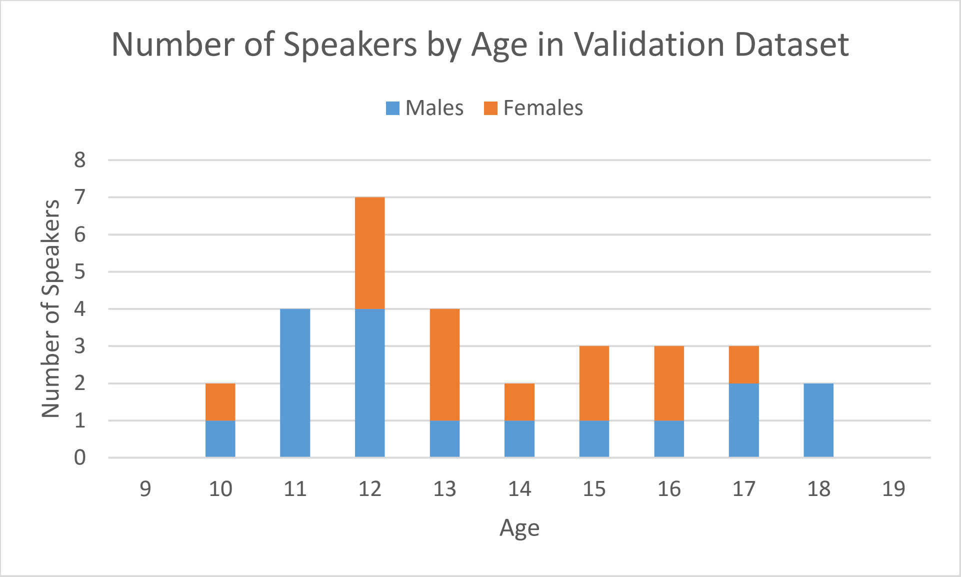


Figure 11 Number of speakers in the validation set by age.

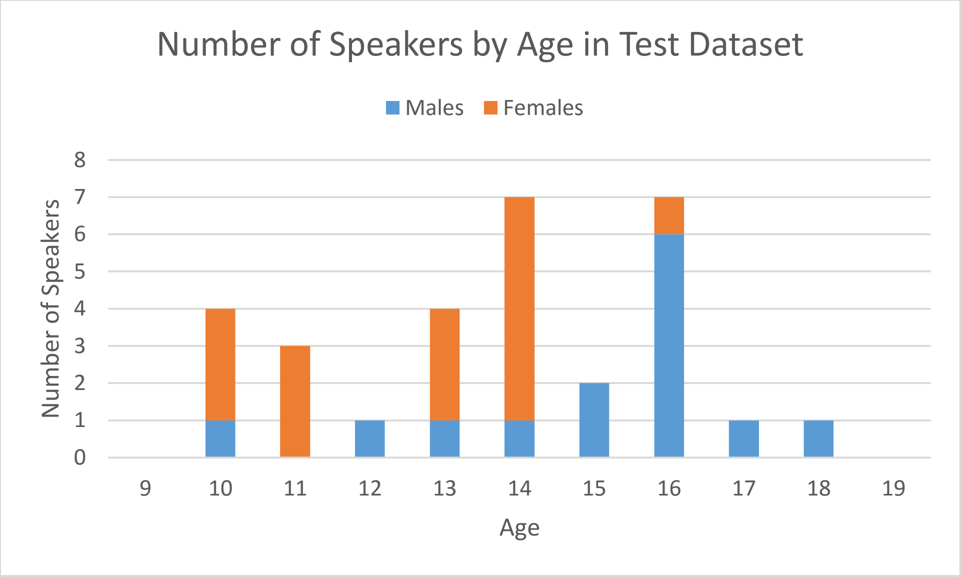


Figure 12 Number of speakers in the test set by age.

4 Experiments

This chapter describes the conducted experiments with neural networks for speech based identification of children's gender and age group. As there are no guarantees that a certain solution is best for the given problem, the process of building neural networks was experimental and many different approaches were tried out. In total, a couple of hundred models with different characteristics were built. The main goal of the built neural networks was to predict either the correct age group out of the three formed sets or the gender of a child speaker based on given input features that represent subject's voice. In addition, models were ensembled to boost the results on test data and the best models were used to identify the age group of boys and girls separately. The research involves experiments with simple feedforward neural networks, multi-output models and recurrent neural networks. Until all the models were built, the accuracies of the models were only measured on training and validation set. The link to the source code of different kinds of feedforward deep neural networks, which achieved highest accuracies, and most of the built recurrent neural networks can be found in Appendix I.

4.1 Technical choices

All the programs were written in Python programming language, due to the ease of use of the language and the connected machine learning ecosystem that includes a lot of libraries for machine learning. For data analysis, Pandas [34], an open source Python library was used. The author of this research was familiar with this library before the research and this helped to save time on data manipulation. In order to train models faster with the power of special GPU, all the calculations were done using EENet computer farm [35]. Tmux [36] was used to keep the server running in the background for models, whose training took long.

4.1.1 Keras

For implementation of neural networks, Keras [37] was used. Keras is a high-level neural networks API, written in Python, that can be used together with Tensorflow and Theano backend. Keras was chosen because it allows easy and fast prototyping and supports feedforward neural networks, convolutional neural networks and recurrent networks, as well as using them together.

4.2 Different Kinds of Built Models

The built models can be divided into different groups based on the structure and purpose. Firstly, two types of models were built that predict only age groups. First of them can be used to predict the exact age of the speaker, which is then manually mapped to the according age group, while others can predict the age group straight away. The advantage of the models that predict age instead of the age group is that they can learn from age data that is ordered and therefore provide more information to the model. To scale the output to the right range, such models used sigmoid activation function in the last hidden layer.

Secondly there are models that predict the gender of the speaker. Those models were in the end used to boost age group prediction accuracy. Figure 13 illustrates possible structure of feedforward neural network for predicting gender with n neurons in the hidden layer. By applying softmax activation function in the output layer of the neural network, probabilistic outputs for the given input being a male or female are received.

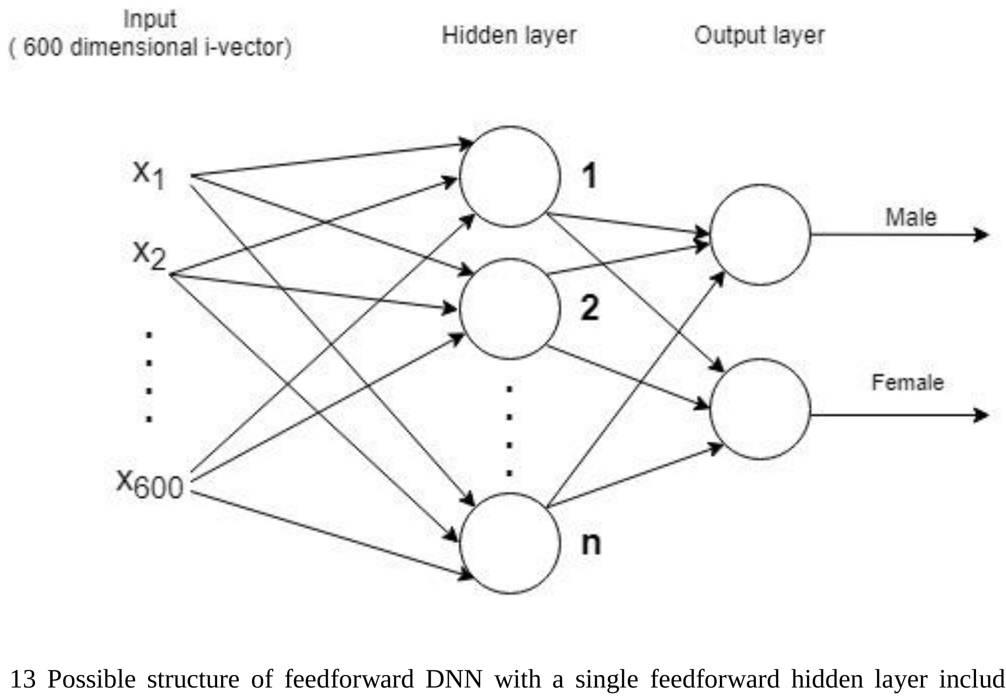


Figure 13 Possible structure of feedforward DNN with a single feedforward hidden layer including n number of neurons.

In addition, some of the built models have two outputs and they can predict either the age and the gender, the age group and the gender or the age and the age group of a given subject. Finally, a few models were built that take as input both, the i-vectors and the sequences of mel-frequency cepstral coefficients. These multi-input and multi-output models could have the advantage over other models by having more information to use in training process and were built using Keras Functional API [38], instead of the Keras

Sequential model API [39] that was used for building other models. Figure 14 illustrates possible structure of a multi-output neural network that predicts the gender and the age of the speaker. In these kinds of networks, each output can have its own loss and activation function.

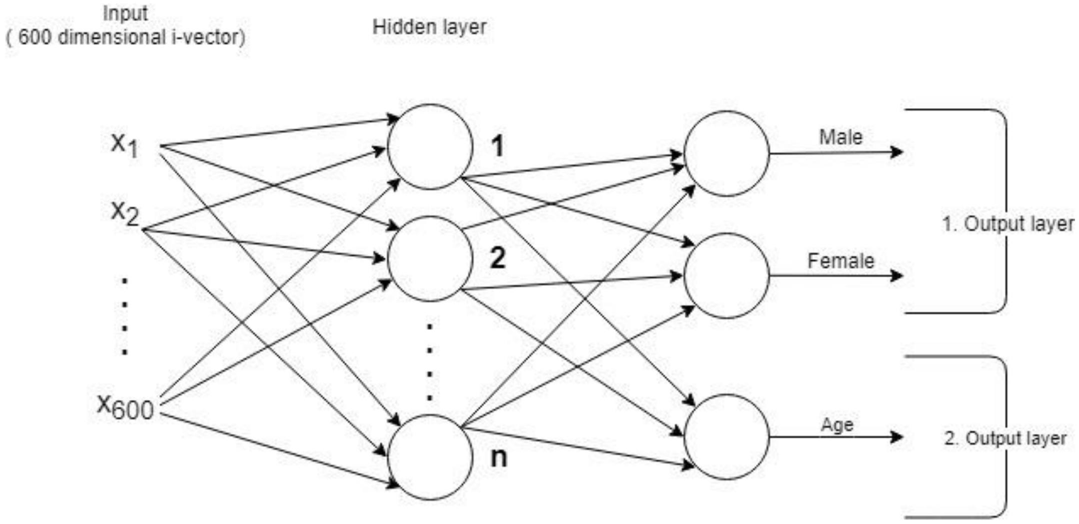


Figure 14 Possible structure of feedforward DNN with two output layers

After building all the models, the accuracies of the ones that had bigger than 70% of accuracy on validation data, were measured on test data. In the end, the models that had top five age group accuracies on test set were determined and their accuracies of predicting males' and females' age groups separately on validation and test data were measured. In addition, these models were separately trained on boys and girls data and the respective accuracies on validation and test data were measured.

4.2.1 Building a Feedforward DNN

First models that were built used only fully connected dense layers. These models took as input the 600 dimensional i-vectors that were calculated for each voice file. As the training of these DNN-s was relatively fast, with one epoch usually taking under 10 seconds, the accuracy of the models on validation data was gradually improved based on quick feedback and only the best models from each kind were saved.

To know if a model is doing well on validation set, Keras offers the possibility of calculating accuracy of the model on a desired set of data after each epoch. The accuracy on validation data was improved by experimenting with changes to parameters and characteristics of the neural networks. Table 2 shows the hyperparameters that were experimented with to build feedforward deep neural networks and majority of the used values for each of them with some of the values taken from Keras documentation [37].

Table 2 Hyperparameters of the built feedforward deep neural networks with used values.

Hyperparameters of the neural networks	**Used values**
Number of hidden dense layers	1, 2, 3, 4, 5, 6
Dimensionality of the output space of each hidden dense layer	Different sizes under 15000
Keras Dense layer kernel_initializers	glorot_uniform, zeros
Keras Dense layer kernel_regularizer	l2 with value of 0.0001
Number of dropout layers	Number of feedforward hidden dense layers in a network or lower
Dropout fraction rate	Tenths from 0.1 to 0.9
Keras activation functions	relu, tahn, sigmoid, LeakyRelu
Keras optimizers	sgd, adam, RMSprop
Learning rate	0.0001, 0.001, 0.05, 0.01, 0.1
Epochs	10 – 50
Batch size	16, 32, 64, 128, 256, 512
Keras loss function	categorical_crossentropy for the models predicting the gender or age group, mse for the models predicting the exact age

,

Keras also offers the possibility to use ready-made callback functions or custom built callback functions that can be run after each epoch. Used callback functions with explanations on why they were used while building deep neural networks are brought out in Table 3.

Table 3 Used callback functions in deep neural networks with explanation

Keras callback function	**Explanation**
EarlyStopping	Was used to stop the training when validation accuracy was not improving for four epochs in a row

Keras callback function	Explanation
ModelCheckpoint	Was used to save the weights and biases of the model in a point where validation accuracy was highest.
ReduceLROnPlateau	Was used to decrease the learning rate when validation accuracy was not improving for two epochs in a row
Custom age_group_accuracy callback	Mapped the age prediction of the model to according age group to get the age group identification accuracy of the models that predicted age instead of age group.

4.2.2 Building Recurrent Neural Networks

Instead of taking fixed length 600 dimensional i-vectors as input, recurrent neural networks have the advantage of learning from sequenced data. Training recurrent neural networks was significantly longer process than training a feedforward deep neural network, because sequences of mel-frequency cepstral coefficients for each voice file are considerably bigger in size than the corresponding i-vectors and the number of computations made in recurrent neural networks is higher. For some of the models without convolutional layers in the beginning, each epoch took multiple hours. As this made getting feedback from the training process longer, the weights of almost all the built recurrent neural networks were saved in the point of biggest validation accuracy from the training process.

Similar callbacks to the ones used in the training process of feedforward neural networks were used in building the recurrent neural networks. The accuracy was measured after each epoch on validation dataset. Table 4 shows different hyperparameters together with used values that were experimented with to build recurrent neural networks. Some of the hyperparameter names and used values are taken from Keras documentation [37].

Table 4 Hyperparameters of the built recurrent neural networks with used values

Hyperparameters of the neural networks	Used values
Keras core layers	Dense, Dropout, Reshape, Masking
Keras convolutional layers	Conv2D
Number of convolutional layers	1, 2, 3, 4, 5, 6

Hyperparameters of the neural networks	**Used values**
Number of filters in convolutional layers	128, 256
Size of the convolutional 2D window	3x20, 5x1, 3x1,
Keras activation function for convolutional layers	relu
Keras recurrent layers	LSTM
Number of recurrent layers	1, 2, 3, 4
Dimensionality of the output space of each recurrent layer	16, 32, 64, 128, 256
Keras layer wrappers	TimeDistributed, Bidriectional
Additional layers	AttentionWithContext [40]
Keras Dense layer kernel_regularizer	l2 with value of 0.0001
Keras optimizers	sgd, adam, RMSprop
Learning rate	0.0001, 0.001, 0.05, 0.01, 0.1
Epochs	10 – 100
Bacth size	16, 32, 64, 128, 256, 512
Keras loss function	categorical_crossentropy for the models predicting the gender or age group, mse for the models predicting the exact age

,

For the parameters that are not mentioned, the default values from Keras framework were used. Along with layers offered by Keras, a special layer called AttentionWithContext [40] was used in some of the models. Instead of considering the whole input equally when making a decision, this let's neural networks to give more importance to certain parts of the input. It allows layers to gain information from different parts of the input and to understand if a part of the input is relevant. In addition, label smoothing [41] was used in some of the models to reduce the risk of overfitting and avoid the model to become too confident in its predictions, which could be helpful for generalization on unknown data.

In the end, some of the recurrent neural networks were trained, by using DNN predictions as labels. This could help to transfer the generalization ability from one model to another from the fact that models' predictions can include probabilities to both, wrong and correct classes. It means that the model could learn that there is a bigger difference between the voice of 18 years old male and 10 years old female than there is between 15 and 13 years old male speakers [42].

4.2.3 Boosting the Accuracy on Test Dataset

After building feedforward deep neural networks and recurrent neural networks, majority voting and weighted averaging was used to boost the accuracies on test data. To this end, the most accurate models for gender and age group identification, as well as for the separate identification of males' age groups and females' age groups were selected. Different combinations of those models were made, by using majority voting and weighted averaging to boost the accuracy for different tasks. Finding the best weights for weighted averaging was done in an experimental way, by trying different combinations of weights. If the majority voting for a given subject resulted in two or more age groups having the same number of votes, the prediction of the model with highest accuracy on the given dataset was used.

In the end, a combination of the built models was used to improve the age group identification accuracy on test dataset. To this end, the models that predicted gender, males' age groups and females' age groups with the highest accuracy on test data were selected. To find the age group of a given example, the gender model was firstly used to predict the gender of the given subject. Based on the prediction, the corresponding model for predicting males' or females' age group was used. In addition, experiments were conducted by not only using the single dedicated model after predicting the gender, but the combination of them. An experiment was done by multiplying the outputs of gender dedicated age group models by the corresponding probability from the gender model and then joining the age group predictions together. The process of joining the separate predictions together, by considering gender predictions as weights, can be seen on Figure 15.

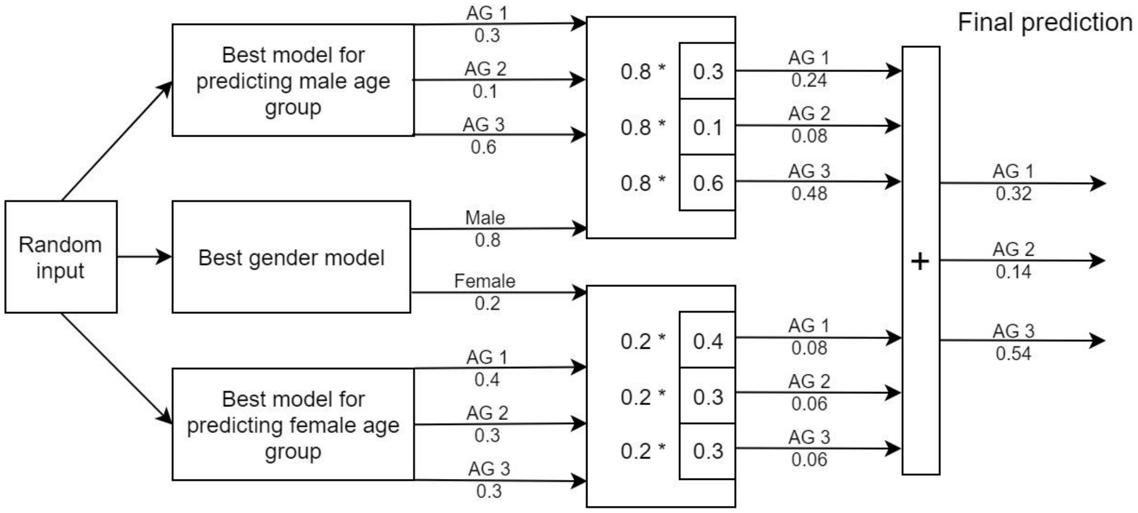


Figure 15 Combining model predictions to boost the age group identification accuracy.

5 Results

This chapter brings out the key results of this research. Different types of models with highest accuracies on test data are brought out and analysed. The weights and biases of the models were mostly saved in the point of highest validation accuracy in training process and the performance on test data was measured, by first loading the saved state of the model. The number of epochs each model was trained for depended on the length of the training process, the amount of overfitting on training data and the stability of the accuracy on validation data after each epoch.

5.1 Identifying Gender of a Speaker

First kind of models were built to identify the gender of a speaker. The following paragraphs bring out different types of built models for gender identification that achieved the highest accuracies on test data.

5.1.1 Feedforward DNN for Gender Identification

The structure of the most accurate feedforward DNN for identifying the gender of a subject on test data, together with used parameters, are brought out in Appendix II. The model was trained for 50 epochs. Figure 16 shows how the accuracy of the model changed on train and validation data after each epoch.

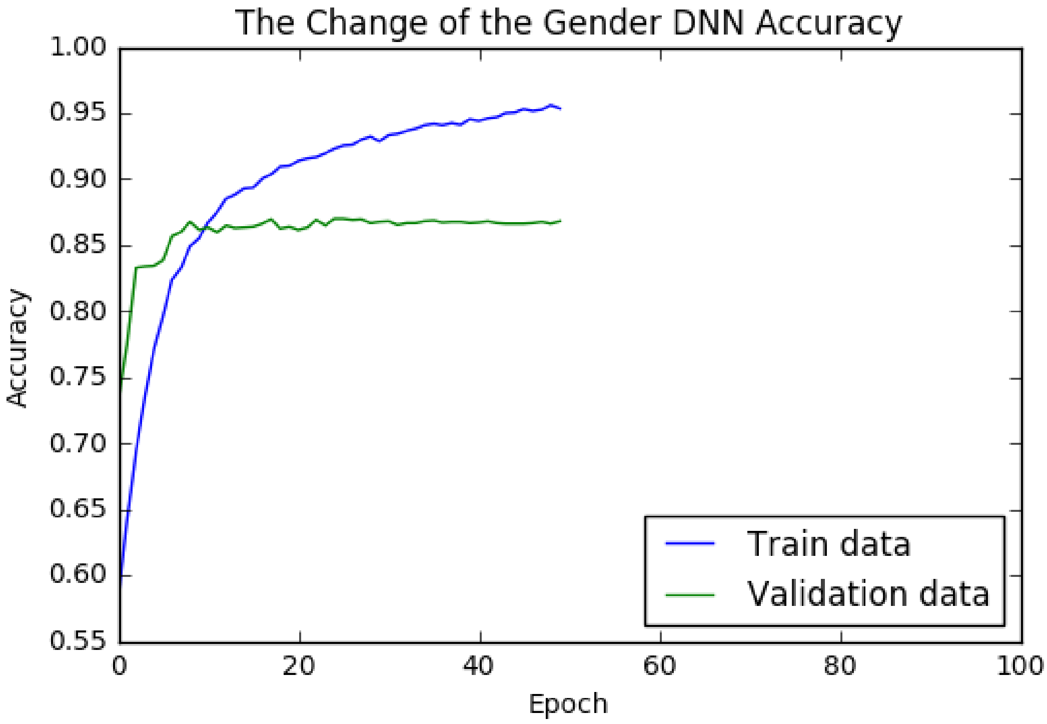


Figure 16 The accuracy of the gender feedforward DNN on train and validation data after each epoch in the training process.

Figure 16 illustrates, how the accuracy change on validation data almost stabilizes completely in the training process. Out of the run 50 epochs, best validation accuracy of 87.0% was achieved after 25 epochs, but close results were visible already after 8 epochs. The accuracy of the model on test dataset with the weights and biases that were saved after 25 epochs is 92.8%, which is the highest accuracy among all the saved models and is also higher than the accuracy of the same model on validation data. The reason behind this could be that there are a lot of male speakers in the third age group of the test data, whose voice has already lowered due to puberty. Figure 17 shows the confusion matrix and the normalized confusion matrix of the predictions of the model on test data.

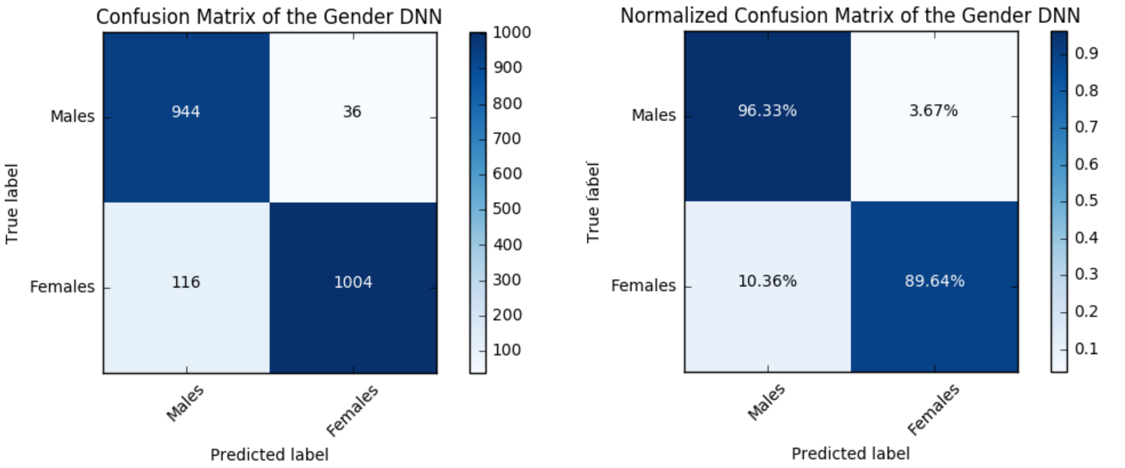


Figure 17 Confusion matrix and normalized confusion matrix of the predictions of the feedforward gender DNN on test data.

The precision of 96.5% for identifying females is higher than the precision of 89.1% for males. In contrast, the recall of 96.3% for males is higher than 89.6% for females. It shows that the model wrongly identifies more female speakers as male speakers than the other way around.

5.1.2 RNN for Gender Identification

The models that achieved the best results among the built recurrent neural networks had convolutional layers in them. The margin between the differences of the accuracies of the best feedforward gender model and the corresponding RNN on test data is a bit less than 8%. The structure of the model that has the highest accuracy on test data out of the built RNN models for identifying gender based on mel-frequency cepstral coefficients is brought out in Appendix III. The accuracies on train and validation data after each epoch in the training process of 50 epochs can be seen on Figure 18.

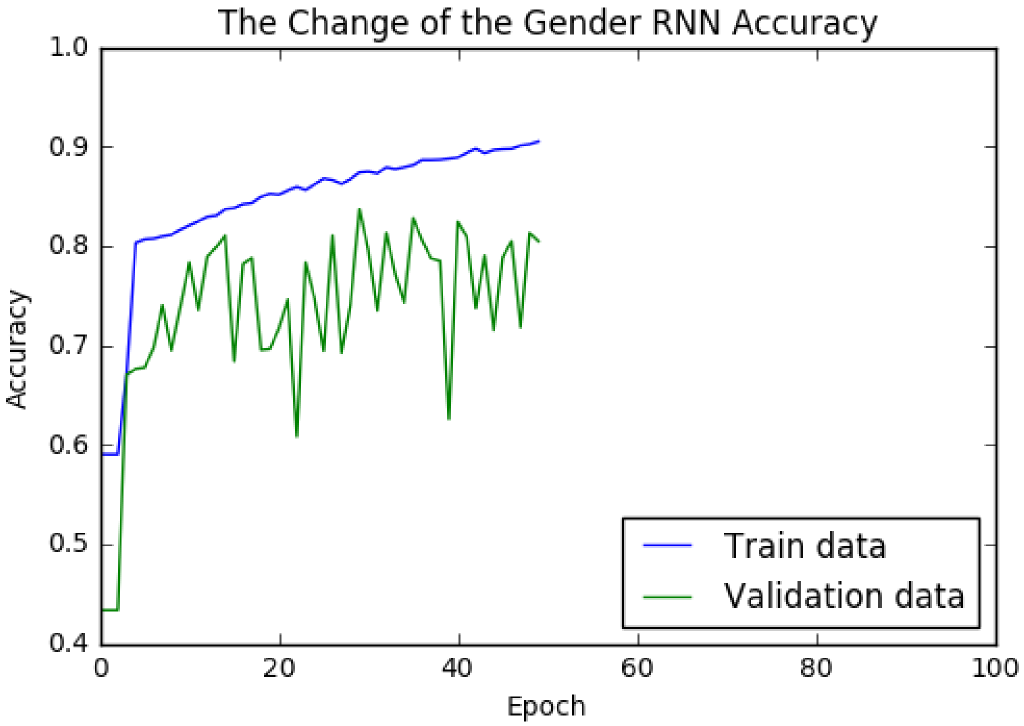


Figure 18 The accuracy of the gender RNN on train and validation data after each epoch in the training process.

Compared to the feedforward DNN brought out in section 6.1.1, the accuracy of the RNN on validation data is unstable and varies a lot after each epoch. Therefore, the line that shows the accuracy change on validation data on Figure 18 is spiky. Overfitting on training data can be detected immediately after first epochs, but it increases slowly compared to the feedforward DNN model. The highest accuracy of 83.7% on validation data was achieved after 30 epochs. The accuracy of the model on test data with the weights saved after 30 epochs is 85.0%. Figure 19 shows the confusion matrix and the normalized confusion matrix of the model's predictions on test data.

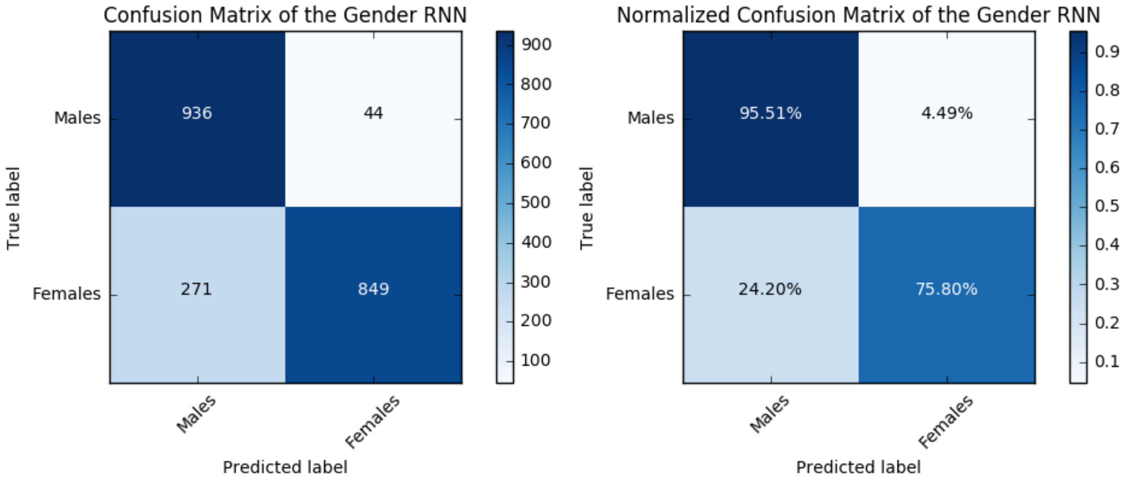


Figure 19 Confusion matrix and normalized confusion matrix of the predictions of the gender RNN on test data.

Figure 19 illustrates that the reason behind the lower accuracy of the RNN model lies behind the low recall of 75.8% for identifying females and low precision of 77.6% for identifying males. Figure 19 shows that compared to the feedforward DNN model, this model wrongly identified over two times more females to be males.

5.1.3 Multi-Input Model for Gender Identification

After using the i-vectors and sequences of MFCC-s separately, some experiments with models that tried to learn from both types of data were conducted. Ordered by accuracy on test data, such model lies between the feedforward DNN and RNN with accuracy of 91.8%. On the other hand, on validation data, this model achieved the highest accuracy of 87.1%, which is a bit bigger compared to the best RNN and feedforward DNN. The structure of the multi-input model, along with the used parameters are brought out in Appendix IV. Figure 20 shows how the accuracy of the multi-input model changed on train and validation data during the training process of 30 epochs.

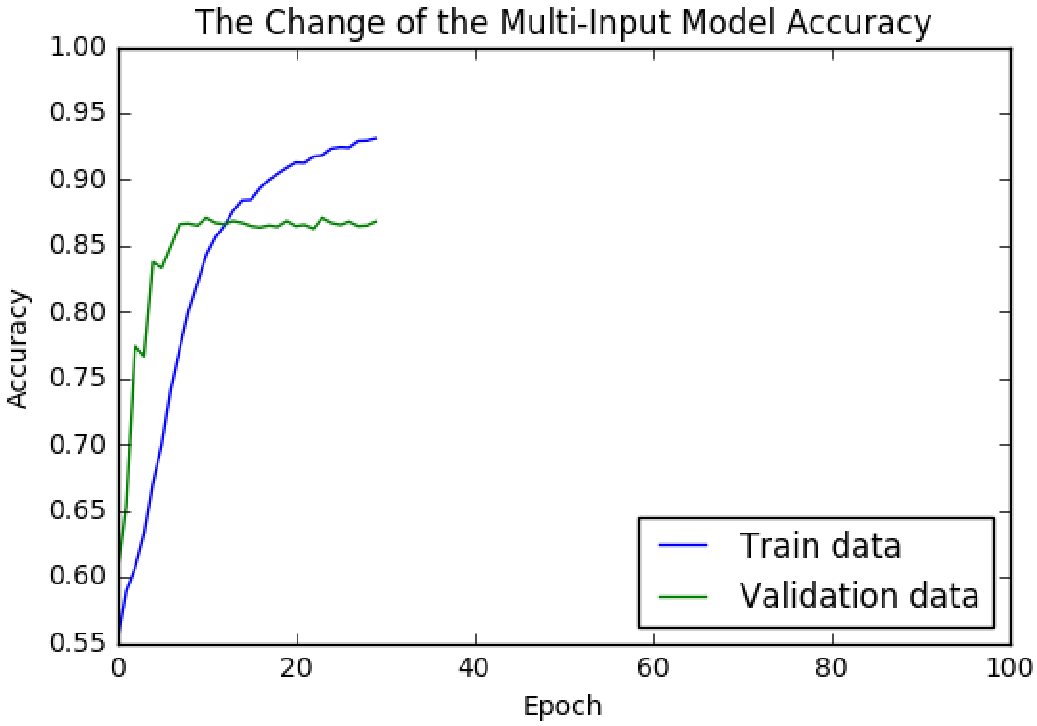


Figure 20 The accuracy of the multi-input gender model on train and validation data after each epoch in the training process.

Figure 20 illustrates, how the accuracy on validation data stabilizes fast after around eight epochs. The changes to the accuracies are very similar to the ones shown on Figure 16. The maximum accuracy of 87.1% on validation data was achieved after 11 epochs and overfitting on training data appears after 14 epochs. Confusion matrix and normalized confusion matrix of predictions of the multi-input model on test data are brought out on Figure 21.

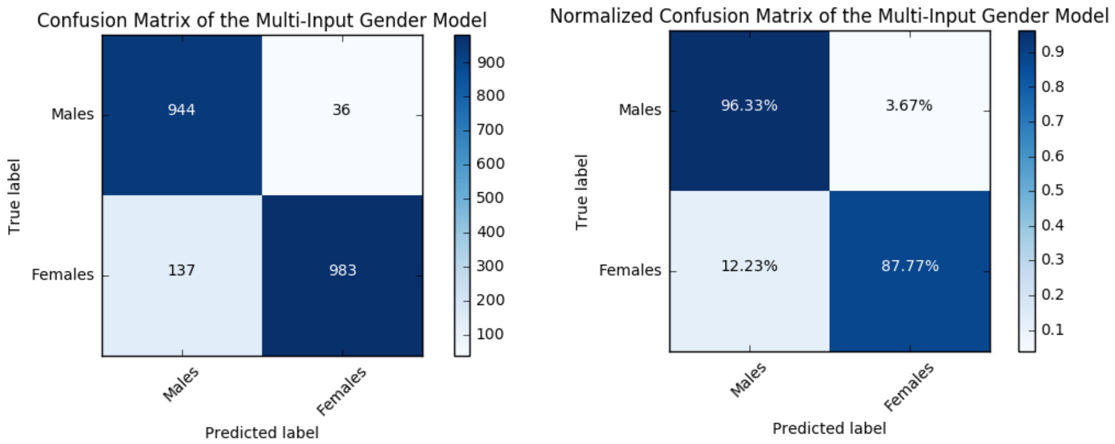


Figure 21 Confusion matrix and normalized confusion matrix of the predictions of the multi-input gender model on test data.

The multi-input model has exactly the same recall of 96.3% for identifying males, but lower precision of 87.3% compared to the feedforward DNN. The recall of 87.8% for identifying females is a little less than 2 percentage points lower. While it is performing a little worse than the feedforward DNN, it outperforms the gender RNN model by accuracy, as well as precision and recall for both genders.

5.2 Identifying the Age Group of a Speaker

As was the case with gender identification, the models built for age group identification can also be divided into three categories. The following paragraphs bring out different types of built models for age group identification, that achieved highest accuracies on test data.

5.2.1 Feedforward DNN for Age Group Identification

Out of the built feedforward neural networks for identifying only the age group based on i-vectors, a model with three fully connected hidden layers achieved the best accuracy on both validation and test set. This model predicts the exact age of the speaker. The age group identification accuracy of the model is measured by mapping the age to the according age group. The structure of the model and parameters that were used to train it are brought out in Appendix V. Figure 22 shows the accuracy of the model on train and validation data after each epoch in the training process of 100 epochs.

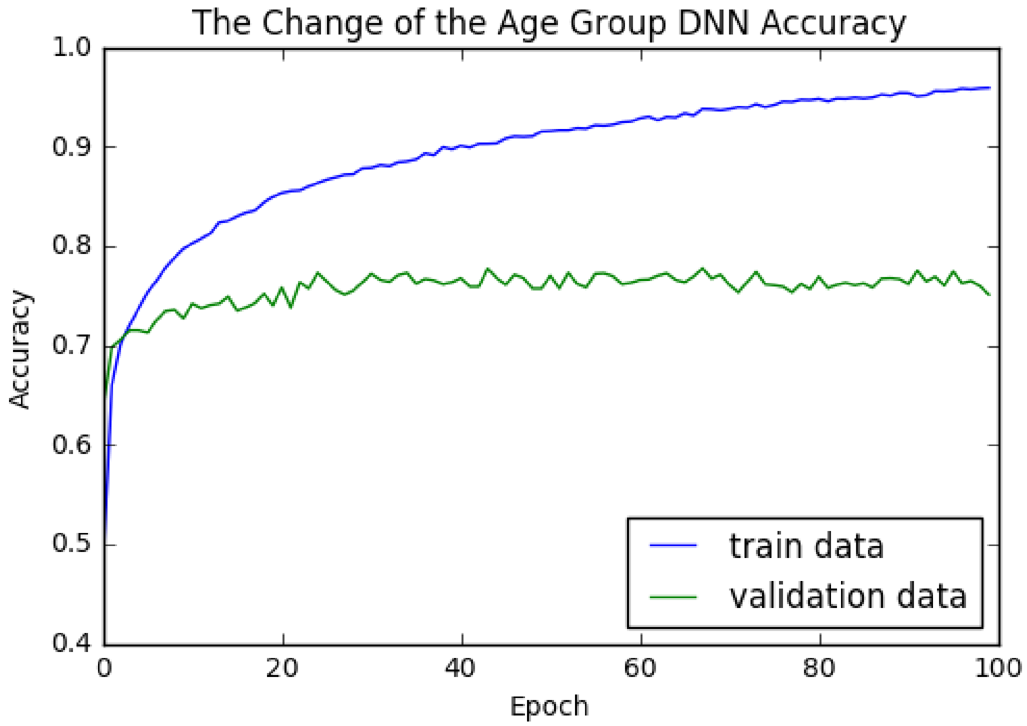


Figure 22 The accuracy of the age group feedforward DNN on train and validation data after each epoch in the training process.

Compared to the feedforward gender model, this model started overfitting on training data earlier at around third epoch. The best accuracy of 77.8% on validation data was achieved after 68 epochs, although similar results were already achieved at around 25 epochs. The model that was saved after 68 epochs achieved accuracy of 72.1% on test data, which means that it is overfitting on validation data. The accuracy measured on test data is similar to the accuracy on validation data from the point of training, where accuracy on train dataset passed the accuracy on validation data. Figure 23 shows the confusion matrix and the normalized confusion matrix of the predictions of the model on test data.

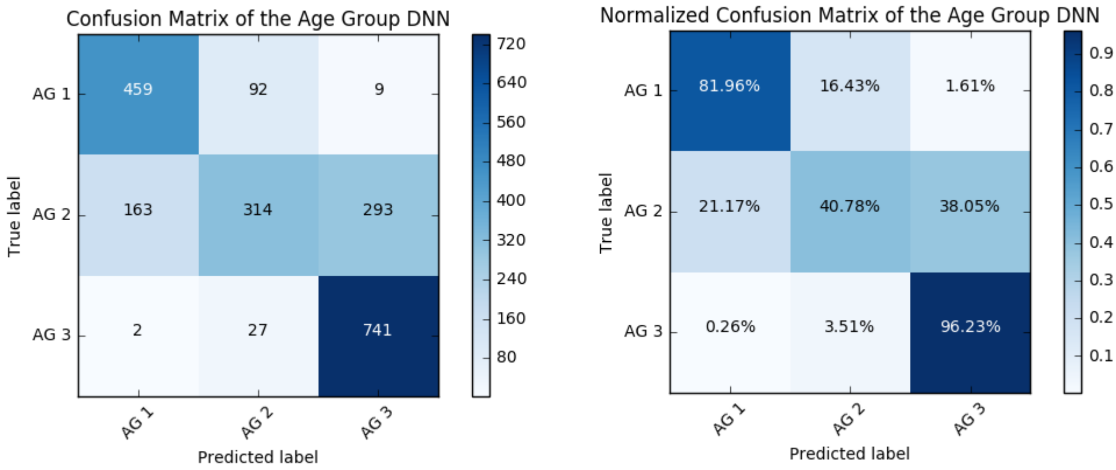


Figure 23 Confusion matrix and normalized confusion matrix of the predictions of the age group DNN on test data.

It can be seen from the confusion matrices, how the model does better on predicting the age group of the youngest and especially oldest speakers. The confusion matrices confirm, what could be seen from the change of F0 statistics of the speakers. F0 statistics of the majority of the 13 years old male speakers was quite close to the younger speakers and most of the 14 years old male speakers had similar F0 statistics to the older speakers. The matrices correlate with the fact that the gap of the F0 statistics for the speakers between 13 years old and younger is bigger than the gap between 14 years old and older speakers. It rather confuses the subjects from second age group with the third age group than with the speakers from the first age group. Still, the model identifies 40.8% of the speakers from the second age group correctly, which is higher than the percentages of wrongly identifying them as speakers from the first or third age group. Although, the recall for identifying speakers from the third age group is highest, the precision for this age group is the lowest. The reason behind this could be that seven out of the total 30 speakers in test dataset are 14 years old, which means they are on the edge of the second and third age group. The fact that out of those seven subjects, six are females also plays an important role, because the difference between the fundamental statistics of 14 years old and 15 years old girls are rather small and it’s hard to distinguish one from another based on voice.

5.2.2 RNN for Age Group Identification

Instead of trying to identify the exact age of the speaker, the RNN model with highest accuracy of identifying age group on test data identifies the age group directly. All of the built RNN models failed to achieve better accuracies on validation data, compared to the corresponding feedforward DNN models. On the other hand, on test data, some of the models performed better. The model with highest accuracy of identifying the age group on test data based on the sequences of MFCC-s is a RNN, that is trained by using feedforward DNN predictions of one of the first built feedforward models as labels. The details of the RNN model are brought out in Appendix VI. Figure 24 shows the accuracy changes of the model on train and validation data during the training process of 25 epochs on feedforward DNN labels.

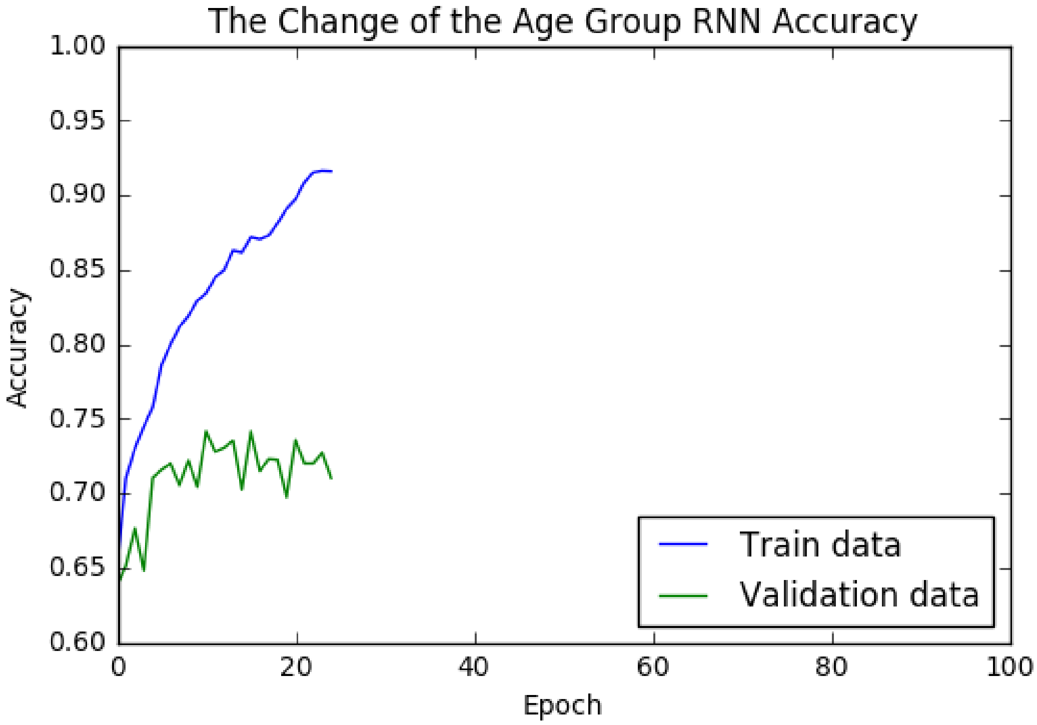


Figure 24 The age group accuracy of the RNN on train and validation data after each epoch in the training process on feedforward DNN labels.

The overfitting of the model on training data increased rapidly during the training process and the accuracy on training data increased faster for this RNN than it did for the feedforward DNN. Accuracy over 90% on training data was achieved in 21 epochs, compared to 41 epochs for the built feedforward DNN. This model achieved the highest accuracy of 74.1% on validation data after 16 epochs. Compared to the feedforward DNN, the accuracy of the RNN on validation data was a bit more unstable and varied more after each epoch. The model saved after 16 epochs has accuracy of 71.2% on test data, which is less than accuracy of 72.1% of the feedforward DNN. Figure 25 shows confusion matrices of the predictions of the model on test data.

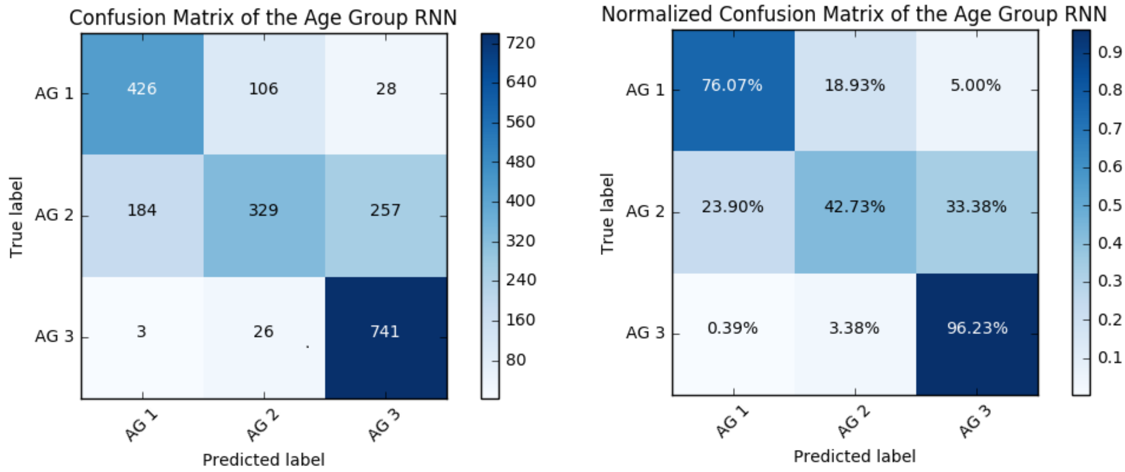


Figure 25 Confusion matrix and normalized confusion matrix of the predictions of the age group RNN on test data.

The figure illustrates, how the RNN also makes most of the mistakes, while trying to identify the age group of the speakers from the second age group. Compared to the feedforward DNN, this model has a bit better recall of 42.7% for the second age group. On the other hand, the recall for the first age group of 76.1% is worse. The precision is worse for both of these age groups. While the recall for the third age group is exactly the same, the precision of 72.2% is better for the RNN model. While the feedforward DNN model made only 11 extreme mistakes by predicting AG 3 for AG 1 or the opposite, the RNN made 31 of such wrong predictions, with the majority of the difference in identifying the speakers from the first age group.

In addition, models were built that used real labels instead of using DNN predictions as labels. The highest accuracy achieved on validation data among those models was 74.9%, but the accuracies on test data were under 70% for all of them. Figure 26 shows the confusion matrix and the normalized confusion matrix of the RNN model predictions, that had highest accuracy on test data among the models that were trained on real labels. The details about the model are brought out in Appendix VII.

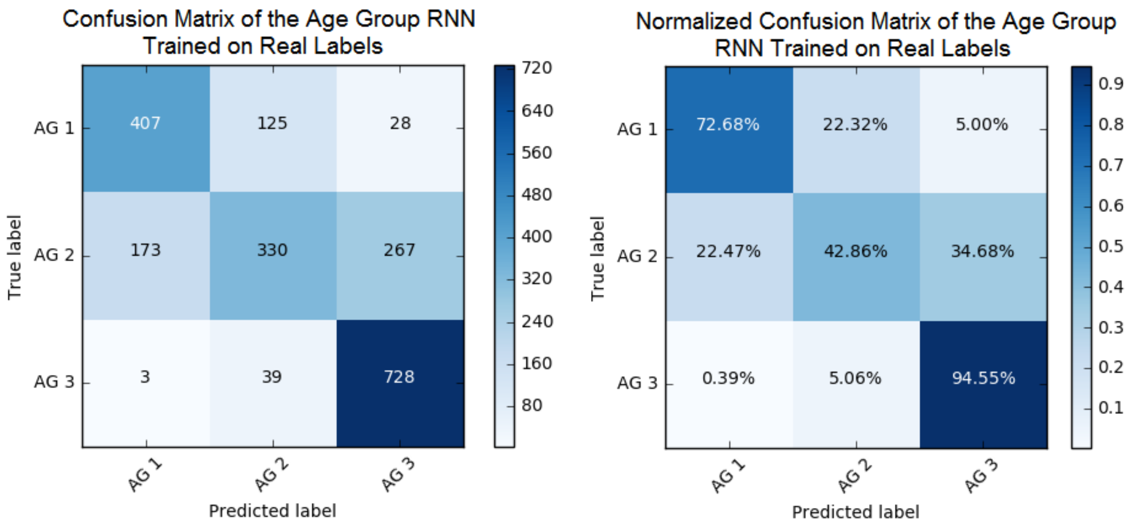


Figure 26 Confusion matrix and normalized confusion matrix of the predictions of the RNN model that was trained on real labels on test data.

Compared to the other age group models, Figure 26 illustrates how the RNN model that was trained on real labels has problems with identifying speakers from the first age group. This RNN has lower recall for the first age group, while other measures are very similar to the model that was trained on DNN predictions.

5.2.3 Multi-Input Model for Age Group Identification

Model that takes as input both the i-vector and sequences of mel-frequency cepstral coefficients achieved the highest accuracy of 73.5% on test data for identifying age group of the speakers. The details about this model are brought out in Appendix VIII. As was the case with the model from section 5.2.2, this model also predicts the age group directly. Figure 27 shows how the accuracy on training and validation data changed after each epoch in the training process.

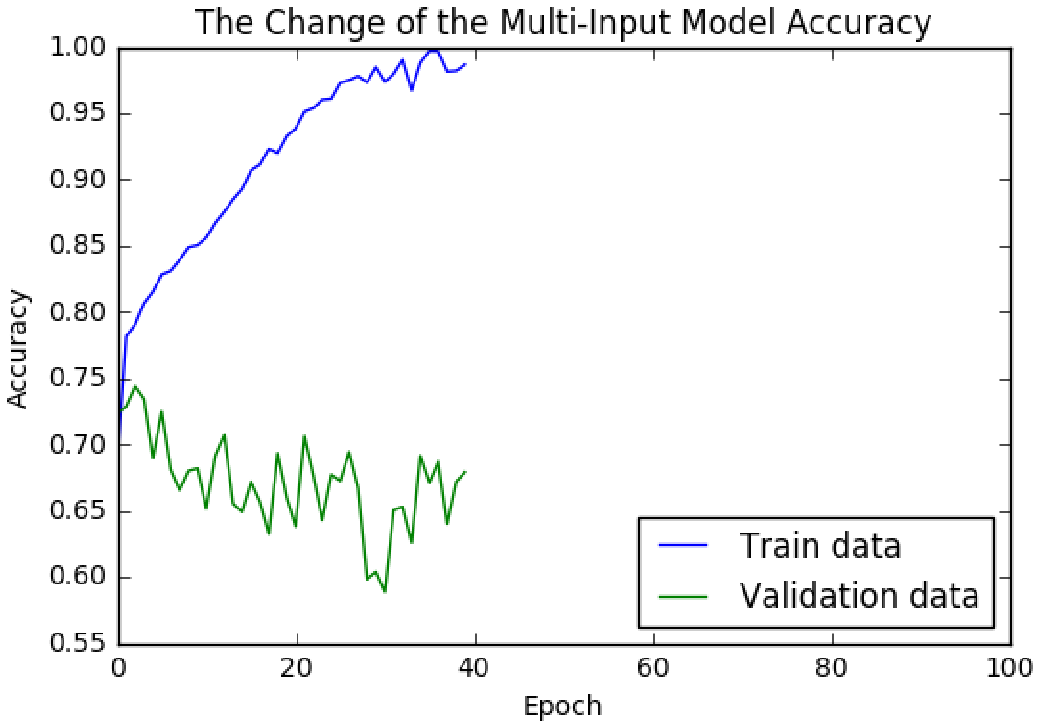


Figure 27 The accuracy of the multi-input age group model on train and validation data after each epoch in the training process.

Compared to the other models for identifying the age group of a speaker, this model achieved its peak in the least amount of epochs. Highest accuracy of 74.4% on validation data was achieved after three epochs and accuracy over 90% on training data was achieved in 15 epochs. It was trained for 40 epochs, during which accuracy of 99.7% was achieved on training data. The reason behind this could be that it had more data to learn from. It is the only model, whose accuracy lowered visibly on validation data after the peak was achieved. The accuracy on validation data is varying highly after each epoch and overfitting on train data increased very quickly. Figure 28 shows confusion matrices of the multi-input age group model predictions on test data.

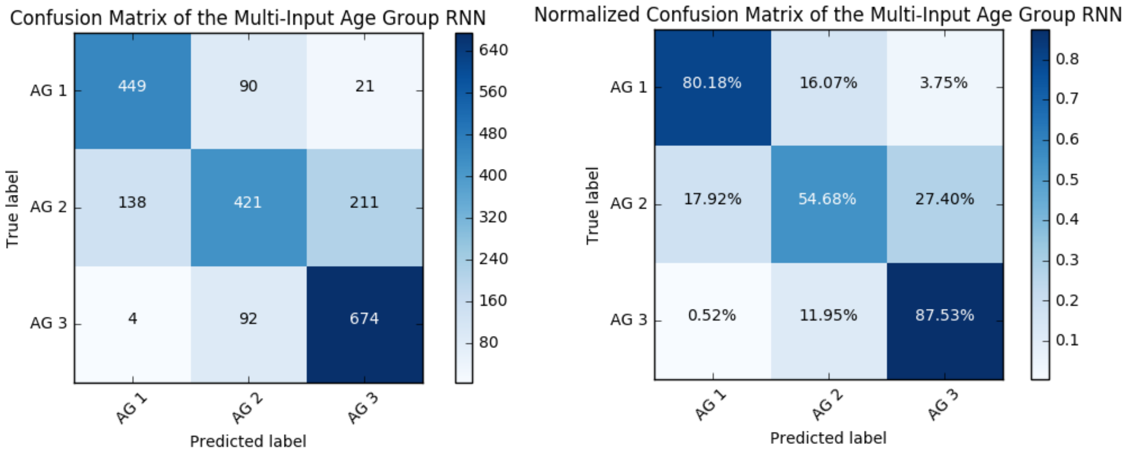


Figure 28 Confusion matrix and normalized confusion matrix of the predictions of the multi-input model on test data.

Compared to the other built age group models, this model has the highest recall of 54.7%, but the worst precision of 69.8% for identifying speakers from the second age group. The precision difference comes from wrongly identifying speakers from the third age group as being from the second age. The recall for the third age group is only 87.5%, which is around 8 percentage points lower than the same measures of the feedforward DNN and RNN. The recall of 80.2% for the first age group, is in between the most accurate feedforward DNN and RNN. In contrast, the precision for the first and third age group are highest for the multi-input model.

5.3 Multi-Output Models for Gender and Age Group Identification

In addition to the models for identifying gender and age group separately, multi-output models were built. Based on one input, such models can make predictions about both the gender and age group of a given subject.

5.3.1 Multi-Output Feedforward DNN for Gender and Age Group Identification

Out of the built feedforward models, a multi-output model achieved the highest accuracy on identifying age group on test data. The structure of this model can be seen in Appendix IX. As was the case with the feedforward DNN from section 5.2.1, this model also predicts the exact age. In addition, this model predicts the gender of the speaker and was also the most accurate model for gender identification on test data out of the built feedforward multi-output models. Figure 29 shows how the age group accuracy and gender accuracy of this model changed during the training process of 50 epochs on train and validation data.

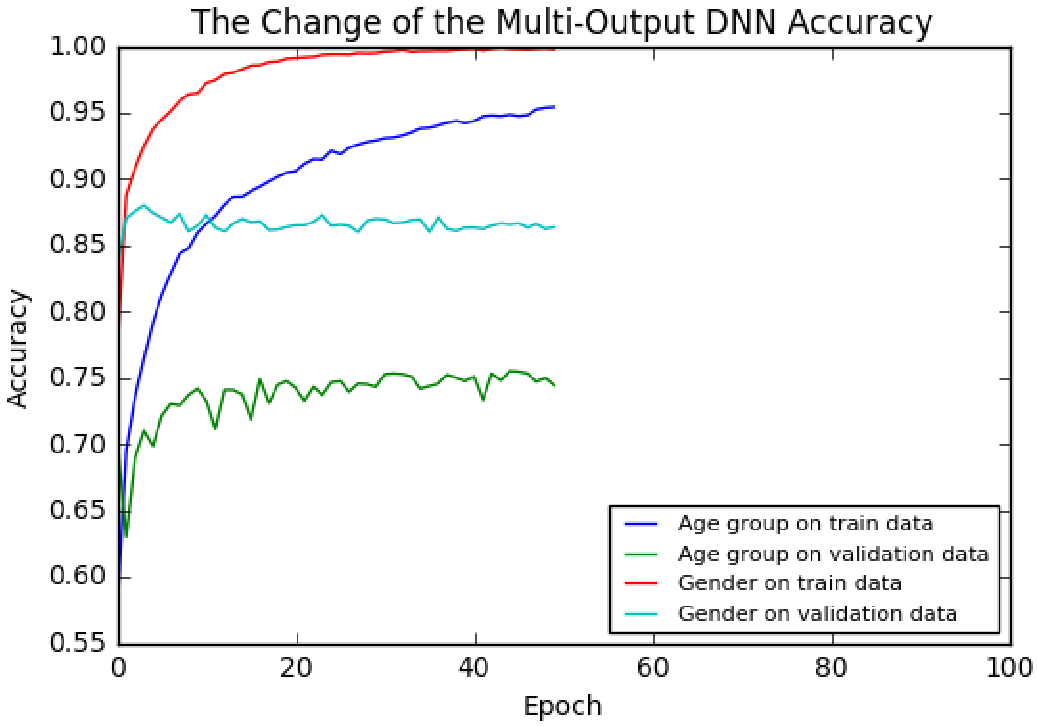


Figure 29 The gender and age group accuracies of the multi-output model on train and validation data after each epoch in the training process.

Figure 29 shows that the best accuracy of 88% for identifying gender on validation data was achieved very quickly, already after four epochs, compared to the highest accuracy of 75.5% for identifying age group, that was achieved after 45 epochs. The peak for gender output was achieved faster than it was achieved during the training process of the single output gender models. For both of the outputs, the model started overfitting on training data very fast during the training process. The model was saved after 45 epochs and predictions were made on test data. It achieved 73.1% of accuracy while predicting the age group and accuracy of 92.3% while predicting the gender on test data. Both of the results are second best on test data in their respective categories out of all the built models. Figure 30 and Figure 31 show the confusion matrices for both of the outputs of the feedforward multi-output model.

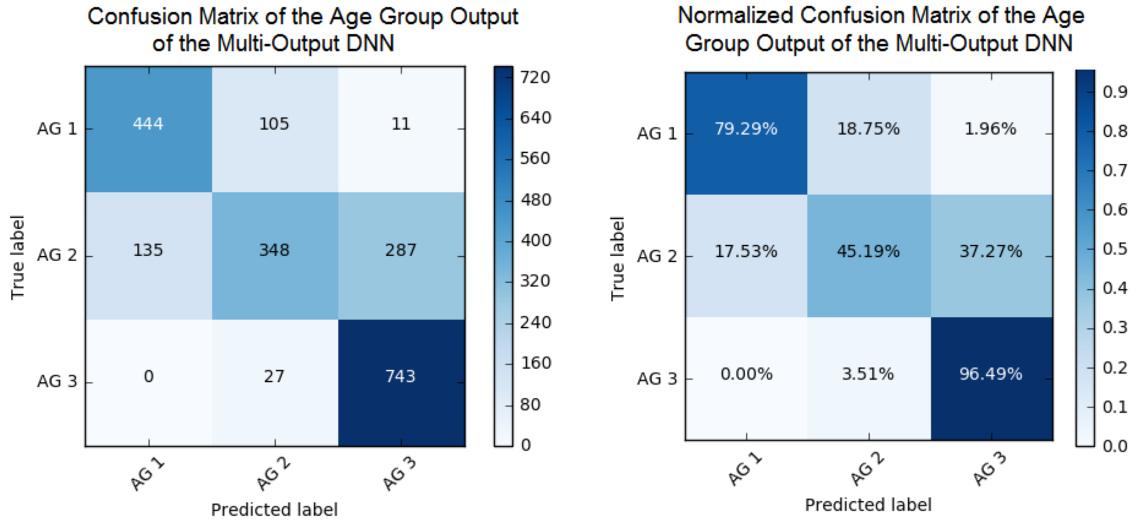


Figure 30 Confusion matrix and normalized confusion matrix of the age group predictions of the multi-output model on test data.

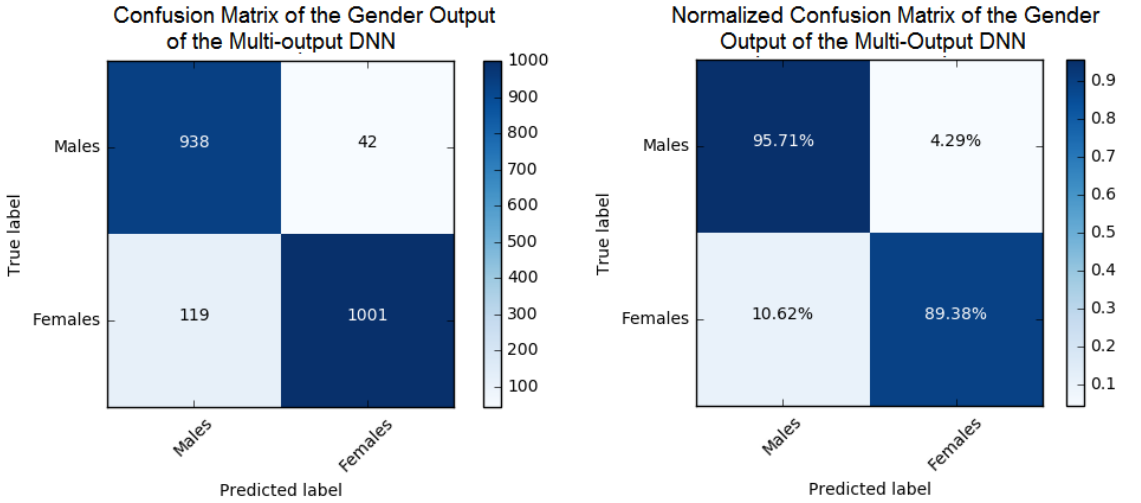


Figure 31 Confusion matrix and normalized confusion matrix of the gender predictions of the multi-output model on test data

Confusion matrices show how the model does better on second age group identification, compared to the feedforward age group model from section 5.2.1. It has a recall of 45.2% for the second age group compared to the 40.8% of the other model, while the precision remained the same. While the recall of this model is 2 percentage points lower for the first age group, the precision for that age group is 3 percentage points higher than the same measures of the model from section 5.2.1. The measures for the third age group are almost the same. The multi-output model did not wrongly identify anyone from the third age group as belonging to the first age group.

The differences between the confusion matrices of the gender output of the multi-output model and the predictions of the feedforward DNN with a single output brought out in section 5.1.1 are minimal. The differences of the recalls and precision for both genders, as well as the overall accuracies of the models were all under 1 percentage point, but the measures were a bit better for the feedforward DNN presented in section 5.1.1.

5.3.2 Multi-Output RNN for Gender and Age Group Identification

Out of all the built models that took only MFCC-s as input, the multi-output model has the highest accuracy for identifying age group of the speaker on test data. Like the models brought out in sections 5.2.1 and 5.3.1, this model also predicts the exact age of the speaker that is mapped to the according age group. The multi-output RNN with the highest accuracy of age group identification on test data, has also the highest accuracy for gender identification on test data out of the built multi-output RNN models. Details of the multi-output RNN are brought out in Appendix X. Figure 32 shows, how the accuracies of the outputs of the RNN changed on train and validation data during the training process of 100 epochs.

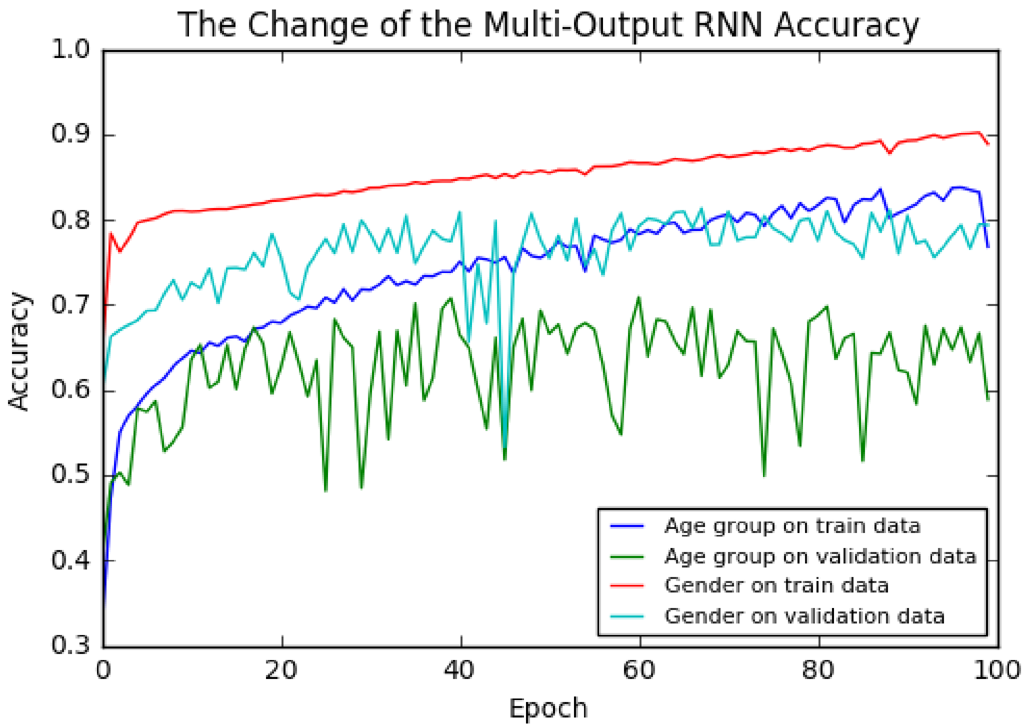


Figure 32 The gender and age group accuracies of the multi-output RNN on train and validation data after each epoch in the training process.

Compared to all the other models, the overfitting on training data during the training process was slowest for this model. The increase of accuracy on training data was slow for both of the outputs. The accuracies on validation data were very unstable compared to the other models that have been brought out in this research. After one epoch in the middle of the training process, the accuracy of the gender output of the model on validation data dropped from 79.9% to 53.2% and the age group accuracy dropped from 66.1% to 51.8%. The highest accuracy of 70.9% for identifying age group on validation data was achieved after 61 epochs and the highest accuracy of 81.33% identifying gender on validation data was achieved after 68 epochs. After 61 epochs the accuracy of the gender output on validation data was 79.2%. The performance of the model on test data were measured with the weights that were saved after 61 epochs. The accuracy of identifying the age group on test data with the mentioned weights was 71.8% and the accuracy of identifying the gender was 82.8%. Compared to the multi-output feedforward DNN, the accuracy of identifying the age group on test data is 1.3 percentage points lower and the accuracy of identifying the gender is 9.5 percentage points lower. Figure 33 and Figure 34 show the confusion matrices of the predictions of both of the outputs of the multi-output RNN.

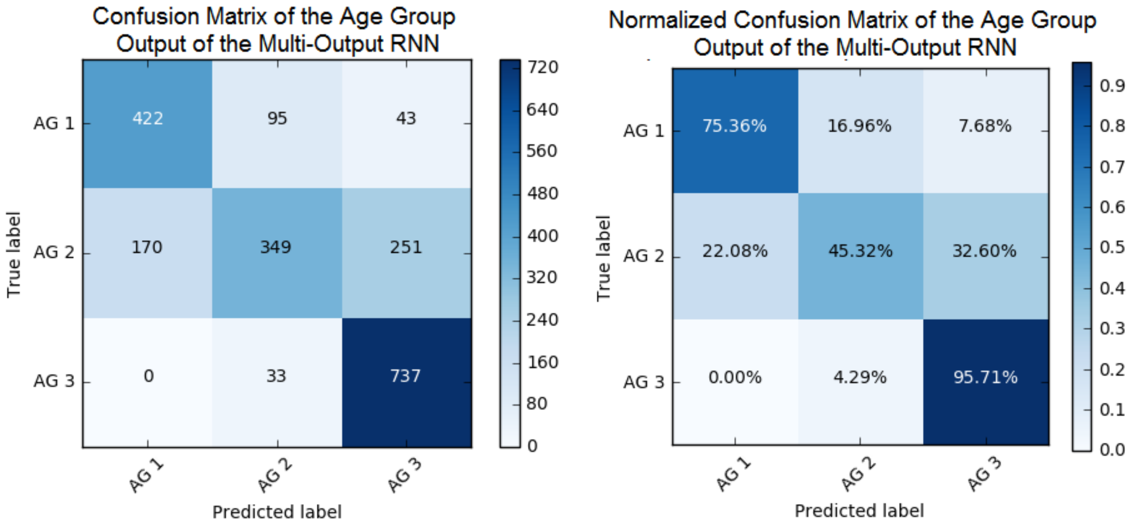


Figure 33 Confusion matrix and normalized confusion matrix of the age group predictions of the multi-output RNN on test data.

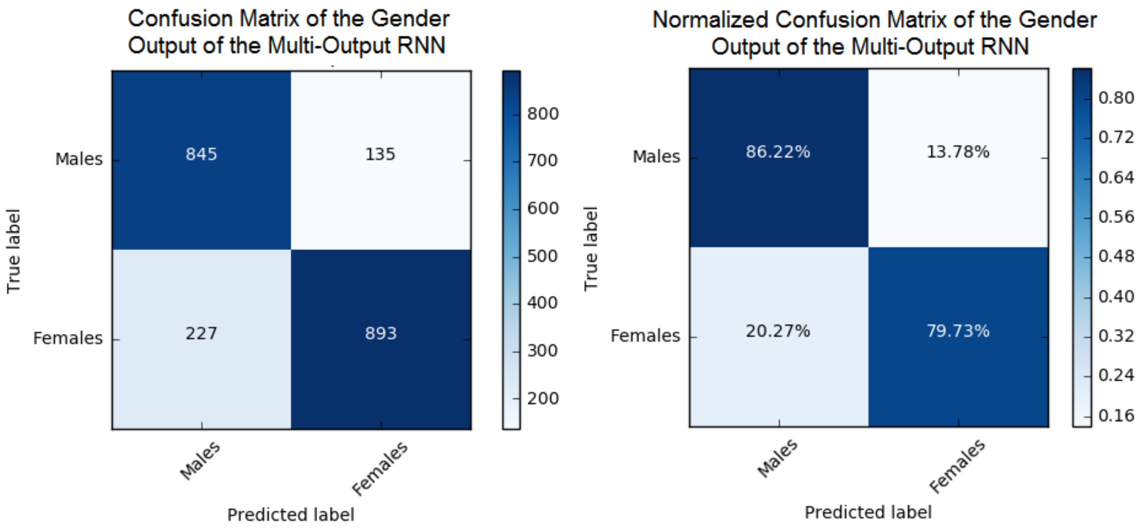


Figure 34 Confusion matrix and normalized confusion matrix of the age group predictions of the multi-output model on test data.

The age group confusion matrix of the multi-output RNN, shown on Figure 33 is very similar to the respective output of the feedforward multi-output DNN that is brought out in section 5.3.1. In addition, the RNN model also does not make any mistakes by identifying the speakers from the third age group as being from the first age group. The key difference of the accuracies of the models comes from the fact that the multi-output RNN has lower recall and precision for the first age group. Compared to the most accurate age group RNN with single output that is brought out in section 5.2.2, the model has a little bit better recall of 45.3% and precision of 73.2% for the second age group, while the recall and precision for other age groups are bit less than 2 percentage points lower.

On the other hand, the confusion matrix of the gender output of the multi-output RNN is quite different from the one of the feedforward multi-output DNN. It has around 9 percentage points worse recall and precision for identifying both of the genders. While the age group output of the model had a little bit better performance than the best RNN with single output for the same task, the gender output has the worst performance on identifying gender out of the models that are brought out in this research.

5.4 Males’ and Females’ Age Group Identification

For all of the saved models that achieved accuracy of 70% or more on age group identification on test data, accuracies of identifying males’ age groups and females’ age groups separately were measured on validation and test data. To this end, female and male samples were separated from both of the datasets and the accuracies of the models were measured separately on the separated datasets. The five most accurate models for identifying males' age groups and females' age groups were selected. The selected models were trained from scratch separately on males' and females' datasets. Multi-output models were not considered while choosing the best models, as there is no advantage in training them on single gender data. The trained models' accuracies of identifying the gender's age group, whose data was used for training, were measured on validation and test data. As the data was separated, the amount of data for validation decreased and the results depend on how the data was randomly divided in the first place. In this section, the most accurate models for separately identifying the age group of males and females are brought out.

5.4.1 Males' Age Group Identification

The most accurate model for males' age group identification on test data was trained on separated males' training data. In essence, it is a RNN and the details about the model can be found in Appendix XI. The model was trained on the average of one hot encoded real labels and corresponding feedforward DNN predictions. The accuracy of the model after each epoch in the training process on separated males' data can be seen on Figure 35.

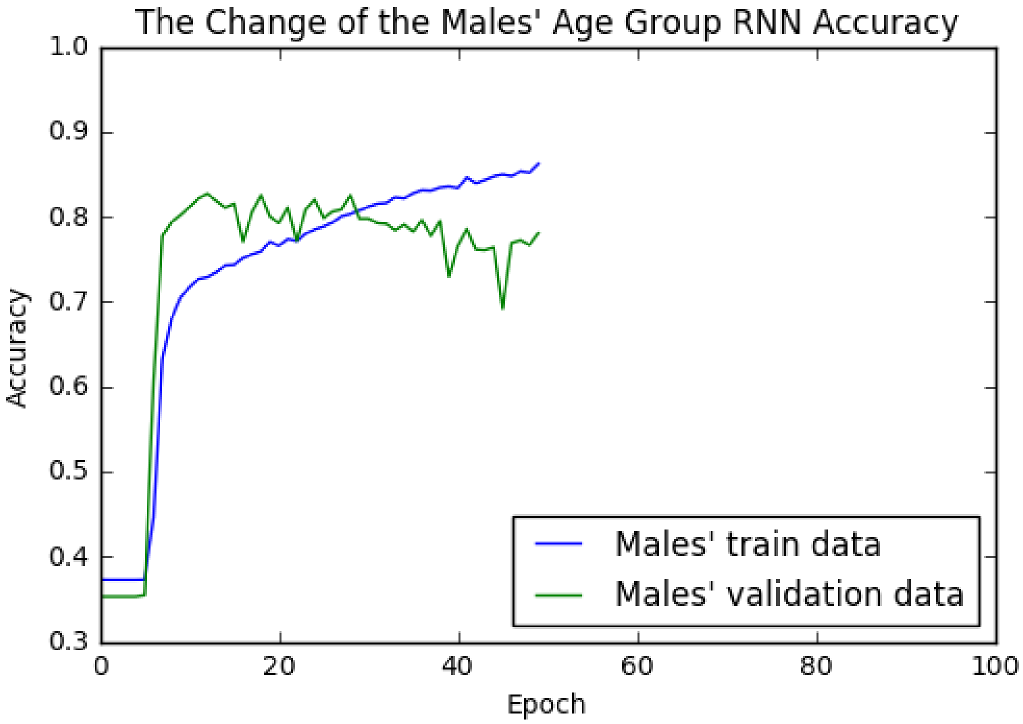


Figure 35 The change of the accuracy of the males' age group RNN on train and validation data after each epoch in the training process.

The figure shows that the change of the accuracies of the males' age group model is quite different from the previously described models. First of all, the accuracies did not change during the first six epochs. After that there was a major increase during two epochs in both of the accuracies. Another difference with the previously described models is that the moment when accuracy on training data passed the accuracy on validation data arrived only after 31 epochs. The highest accuracy of 82.7% on validation data was achieved after 13 epochs, when the accuracy on training data was 72.9%. The model's accuracy on test data after 13 epochs was 85.5%. As the accuracy on training data at the highest point of accuracy on validation data was much lower than the accuracies on validation and test data, the second best model by accuracy on test data was analysed. The second most accurate model for males' age group identification on test data is the same model that is brought out in section 5.2.1, with the difference that it was trained on separated males' dataset. Figure 36 shows the changes to the accuracies of the model after each epoch in the training process on separated males' dataset.

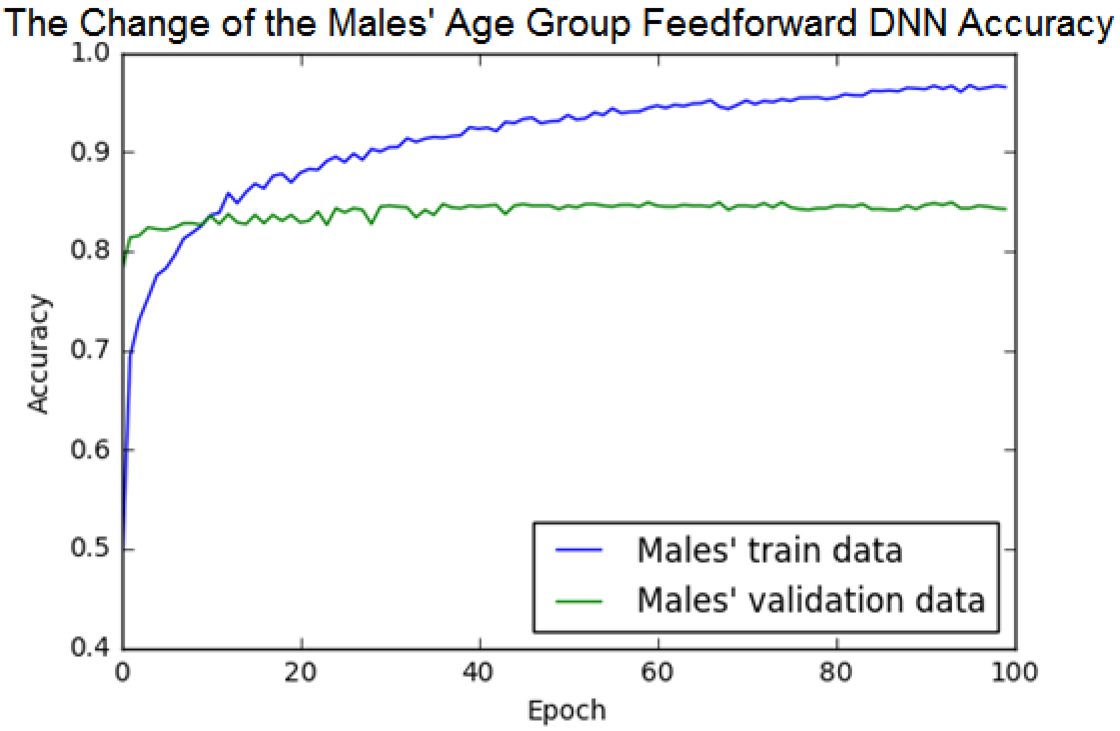


Figure 36 The change of the accuracy of the males' age group feedforward DNN on train and validation data after each epoch in the training process.

Figure 36 shows that the training process of the second most accurate model was smoother and more similar to the other built models. The accuracy on training data passed the accuracy on validation data after 11 epochs and the model is slowly overfitting on training data from that point. The changes to the accuracies after each epoch do not vary a lot and the best accuracy of 84.9% on validation data was achieved after 60 epochs. The accuracy of the model that was saved after 60 epochs on test data was 85.1%, which is very close to the previously described model’s accuracy on test data. As the accuracies on train and validation data at the moment of highest accuracy on validation data are bigger for this model, it is considered to be the best model out of the built models for males’ age group identification in this research. The higher accuracies on training and validation data show that the model is working better on more amount of different data. The confusion matrices and the normalized confusion matrices of the predictions of this model on validation and test data are brought out on Figure 37 and Figure 38.

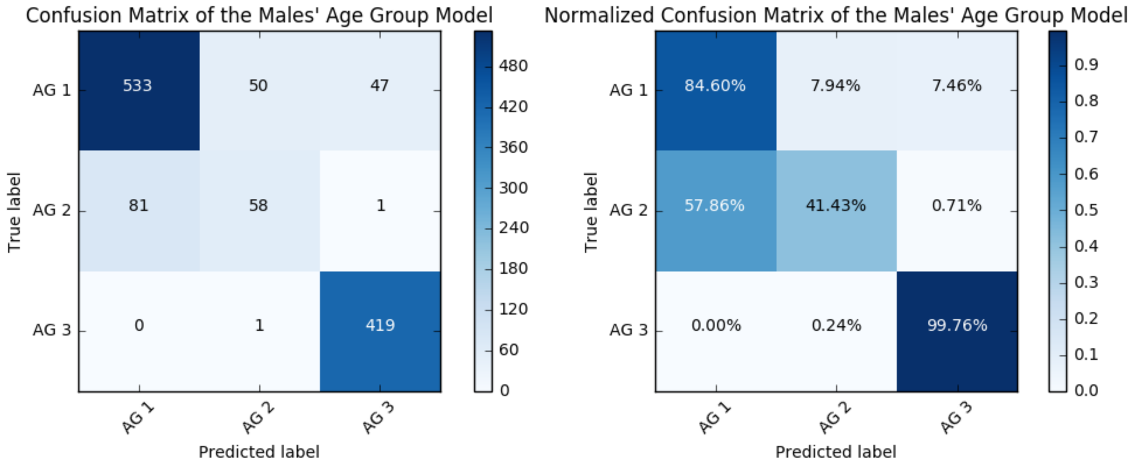


Figure 37 Confusion matrix and normalized confusion matrix of the males’ age group model on validation data.

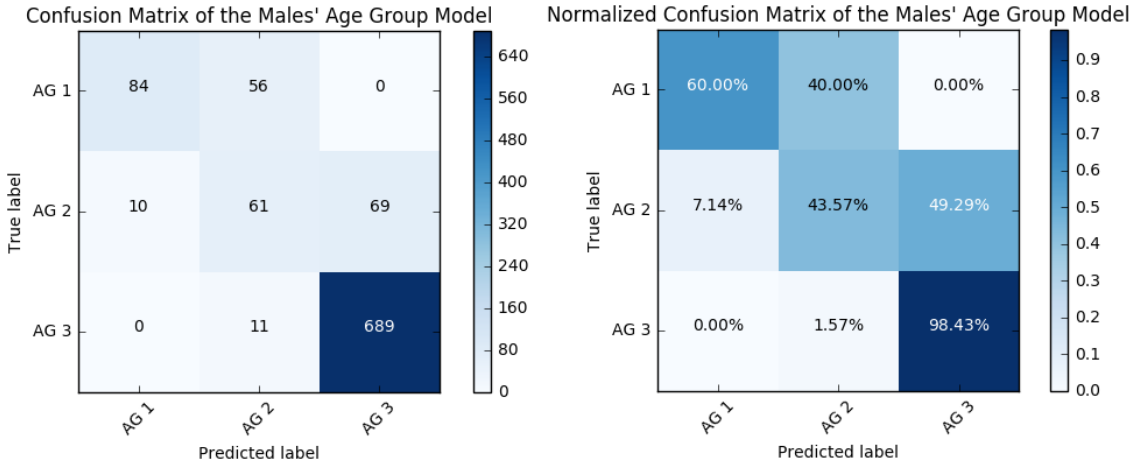


Figure 38 Confusion matrix and normalized confusion matrix of the males’ age group model on test data.

Figure 38 shows that the model is doing well on males’ test data, because the majority of the male speakers in test dataset are from the third age group. For this purpose, the confusion matrices of the predictions for voice files from males’ validation data are brought out. Figure 37 shows that the model is achieving close results with a little bit more equally distributed validation data. The model does well by not identifying anyone from the third age group as being from the first age group on both of the datasets. On test data, the model does not do very well on predicting the first age group, as the recall for this group is only 60.0%. The reason could be that in the first age group in test data, there is one 10 years old speaker and one 12 years old speaker and the second one is often wrongly predicted to be from the second age group. In addition, the confusion matrices show, that the model is having problems with identifying speakers from the second age group. On validation data, the model wrongly identifies a lot of speakers from the second age group to represent the first age group and on test data, it often identifies speakers from the second age group to represent the third age group. The reason behind this could be that there are voice files from one 13 years old male speaker and one 14 years old male speaker in both validation and test datasets. For validation data, one of the speakers could be often wrongly predicted to be from the first age group, while for the test data, the model could identify the voice files from one of the speakers to be from the third age group.

5.4.2 Females' Age Group Identification

The most accurate model for females' age group identification on test data is the multi-input model from section 5.2.3. None of the models that were trained on separated females data outperformed this model, that was trained on the full training dataset. The training process of the model can be seen on Figure 27. The model that was saved at the highest point of accuracy on validation data achieved 67.1% of accuracy on validation data and 69.5% of accuracy on test data on females' age group identification. Confusion matrix and the normalized confusion matrix of the model's prediction for separated female's validation and test data can be seen on Figure 39 and Figure 40.

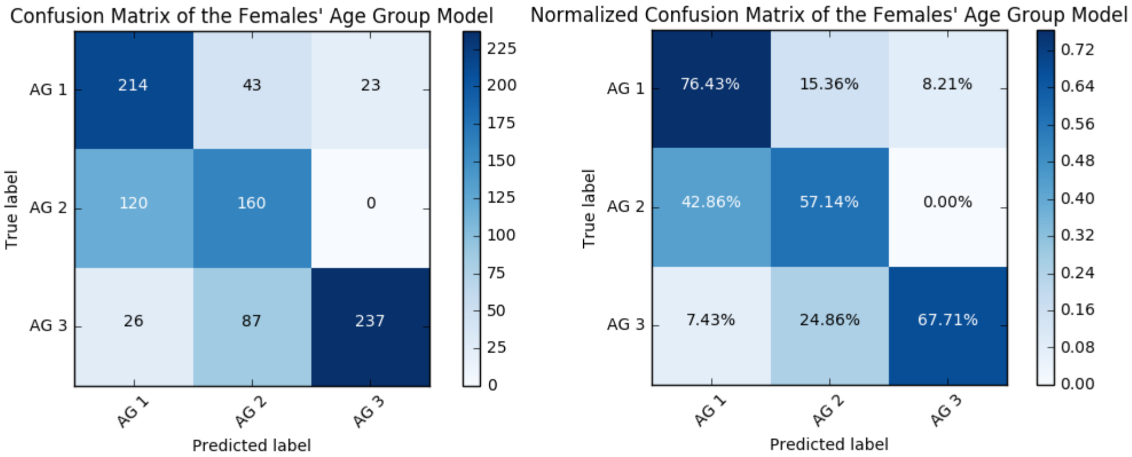


Figure 39 Confusion matrix and normalized confusion matrix of the females' age group model on validation data.

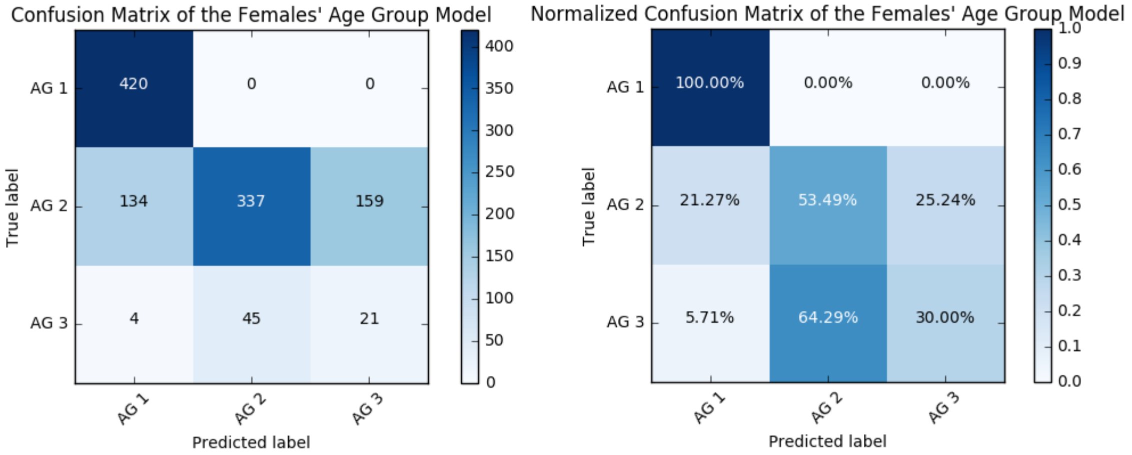


Figure 40 Confusion matrix and normalized confusion matrix of the females' age group model on test data.

The females' age group model has bad accuracy on identifying the voice files from third age group on females' test data and has recall as little as 30.0% and precision of 11.7% for this age group. The reason could be that there are voice files from only one 16 years old female speaker in the third age group of the test data. On the other hand, on females' validation data, it does a little bit better by having recall of 67.7% and precision of 91.2% for the third age group. The model does not make any mistakes, while identifying the female speakers from the first age group of test data, but has precision of 75.3% for that age group, while the corresponding recall and precision on validation data are 76.4% and 59.4%. Like the males' age group model, the females' model also has problems on identifying speaker from the second age group, by having recall under 60% on both of the datasets. On the other hand, it still makes more correct predictions than wrong predictions on both of the datasets for the speakers from that age group.

5.5 Connecting Models for Boosting the Accuracy on Test Data

To boost the age group and gender identification accuracies on test data, ensemble models were built by experimenting with different combinations of the built models that had highest accuracies on test data.

5.5.1 Boosting the Males’ Age Group Identification Accuracy

By trying out different combinations of the models that had highest accuracies on test data for males’ age group identification, majority voting did not increase the accuracy on test data, while averaging did. By combining four of the best models for males’ age group identification through averaging their predictions, the accuracy increased up to 85.6% on test data. The details about this ensemble model are brought out in Appendix XII. The same model has accuracy of 85.0% on validation data, which is also better compared to the same accuracies of the models brought out in section 5.4.1.

5.5.2 Boosting the Females’ Age Group Identification Accuracy

For females’ age group identification, majority voting and averaging both increased accuracy on test data. Using four of the built models, majority voting increased the females’ age group identification more significantly to 71.8% on test data. The same combination of models achieved accuracy of 69.2% on validation data. Both of these accuracies are over 2 percentage points higher compared to the accuracies of the females’ age group model that was brought out in section 5.4.2. The details about the ensemble model that uses majority voting are brought out in Appendix XIII.

5.5.3 Boosting the Gender Identification Accuracy

Different combinations with the models that had highest accuracies of identifying gender on test data were made, but the accuracy did not increase. The model that was brought out in section 5.1.1 with the accuracy of 92.8% remained to have the highest accuracy on gender identification on test data.

5.5.4 Boosting the Age Group Identification Accuracy

For age group identification, averaging the predictions of the built models worked better than majority voting by increasing the age group identification accuracy up to 74.1%, compared to the accuracy of 73.5% of the single most accurate model brought out in section 5.2.3. On validation data, the ensemble model has accuracy of 76.6%, compared to the 74.4% of the single multi-input model. The details about the combined model that uses averaging are brought out in Appendix XIV.

Secondly, models were combined by first predicting the gender with the most accurate model for gender identification from section 5.1.1. Depending on the gender prediction, either the best model for males' age group identification from section 5.4.1 or the most accurate model for females' age group identification from section 5.4.2 was used to predict the age group of the given subject. By combining the mentioned models, the accuracy of age group identification increased up to 75.8% on test data, while the same combination of models has accuracy of 74.4% on validation data. Furthermore, model was built by using the weights of the gender prediction for each subject to give weight for males' and females' age group models as described in section 4.2.3. By using the weights of the gender predictions, accuracy increased up to 75.6% on test data. On validation data, the same model has accuracy of 75.1%.

In addition, instead of using single gender specific model for age group identification, combined model was built by using the best ensemble models from sections56.5.1 and 5.5.2. After making the gender prediction, the most accurate ensemble model that uses majority voting was used for females' age group identification and the most accurate model that was gotten through averaging was used for males' age group identification. By using the gender predictions as weights the accuracy of the model on validation data is 75.6% and 75.8% on test data. Without using the gender predictions as weights the accuracy of this model on validation data is 76.0% and on test data 76.3%, which is the highest accuracy for age group identification on test data out of all the built models. Figure 41 shows the confusion matrix and the normalized confusion matrix of the most accurate model's predictions on test data.

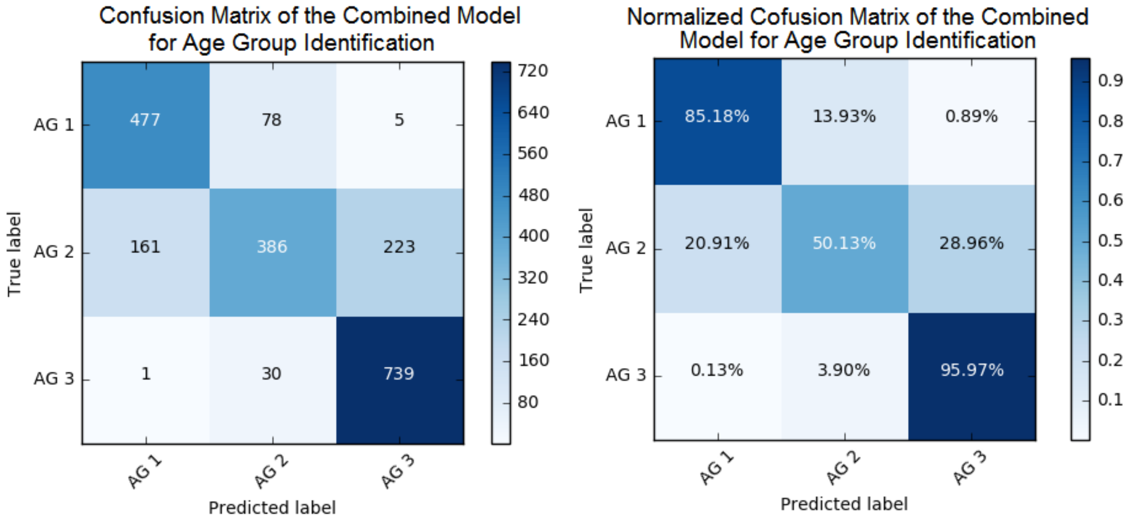


Figure 41 Confusion matrix and normalized confusion matrix of the most accurate age group model on test data.

Figure 41 shows that the main difference compared to the other models for age group identification comes from the high recall of 85.2% for identifying speakers from the first age group. The precisions for all of the age groups are the highest for this model, except from the precision of 74.7% for the first age group, which is only under 1.5% better for the multi-input age group model. The recall of 50.13% for the second age group is also one of the best and is only higher for the multi-input model, but the precision of 78.1% for the same age group is over 4% higher than for any other model. The precision of identifying the speakers from the third age group is 76.4%.

5.6 Comparison to the Baseline Method

Compared to the i-vector baseline method, the built models achieved higher overall accuracies for age group and gender identification on test data, while the highest accuracies on validation data were very similar. For age group identification, the implemented baseline method achieved accuracy of 77.4% on validation data and 71.7% on test data. Figure 42 brings out the confusion matrix and the normalized confusion matrix of the predictions of the baseline age group system on test data.

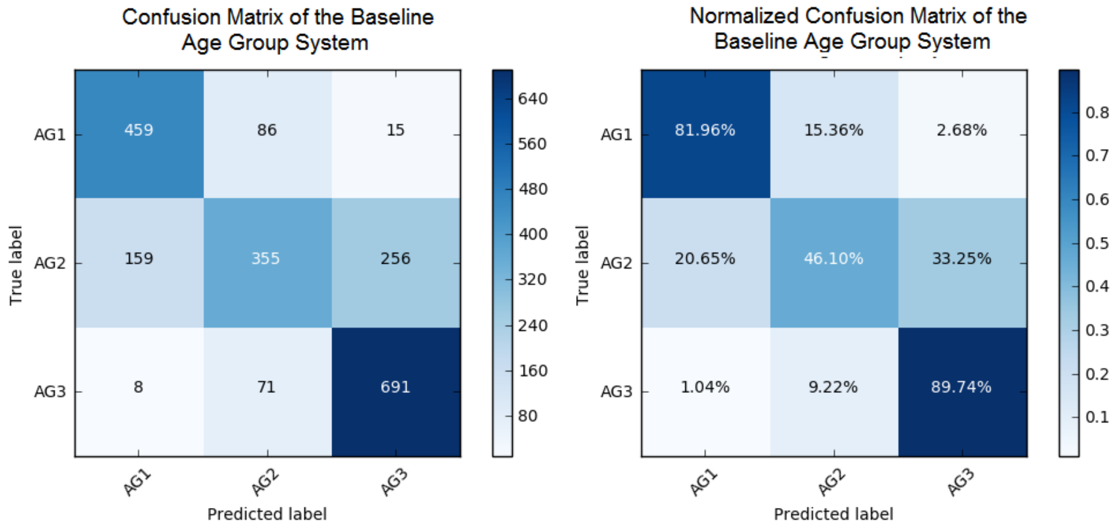


Figure 42 Confusion matrix and normalized confusion matrix of the baseline age group system on test data.

The confusion matrices show that the baseline method also makes most mistakes on identifying speakers from the second age group. The recalls of 82% and 46.1% for identifying speakers from the first and second age group are very similar to the recalls of the models brought out in sections 5.2 and 5.3. Compared to the combined model brought out in section 5.5.4, they are over 3 percentage points lower. The recall of 89.7% of the baseline model for the third age group is relatively low: most of the models brought out in this research have recall higher than 90% for the same age group. While the baseline method’s precision of 73.3% for identifying the speakers from the first age group is quite similar to the one of the combined model, the precisions for identifying speakers from the second and third age group are over 4 percentage points lower.

For gender identification, the same method achieved 88.8% of accuracy on validation data and 90.2% on test data. Figure 43 shows the confusion matrices of the baseline gender system on test data.

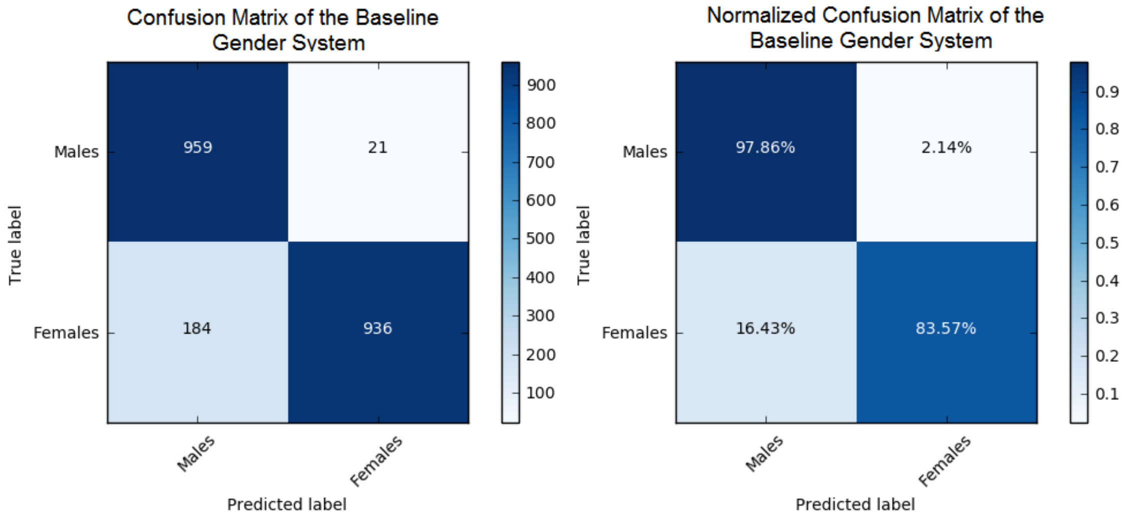


Figure 43 Confusion matrix and normalized confusion matrix of the baseline gender system on test data.

Compared to the feedforward neural network with highest accuracy on test data from section 5.1.1, the baseline system's recall of 97.9% for identifying males is 1.5 percentage points higher. The overall accuracy of the baseline system is lower, because of the low recall of 83.6% for identifying females. The baseline system's precision of 83.8% for identifying males is over 5 percentage points lower, but the precision of 97.8% for identifying females is 1.3 percentage points bigger compared to the most accurate neural network for gender identification.

5.7 Conclusions of the Results

Most of the different types of built models have similar accuracies for identifying subjects in different ages. Figure 44 brings out the accuracies of different gender identification models on test data for speakers in different ages

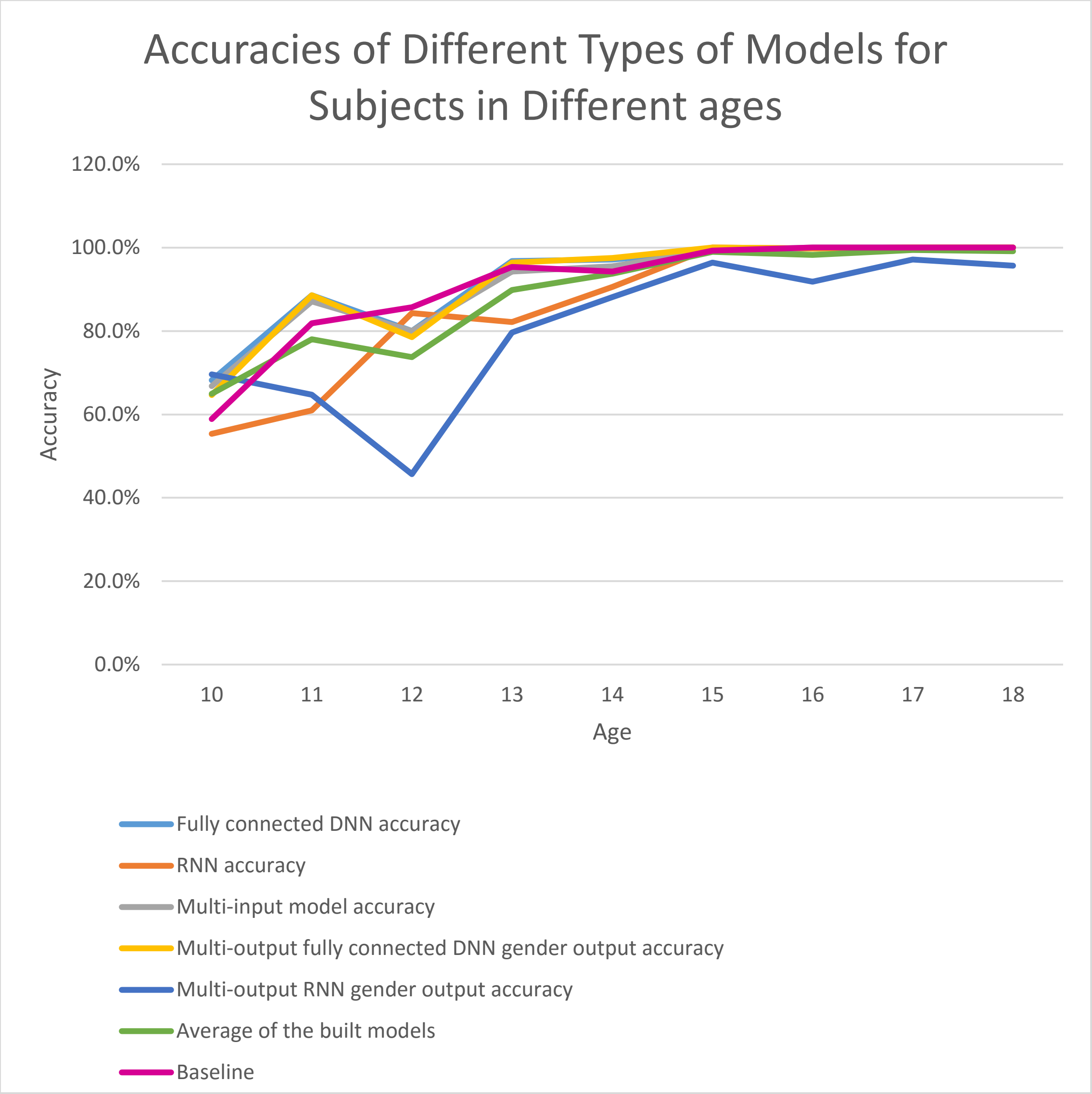


Figure 44 The accuracies of different types of models for subjects in different ages.

All of the models work better for older subjects. The reason behind this is that older male subjects have been through puberty and their voice is lower compared to the female subjects of same age. Figure 44 shows that the RNN models differ from feedforward DNN models and multi-input models. The RNN with single output has lower accuracy for most of the younger speakers compared to all other models. While the accuracies of all the other models for identifying the gender of 11 years old subjects are lower than for 12 years old subjects, the accuracy of the RNN and the baseline system increase between those ages. The accuracy of the multi-output RNN changes the most and the line that shows the predictions of that model on Figure 44 is unstable similarly to the line of the accuracy changes shown on Figure 32 for the same model. From all of the brought out models in this research, the multi-output RNN is the only model that does not have convolutional layers in it, which has effect on both the training process and performance of the model. The convolutional layers in the beginning of recurrent neural networks smoothen the training process by bundling the information from the multiple neighbouring mel-frequency cepstral coefficients for the LSTM layer. Figure 44 also shows that the identification accuracies of the multi-input model are very similar to the feedforward models. This indicates that the multi-input model fails to gain useful information from the mel-frequency cepstral coefficients and makes predictions similarly to the models that take only i-vectors as input.

Models that make predictions based on i-vectors outperform the models that make predictions based on only mel-frequency cepstral coefficients. Table 5 and Table 6 show accuracies of gender and age group identification of different types of models.

Table 5 Different models’ accuracies of identifying the gender of a subject on validation and test data.

Type of Model	**Accuracy on validation data**	**Accuracy on test data**
Feedforward DNN	87.0%	92.8%
RNN	86.4%	89.2%
Multi-input model	87.1%	91.8%
Multi-output feedforward DNN gender output	86.6%	92.3%
Multi-output RNN gender output	79.2%	82.8%
Baseline i-vector system	88.8%	90.2%

Table 6 Different models’ accuracies of identifying the age group of a subject on validation and test data.

Type of Model	**Accuracy on validation data**	**Accuracy on test data**
Feedforward DNN	77.8%	72.1%
RNN	74.1%	71.2%
Multi-input model	74.4%	73.5%
Multi-output feedforward DNN age group output	75%	73.1%
Multi-output RNN age group output	70.9%	71.8%
Combined model with highest accuracy on test data	76.0%	76.3%
Baseline i-vector system	77.4%	71.7%

Without considering the combined model, the accuracies of the age group identification models on test data differ 2.3% in absolute, while the same accuracies of the gender identification models differ 10% in absolute. The multi-output RNN is 6.4 percentage points worse than the second worst model for gender identification on test data. On the other hand, for age group identification, the same model has a little bit higher accuracy on test data than the single output RNN. On validation data, the model is still 3.2 percentage points worse than the second worst model by accuracy. The reason behind relatively bad performance of the RNN-s may be that instead of learning to generalize from the data, they learned subject specific information. As there were many voice files from same speakers, RNN-s may have learned to identify each subject, instead of learning to identify the age groups. I-vector is a kind of generalization made from the mel-frequency cepstral coefficients and they seem to make it easier for neural networks to generalize from training data to identify the age group and gender of a speaker.

Most of the models that had highest accuracies on validation data did not have the highest accuracies on test data. For example, one of the built multi-output feedforward DNN models had accuracy of 79.2% for age group identification and 87.4% for gender identification on validation data. Both of these measures are highest in their respective categories on validation data. The same model had accuracy of 70.7% for age group identification and 91.3% for gender identification on test dataset, which are not among the top three accuracies in their respective categories. Almost all the built models for age group identification had higher accuracies on validation data, compared to the accuracies on test data. As the gap for some of the models was over 8%, then it shows that the datasets contain voice files from really different subjects and it is hard to make a model that would generalize similarly on both of the datasets. The combined models had more similar results on validation and test data and most of them had even a bit higher accuracy on test dataset. The reason behind this is that they were combined by trying to maximize the accuracy on test data. The most accurate model for age group identification on test data is a combined model that has similar accuracies on both of the datasets.

In most of the categories, where the models that output the exact age and the models that predict the age group directly were built, the models that try to identify the exact age had higher accuracy for age group identification. The reason behind this could be that the models that predict exact age have information that is ordered and therefore have more data to learn from. The single output RNN and multi-input model were the only types of models that failed to achieve better accuracies, while predicting the exact age.

As the second age group is the smallest, containing speakers only in two ages, and the time of puberty voice changes of male speakers can vary, then the built models often wrongly identified the age group of speakers from that group. The built models confused speakers from second age group with speakers from third rather than first age group. Figure 8 shows that the puberty voice changes of male speakers can happen later, when the speaker is already in the third age group.

As was brought out in section 5.4, it is harder for models to predict the age group of female than male speakers. They do not have similar puberty voice mutation to male speakers and therefore their voice lowers gradually like shown on Figure 9. The fact that the most accurate model for identifying the females' age groups was trained on the whole training data shows that there is not enough data from female speakers, to make a model that would generalize well only by training on the separated females' dataset.

5.8 Threats to the Validity

As the study in hand involved building and analysis of over 100 models, there is a threat of misinterpretation. As most of the built models' parameters were saved together with their training history, the amount of data to consider in analysis was big and something may have been missed. Mistakes in code could have led to unwanted data changes and wrong results.

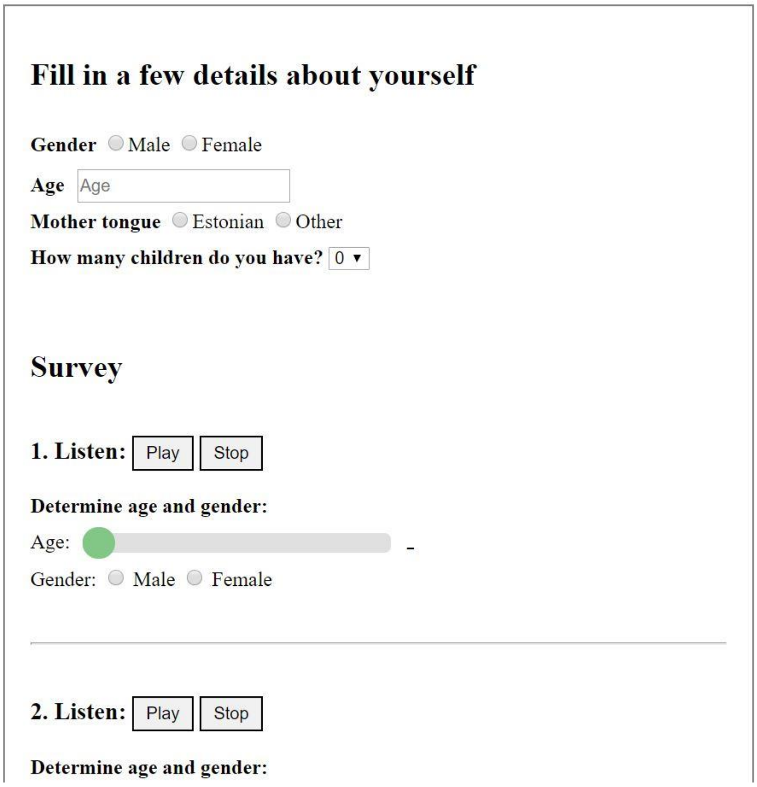
In addition, the research was done on a dataset that contained voice files from 300 different speakers from 4 different dialectal areas collected in similar manner. The variety of the dataset was limited and there is a threat that the built models do not perform similarly on new data.

6 Human Performance

After experiments on age group and gender identification with neural networks, a survey was conducted among real human listeners. The goal of the survey was to find out human accuracy on age group and gender identification on test data. To this end, first nine voice files from each subject in test data were excluded, as they contained recordings about personal information. This chapter brings out the results of the survey, as well as key differences between the performances of neural networks and human listeners on the remaining of the test data.

6.1 Survey Web Application

As the author of this research did not find any good available solutions to conduct the survey, a web application was built. The application was deployed to Heroku [43] platform and the voice files were stored on a Raspberry Pi web server. Figure 45 shows a screenshot taken from the built web application.



Fill in a few details about yourself

Gender ○ Male ○ Female

Age Age

Mother tongue ○ Estonian ○ Other

How many children do you have? 0 ▾

Survey

1. Listen: Play Stop

Determine age and gender:

Age: ● -

Gender: ○ Male ○ Female

2. Listen: Play Stop

Determine age and gender:

Figure 45 A screenshot taken from the built web application.

The participants could choose, whether they wanted to fill in the survey in English or Estonian. They were asked to first insert some information about themselves like the age, number of children they have, age of each child and whether their mother tongue is Estonian or not. After that, they had to predict the exact age and gender of 10 different speakers based on a randomly selected voice files out of the 1830 voice files included in the survey.

6.2 Results

In total 64 people between ages of 21 and 75 filled in the survey of whom 34 were males and 30 were females. 58 of the participants marked Estonian as their mother tongue and 14 marked that they have one or more child. Table 7 shows the accuracy of human listeners on different tasks, brings out the accuracy of the most accurate combined model brought out in section 5.5.4 for the age group identification tasks and the accuracy of the most accurate gender identification model from section 5.1.1 for gender identification on the survey dataset.

Table 7 Accuracies from the conducted survey compared to the most accurate ensemble model and gender model accuracies.

Task	**Accuracies of human listeners**			**Model accuracy**
	All participants	**Males**	**Females**	
Age group identification	64.7%	62.6%	67.0%	76.0%
Males' age group identification	78.8%	75.2%	83.1%	83.0%
Female's age group identification	52.5%	51.4%	53.7%	69.9%
Gender identification	85.6%	87.4%	83.7%	92.5%

The overall age group determination accuracy was 64.7% which is over 10 percentage points lower than the accuracy of 76% of the most accurate model for age group identification that was brought out in section 5.5.4. The difference in the age group identification accuracy comes mainly from identifying the age groups of female speakers. Figure 46 shows normalized confusion matrices of humans' age group predictions and the combined model's predictions on the dataset used in the survey.

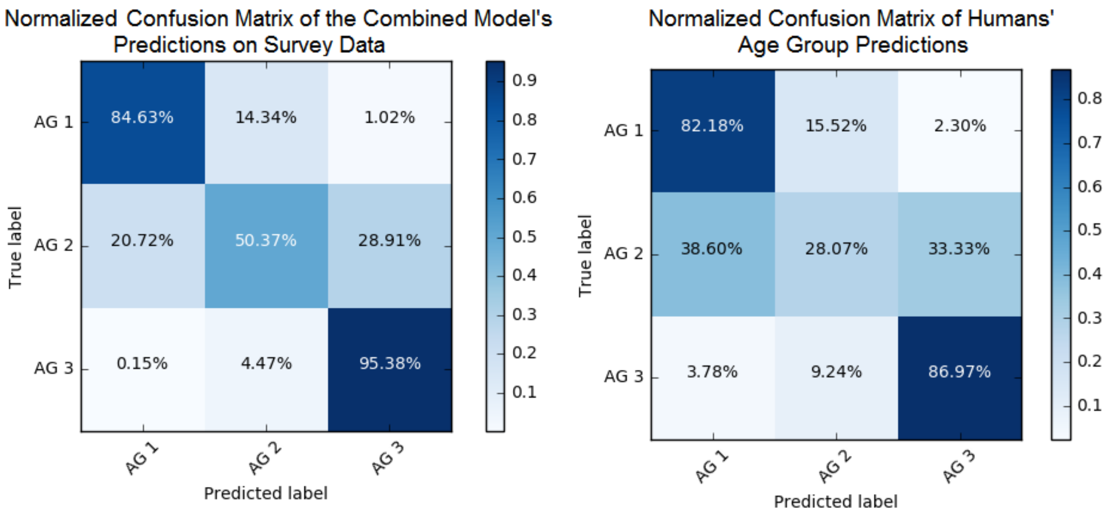


Figure 46 Normalized confusion matrices of the most accurate model's age group predictions and humans' age group predictions

Confusion matrices show that humans have a recall of 28.1% on identifying speakers from the second age group, which is over 20 percentage points lower compared to the recall of 50.37% of the most accurate model. While the recalls for identifying speakers from the first age group are very similar, the recall of identifying speakers from third age group is over 10 percentage points lower for human listeners. Table 8 brings out mean absolute errors for humans' predictions. In addition, the mean absolute errors for the most accurate model on test data that identifies the exact age and was brought out in section 5.3.1 are shown.

Table 8 Mean absolute errors for identifying speakers from different age groups.

	Mean absolute error of human listeners	**Model mean absolute error**
AG 1	1.1	1.2
AG 2	1.7	1.1
AG 3	1.4	0.6
Overall	1.4	0.9

The human listeners achieved accuracy of 85.6% for gender identification, while the most accurate model for gender identification brought out in section 5.1.1 has accuracy of 92.5% on the same dataset. Figure 47 shows normalized confusion matrices of the humans' gender predictions and the most accurate model's predictions from section 5.1.1 on the survey dataset.

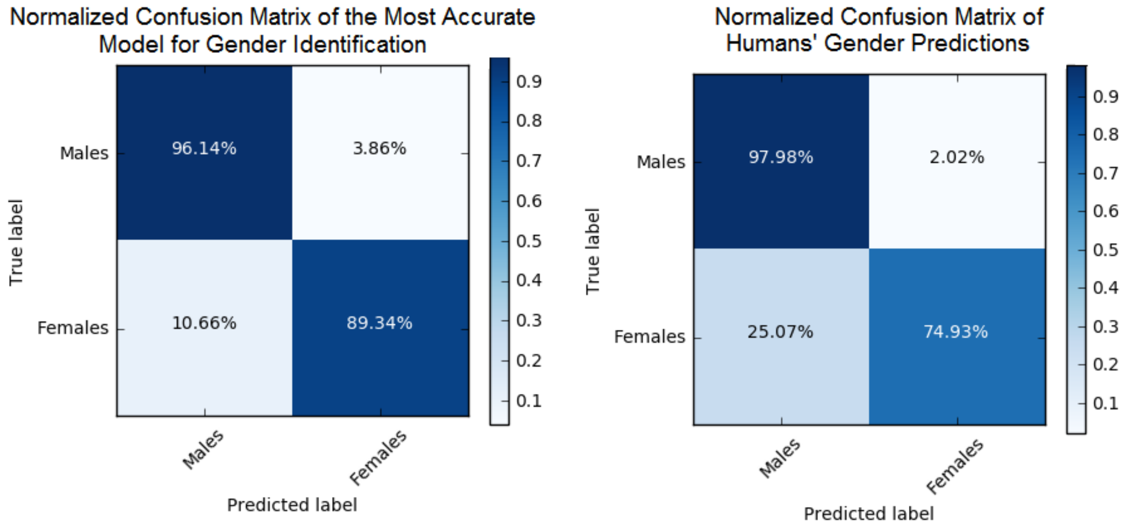


Figure 47 Normalized confusion matrices of the most accurate model's gender predictions and humans' gender predictions

Confusion matrices show that the recall for identifying male speakers is nearly 2 percentage points bigger for human listeners, but the recall for identifying female speakers is over 14 percentage points lower. The precisions of the human predictions for both of the genders are over 10 percentage points lower compared to the precisions of the neural network. Human listeners made most of the mistakes by wrongly identifying female speakers as males.

There was no significant difference between the age group and gender determination accuracies of participants, who were over 40 compared to the ones under 40. Female and male participants achieved similar results. Also having children did not have effect on the accuracies The number of participants, who marked other language than Estonian as their mother tongue was too low to make any conclusions.

7 Applications and Future Work

In this section possible fields of usage and future work are brought out.

7.1 Application

The following paragraphs bring out possible fields of usage of the built models.

7.1.1 Content Adaption

Content adaption is used in many modern applications to show relevant ads and give suggestions based on a user. Age and gender of a speaker can give useful information to show relevant content. For instance, in a movie viewing application, age identification can be used to restrict access to certain content. In addition, age and gender can be used to show relevant information in applications used in studies.

7.1.2 Child protection

Some laws protect children under specific age. Children may not evaluate situations correctly and they are in danger of being abused. In addition to restricting child's access to some content, age and gender identification can be used to protect children from adults pretending to be children in real life conversations or social media.

7.1.3 Improve Performance of Other Tasks

Age and gender recognition can be used together with other methods for subject identification related tasks. Built models could improve speaker recognition and user identification by considering additional information from the subject. Gender identification helped to improve age group identification accuracy in this research and could help to improve speech recognition as well.

7.2 Future Work

The following paragraphs bring out ways to improve the built models and their usability.

7.2.1 Models for Different Number of Age Groups

Figure 8 shows that most of the male speakers' voice mutation happens between ages 13 and 14. In addition to experiments conducted in this research, two age groups could be used to build models for identifying, whether the speaker is over or under 14 years of age. This could lead to higher accuracies compared to the models built in this research and could be useful for child protection purposes.

7.2.2 Combined Model for Age Identification

Another thing to consider is to combine the models in a way that the output would be exact age instead of an age group. Right now the predictions of the models that predict exact age are mapped to their respective age groups. This limits the possible usage fields of the models. An ensemble model could be built, where for the predictions of the age group models, the average of the ages of the speakers in that specific age group is considered.

7.2.3 Identifying Gender and Age Group Based on a Voice File

To simplify the usage of the built models an application could be built, that first transforms a voice file into sequences of MFCC-s or i-vectors and then uses built models to identify the exact age, age group or gender of a speaker. This kind of an application could be open-sourced that, in the end, could lead to improved models based on the ideas from different contributors.

7.2.4 Training and Testing of the Built Models on Different Dataset

Experiments could be conducted on additional data from Estonian and foreign speakers. This could help improve the performance and generalization power of the built models. In addition to gathering data by recording voice from real life conversations, available data from already made recordings like podcasts could be used.

7.2.5 Using Additional Information to Improve the Models

In order to improve the models, additional information about a subject could be used. For example, image of the subject could be used to build neural networks that could be combined together with the built models in this research for gender identification. In addition, the accuracy of the models could be improved if the name of a subject is known, as it can provide additional information about the gender of a subject.

8 Conclusion

The goal of this thesis was to analyse the performance of different types of neural networks for gender and age identification based on children's speech. I-vectors were used as input for feedforward neural networks and mel-frequency cepstral coefficients were used as input for recurrent neural networks with convolutional and LSTM layers.

Out of the built models, a feedforward DNN had the highest accuracy of 92.8% on test data for gender identification, which is over 2.6 percentage points higher compared to the baseline system accuracy on the same dataset. For age group identification better results were achieved by predicting the exact age of the speaker, which was mapped to the corresponding age group. Without combining models, the highest accuracy of 73.5% for age group identification on test data was achieved with multi-input model, that took both i-vectors and mel-frequency cepstral coefficients as input. The complexly structured RNN-s did not outperform simple feedforward DNN-s.

To boost the age group identification accuracy on test data, models were combined by firstly identifying the gender of a given subject. After that, an ensemble model for males' or females' age group identification was used based on the output of the gender identification model. That led to the highest overall accuracy of 76.3% for age group identification on test data, which is 4.6 percentage points higher compared to the age group identification accuracy of the baseline system on the same dataset.

Due to the voice mutation of male speakers, the accuracy of 85.1% achieved for identifying the age group of male speakers was higher than the accuracy of 69.5% for identifying the age group of female speakers. The built neural networks made most mistakes when identifying the age group of male speakers between 12-15 years, which are the most common ages for puberty voice mutation in the dataset of native Estonian subjects used in this research. For female speakers, the most accurate model made majority of its mistakes on identifying the age group of the children between 15 and 18 years of age. At these ages the F0 statistics of female speakers is smoothened and the voice tones do not differ much from the speakers between 13 and 14 years of age.

In the end, a survey was conducted to measure human performance on the given tasks. Participants achieved accuracy of 64.7% for age group identification and 85.6% for gender identification on reduced test data, where voice files containing personal information were excluded. Both of these accuracies are about 9 percentage points lower compared to the accuracies of the most accurate built models on the same dataset.

All in all, the built neural networks outperformed real humans on gender, age and age group identification. In addition, higher accuracies were achieved on gender and age group identification, compared to the chosen baseline method. The built models can be used to protect children, improve child-computer interaction systems and possibly improve the performance of other speech related tasks, such as speech or speaker recognition.

References

[1] S. E. Shepstone, Z.-H. Tan and S. H. Jensen, “Audio-based Age and Gender Identification to Enhance the Recommendation of TV Content,” *IEEE Transactions on Consumer Electronics,* vol. 59, no. 3, pp. 721-729, 2013.

[2] S. Safavi, M. J. Russel and P. Jančovič, “Identification of Age-Group from Children’s Speech by Computers and Humans,” in *INTERSPEECH,* Singapore, Republic of Singapore, 2014.

[3] T. Bocklet, A. Maier, J. G. Bauer, F. Burkhardt and E. Nöth, “Age and gender recognition for telephone applications based on GMM supervectors and support vector machines,” in *IEEE International Conference on Acoustics, Speech and Signal Processing,* Las Vegas, USA, 2008.

[4] Y. Wang and B. Lawlor, “Speaker recognition based on MFCC and BP neural networks,” in *Signals and Systems Conference (ISSC),* Killarney, 2017.

[5] N. Chauhan and M. Chandra, “Speaker recognition and verification using artificial neural network,” in *Wireless Communications, Signal Processing and Networking (WiSPNET),* Chennai, 2017.

[6] A. Graves, A.-r. Mohamed and G. Hinton, “Speech recognition with deep recurrent neural networks,” in *Acoustics, Speech and Signal Processing (ICASSP),* Vancouver, 2013.

[7] L. Meister and E. Meister, “Development of the Corpus of Estonian Adolescent Speech,” *The Baltic Perspective,* pp. 243-247, 2014.

[8] K. Shobaki, J. P. Hosom and R. A. Cole, “The ogi kids' speech corpus and recognizers.,” in *INTERSPEECH,* Bejing, 2000.

[9] D. Reynolds, “Universal Background Models,” in *Encyclopedia of Biometrics,* Lexington, Massachusetts: Springer US, 2015, pp. 1547-1550.

[10] E. Khoury and M. Garland, “I-Vectors for Speech Activity Detection,” in *Odyssey,* Bilbao, Spain, 2016.

[11] N. Dehak, P. A. Torres-Carrasquillo, D. A. Reynolds and R. Dehak, “Language Recognition via I-Vectors and Dimensionality Reduction,” in *INTERSPEECH,* Florence, Italy, 2011.

[12] D. Garcia-Romero and C. Y. Espy-Wilson, “Analysis of I-vector Length Normalization in Speaker Recognition Systems,” in *INTERSPEECH,* Florence, Italy, 2011.

[13] D. Snyder, D. Garcia-Romero and D. Povey, “Time delay deep neural network-based universal background models for speaker recognition,” in *Automatic Speech Recognition and Understanding,* Scottsdale, USA, 2015.

[14] C. S. Greenberg, D. Bansé, G. R. Doddington, D. Garcia-Romero, J. J. Godfrey, T. Kinnunen, A. F. Martin, A. McCree, A. Przybocki and D. A. Reynolds, “The NIST 2014 speaker recognition i-vector machine learning challenge Authors,” in *Odyssey: The Speaker and Language Recognition Workshop,* Joensuu, Finland, 2014.

[15] M. Nielsen, “Neural networks and deep learning,” Determination Press, 2015.

[16] Y. Bengio, “Learning Deep Architectures for AI,” *Foundations and Trends® in Machine Learning,* vol. 2, no. 1, pp. 1-127, 2009.

[17] I. Aniemeka, “A Friendly Introduction to Convolutional Neural Networks | Hashrocket,” 22 9 2017. [Online]. Available: https://hashrocket.com/blog/posts/a-friendly-introduction-to-convolutional-neural-networks. [Accessed 23 4 2018].

[18] S. Hochreiter and J. Schmidhuber, “Long Short-term Memory,” *Neural Computation,* vol. 9, no. 8, pp. 1735-1780, 1997.

[19] F. A. Gers, J. A. Schmidhuber and F. A. Cummins, “Learning to Forget: Continual Prediction with LSTM,” *Neural Computation,* vol. 12, no. 10, pp. 2451 - 2471, 2000.

[20] C. Olah, “Understanding LSTM Networks -- colah's blog,” 27 9 2015. [Online]. Available: http://colah.github.io/posts/2015-08-Understanding-LSTMs/. [Accessed 4 29 2018].

[21] K. Mishra, R. Sathish and D. Sheet, “Tracking of Retinal Microsurgery Tools Using Late Fusion of Responses from Convolutional Neural Network over Pyramidally Decomposed Frames,” in *Computer Vision, Graphics, and Image Processing,* Guawahati, India, 2016.

[22] Z. Qawaqneh, A. Abumallouh and B. D. Barkana, “Deep Neural Network Framework and Transformed MFCCs for Speaker's Age and Gender Classification,” *Knowledge-Based Systems,* vol. 115, pp. 5-14, 2016 .

[23] M. H. Bahari, M. McLaren, H. V. Hamme and D. A. van Leeuwen, “Speaker age estimation using i-vectors,” *Engineering Applications of Artificial Intelligence,* vol. 34, no. C, pp. 99-108, 2014.

[24] A. Fedorova, O. Glembek, T. Kinnunen and P. Matejka, “Exploring ANN Back-Ends for i-Vector Based Speaker Age Estimation,” in *INTERSPEECH,* Dresden, Germany, 2015.

[25] J. Sas and A. Sas, “Gender recognition using neural networks and ASR techniques,” *Journal of Medical Informatics & Technologies,* vol. 22, pp. 179-187, 2013.

[26] S. Khanum and M. Sora, “Speech based Gender Identification using Feed Forward Neural Networks,” in *IJCA Proceedings on National Conference on Recent Trends in Information Technology,* Karad, 2016.

[27] H. Harb and L. Chen, “Voice-Based Gender Identification in Multimedia Applications,” *Journal of Intelligent Information Systems,* vol. 24, no. 2-3, pp. 179-198, 2005.

[28] S. Safavi, M. J. Russel and P. Jancovic, “Identification of Gender from Children’s Speech by Computers and Humans,” in *INTERSPEECH,* Lyon, France, 2013.

[29] J. Gonzalez-Dominguez, I. Lopez-Moreno, H. Sak and P. J. Moreno, “Automatic language identification using Long Short-Term Memory recurrent neural networks,” in *INTERSPEECH,* Singapore, Republic of Singapore , 2014.

[30] A. Eek and E. Meister, “Estonian speech in the BABEL multi-language database: Phonetic-phonological,” in *Proceedings of LP’98,* Prague: Karolinum Press, 1999.

[31] Z. Zhang, “Mechanics of human voice production and control,” *The Journal of the Acoustical Society of America,* vol. 140, no. 4, p. 2614–2635, 2016.

[32] E. Meister and L. Meister, “Estonian adolescent speech I. Acoustic analysis of fundamental frequency,” *Keel ja Kirjandus,* vol. 60, no. 7, pp. 518 - 533, 2017.

[33] “GitHub - kaldi-asr/kaldi: This is now the official location of the Kaldi project.,” [Online]. Available: https://github.com/kaldi-asr/kaldi. [Accessed 20 3 2018].

[34] “Python Data Analysis Library — pandas: Python Data Analysis Library,” [Online]. Available: https://pandas.pydata.org/. [Accessed 20 3 2018].

[35] “EENet » Kobararvuti ressursid,” [Online]. Available: http://www.eenet.ee/EENet/grid_ressursid.html. [Accessed 20 3 2018].

[36] “Home · tmux/tmux Wiki · GitHub,” [Online]. Available: https://github.com/tmux/tmux/wiki. [Accessed 21 3 2018].

[37] “Keras Documentation,” [Online]. Available: https://keras.io/. [Accessed 20 3 2018].

[38] “Guide to the Functional API - Keras Documentation,” [Online]. Available: https://keras.io/getting-started/functional-api-guide/. [Accessed 21 3 2018].

[39] “Getting started with the Keras Sequential model,” [Online]. Available: https://keras.io/getting-started/sequential-model-guide/. [Accessed 21 3 2018].

[40] C. Baziotis, “Keras Layer that implements an Attention mechanism, with a context/query vector, for temporal data. Supports Masking. Follows the work of Yang et al. [https://www.cs.cmu.edu/~diyiy/docs/naacl16.pdf] "Hierarchical Attention Networks for Document Classification",” [Online]. Available: https://gist.github.com/cbaziotis/7ef97ccf71cbc14366835198c09809d2. [Accessed 21 3 2018].

[41] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens and Z. Wojna, “Rethinking the Inception Architecture for Computer Vision,” *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* pp. 2818-2826, 2016.

[42] G. Hinton, O. Vinyals and J. Dean, “Distilling the Knowledge in a Neural Network,” *arXiv:1503.02531 ,* 2015.

[43] “Cloud Application Platform | Heroku,” Salesforce, [Online]. Available: https://www.heroku.com/. [Accessed 1 5 2018].

[44] L. S. Sterling, The Art of Agent-Oriented Modeling, London: The MIT Press, 2009.

[45] N. Dehak, J. K. Patrick, R. Dehak, P. Dumouchel and P. Ouellet, “Front-End Factor Analysis for Speaker Verification,” vol. 19, no. 4, pp. 788-798, 2011.

Appendix I – Code

The code produced in this research can be accessed from Github https://github.com/leopiel/mastersthesis. The code in the repository cannot be run, because training, validation and test data cannot be uploaded to this repository due to copyright and large file sizes.

Appendix II – The Most Accurate Feedforward DNN for Gender Identification

This appendix gives details about a feedforward DNN that predicts only gender and had the highest accuracy on test data among all the built models. The model can be found from dnn_3.py in the Github repository. The features with corresponding values of the model are brought out in the following Table 9, with some of the values taken from Keras documentation [37].

Table 9 Structure of the feedforward DNN that identifies gender of the speaker.

Feature	**Property**	**Value**
1. hidden layer	Type of layer	Dense
	Size	4000 neurons
	Activation function	relu
	Dropout	0.5
2. hidden layer	Type of layer	Dense
	Size	2000 neurons
	Activation function	relu
	Dropout	0.4
Output layer	Type of layer	Dense
	Size	2 neurons
	Activation function	softmax
	Loss function	categorical_crossentropy
Optimizer		sgd
Learning rate		0.001
Epochs		50
Batch size		32

Appendix III – The Most Accurate RNN for Gender Identification

This appendix brings out the details about a RNN model, that is built for gender identification and has the highest accuracy out of the built gender RNN models on test data set. The model can be found from file gender_9.py in the Github repository. Table 10 shows the structure of the RNN, with some of the values taken from Keras documentation [37].

Table 10 Structure of the RNN that predicts gender of the speaker.

Feature	Property	Value
1. hidden layer	Type of layer	Conv2D
	Filters	128
	Size of convolutional window	3x20
	Activation function	relu
	border_mode	valid
2. hidden layer	Type of layer	Conv2D
	Filters	128
	Size of convolutional window	3x1
	Activation function	relu
	border_mode	valid
3. hidden layer	Type of layer	Conv2D
	Filters	128
	Size of convolutional window	3x1
	Activation function	relu
	border_mode	valid
4. hidden layer	Type of layer	Conv2D
	Filters	128
	Size of convolutional window	3x1
	Activation function	relu
	border_mode	valid
5. hidden layer	Type of layer	Reshape
	Target shape	(-1, 128)

Feature	**Property**	**Value**
6. hidden layer	Type of layer	Bidirectional LSTM
	Size	64
	return_sequences	True
7. hidden layer	Type of layer	Bidirectional LSTM
	Size	64
	return_sequences	True
8. hidden layer	Type of layer	AttentionWithContext
Output layer	Type of layer	Dense
	Size	2 neurons
	Activation function	softmax
	Loss function	categorical_crossentropy
Optimizer		sgd
Learning rate		0.001
Epochs		50
Batch size		128

Appendix IV – The Most Accurate Multi-Input Model for Gender Identification

In this appendix, the details about the best multi-input gender model by accuracy on test data are brought out. Figure 48 shows the structure of this multi-input model.

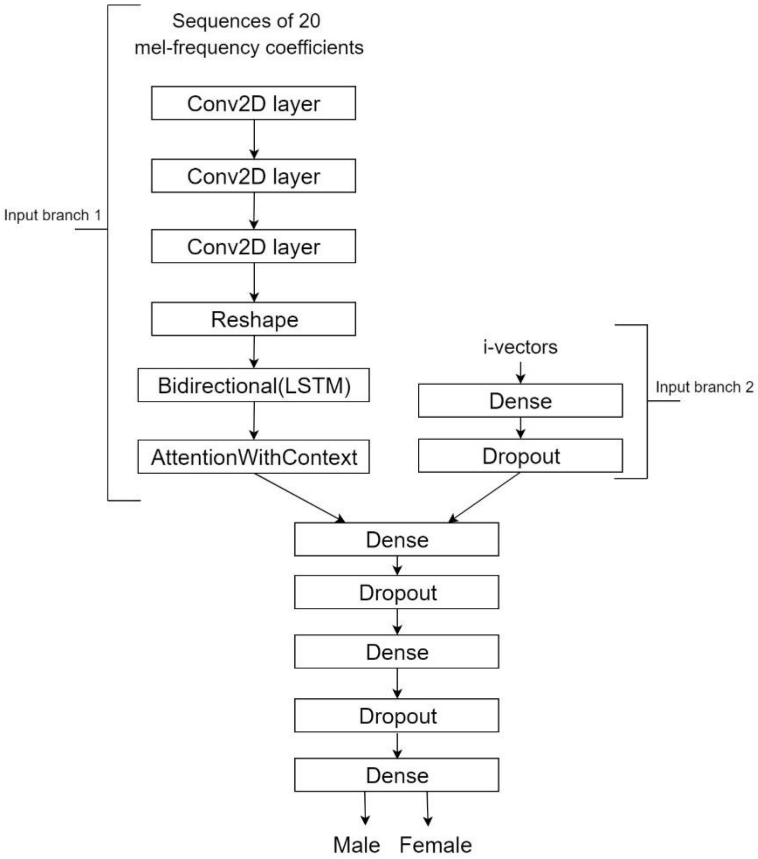


Figure 48 Structure of the multi-input model that had best accuracy of predicting gender on test data

As can be seen from the above figure, there are several layers in the model, that perform calculations only on sequences of mel-frequency cepstral coefficients and other isolated layers that perform calculation on i-vectors, before there is a dense layer that concatenates them. The model can be found from file model_90.py in the Github repository. Table 11 shows the parameters and the structure of the multi-input model.

Table 11 Structure of the multi-input model that predicts the gender of the speaker.

Feature	**Property**	**Value**
1. hidden layer (Input branch 1)	Type of layer	Conv2D
	Filters	128
	Size of convolutional window	3x20

Feature	**Property**	**Value**
	Activation function	relu
	border_mode	valid
2. hidden layer (Input branch 1)	Type of layer	Conv2D
	Filters	128
	Size of convolutional window	5x1
	Strides	(3,1)
	Activation function	relu
	border_mode	valid
3. hidden layer (Input branch 1)	Type of layer	Conv2D
	Filters	128
	Size of convolutional window	5x1
	Strides	(3,1)
	Activation function	relu
	border_mode	valid
4. hidden layer (Input branch 1)	Type of layer	Reshape
	Target shape	(-1, 128)
5. hidden layer (Input branch 1)	Type of layer	Bidirectional LSTM
	Size	128
	return_sequences	True
6. hidden layer (Input branch 1)	Type of layer	AttentionWithContext
7. hidden layer (Input branch 2)	Type of layer	Dense
	Size	4000
	Dropout	0.7
8. hidden layer (merges the inputs)	Type of layer	Dense
	Size	1500 neurons
	Dropout	0.5
8. hidden layer (merges the inputs)	Type of layer	Dense
	Size	300 neurons
	Dropout	0.3
Output layer	Type of layer	Dense

Feature	**Property**	**Value**
	Size	2 neurons
	Activation function	sigmoid
	Loss function	categorical_crossentropy
Optimizer		sgd
Learning rate		0.01
Epochs		30
Batch size		128

Appendix V – The Most Accurate Feedforward DNN for Age Group Identification

This appendix gives an overview of a feedforward DNN that predicts the age of the speaker that can be mapped to the according age group. Out of the feedforward DNN models built for only age group identification, this model had the best accuracy on both validation and test set. The model can be found from file dnn_2.py in the Github repository. The features with corresponding values of the model are brought out in the following Table 12, with some of the values taken from Keras documentation [37].

Table 12 Structure of the feedforward DNN that predicts age of the speaker.

Feature	**Property**	**Value**
1. hidden layer	Type of layer	Dense
	Size	2000 neurons
	Activation function	relu
	Dropout	0.7
2. hidden layer	Type of layer	Dense
	Size	1000 neurons
	Activation function	relu
	Dropout	0.4
3. hidden layer	Type of layer	Dense
	Size	100 neurons
	Activation function	sigmoid
Output layer	Type of layer	Dense
	Size	1 neuron
	Loss function	mse
Optimizer		sgd
Learning rate		0.01
Epochs		100
Batch size		32

Appendix VI – The Most Accurate RNN for Age Group Identification Trained on Feedforward DNN predictions

This appendix brings out the details about a RNN model, that predicts age group of a subject and has the highest accuracy out of the built age group RNN models on test data. The model was trained using predictions of a feedforward DNN as labels and can be found from file model_65.py in the Github repository. Table 13 shows the structure of the RNN, with some of the values taken from Keras documentation [37].

Table 13 Structure of the RNN that predicts age group of the speaker.

Feature	**Property**	**Value**
1. hidden layer	Type of layer	Conv2D
	Filters	128
	Size of convolutional window	(3,1)
	Activation function	relu
	kernel_regularizer	l2(0.0001)
	border_mode	valid
2. hidden layer	Type of layer	Conv2D
	Filters	128
	Size of convolutional window	5x1
	Strides	(3,1)
	Activation function	relu
	kernel_regularizer	l2(0.0001)
	border_mode	valid
3. hidden layer	Type of layer	Conv2D
	Filters	128
	Size of convolutional window	5x1
	Strides	(3,1)
	Activation function	relu
	kernel_regularizer	l2(0.0001)
	border_mode	valid
4. hidden layer	Type of layer	Conv2D
	Filters	128

Feature	**Property**	**Value**
	Size of convolutional window	5x1
	Strides	(3,1)
	Activation function	relu
	kernel_regularizer	l2(0.0001)
	border_mode	valid
5. hidden layer	Type of layer	Conv2D
	Filters	128
	Size of convolutional window	5x1
	Strides	(3,1)
	Activation function	relu
	kernel_regularizer	l2(0.0001)
	border_mode	valid
6. hidden layer	Type of layer	Reshape
	Target shape	(-1, 128)
7. hidden layer	Type of layer	Bidirectional LSTM
	Size	128
	return_sequences	True
8. hidden layer	Type of layer	AttentionWithContext
9. hidden layer	Type of layer	Dense
	Size	100
	Activation function	relu
	Dropout	0.1
Output layer	Type of layer	Dense
	Size	3 neurons
	Activation function	softmax
	Loss function	categorical_crossentropy
Optimizer		adam
Learning rate		0.001
Epochs		25
Batch size		128

Appendix VII – The Most Accurate RNN for Age Group Identification Trained on Real Labels

This appendix brings out the details about a RNN model, that predicts age group of a subject and has the highest accuracy on test data out of the built age group RNN models that were trained on real labels. The model can be found from file model_31.py in the Github repository. Table 14 shows the structure of the RNN, with some of the values taken from Keras documentation [37].

Table 14 Structure of the RNN that predicts age group of the speaker.

Feature	Property	Value
1. hidden layer	Type of layer	Conv2D
	Filters	128
	Size of convolutional window	(3,1)
	Activation function	relu
	kernel_regularizer	l2(0.0001)
	border_mode	valid
2. hidden layer	Type of layer	Conv2D
	Filters	128
	Size of convolutional window	5x1
	Strides	(3,1)
	Activation function	relu
	kernel_regularizer	l2(0.0001)
	border_mode	valid
3. hidden layer	Type of layer	Conv2D
	Filters	128
	Size of convolutional window	5x1
	Strides	(3,1)
	Activation function	relu
	kernel_regularizer	l2(0.0001)
	border_mode	valid
4. hidden layer	Type of layer	Conv2D
	Filters	128

Feature	**Property**	**Value**
	Size of convolutional window	5x1
	Strides	(3,1)
	Activation function	relu
	kernel_regularizer	l2(0.0001)
	border_mode	valid
5. hidden layer	Type of layer	Conv2D
	Filters	128
	Size of convolutional window	5x1
	Strides	(3,1)
	Activation function	relu
	kernel_regularizer	l2(0.0001)
	border_mode	valid
6. hidden layer	Type of layer	Reshape
	Target shape	(-1, 128)
7. hidden layer	Type of layer	Bidirectional LSTM
	Size	128
	return_sequences	True
8. hidden layer	Type of layer	AttentionWithContext
Output layer	Type of layer	Dense
	Size	3 neurons
	Activation function	softmax
	Loss function	categorical_crossentropy
Optimizer		adam
Learning rate		0.001
Epochs		15
Batch size		128
Label smoothing	smooth_factor	0.2

Appendix VIII – The Most Accurate Multi-Input Model for Age Group Identification

In this appendix, the details about the model that had the highest accuracy of identifying the age group on test data are brought out. Figure 49 illustrates the structure of this multi-input model.

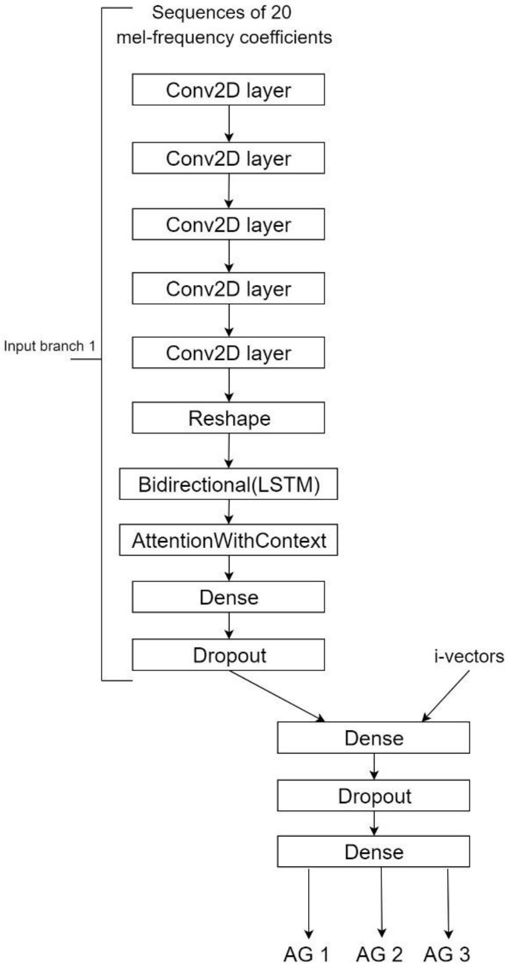


Figure 49 Structure of the multi-input model that had best accuracy of predicting age group on test data.

As can be seen from the above figure, there are several layers in the model, that perform calculations only on sequences of mel-frequency cepstral coefficients, before the corresponding i-vector is fed into the model. The model can be found from file model_63.py in the Github repository. Table 15 shows the parameters and the structure of the multi-input model.

Table 15 Structure of the multi-input model that predicts the age group of the speaker.

Feature	**Property**	**Value**
1. hidden layer (Input branch 1)	Type of layer	Conv2D
	Filters	128
	Size of convolutional window	3x20
	Activation function	relu
	border_mode	valid
	kernel_regularizer	l2(0.0001)
2. hidden layer (Input branch 1)	Type of layer	Conv2D
	Filters	128
	Size of convolutional window	5x1
	Strides	(3,1)
	Activation function	relu
	border_mode	valid
	kernel_regularizer	l2(0.0001)
3. hidden layer (Input branch 1)	Type of layer	Conv2D
	Filters	128
	Size of convolutional window	5x1
	Strides	(3,1)
	Activation function	relu
	border_mode	valid
	kernel_regularizer	l2(0.0001)
4. hidden layer (Input branch 1)	Type of layer	Conv2D
	Filters	128
	Size of convolutional window	5x1
	Strides	(3,1)
	Activation function	relu
	border_mode	valid
	kernel_regularizer	l2(0.0001)
5. hidden layer (Input branch 1)	Type of layer	Conv2D
	Filters	128
	Size of convolutional window	5x1
	Strides	(3,1)
	Activation function	relu

Feature	Property	Value
	border_mode	valid
	kernel_regularizer	l2(0.0001)
6. hidden layer Reshape layer (Input branch 1)	Type of layer	Reshape
	Target shape	(-1, 128)
7. hidden layer (Input branch 1)	Type of layer	Bidirectional LSTM
	Size	128
	return_sequences	True
8. hidden layer (Input branch 1)	Type of layer	AttentionWithContext
9. hidden layer (Input branch 1)	Type of layer	Dense
	Size	100 neurons
	Dropout	0.1
10. hidden layer (merges the inputs)	Type of layer	Dense
	Size	100 neurons
	Dropout	0.1
Output layer	Type of layer	Dense
	Size	3 neurons
	Activation function	softmax
	Loss function	categorical_crossentropy
Optimizer		adam
Learning rate		0.001
Epochs		40
Batch size		128

Appendix IX – The Most Accurate Multi-Output Feedforward DNN for Age Group Identification

This appendix brings out the structure of a multi-output neural network that predicts the gender and age group of a given speaker. The model can be found from file dnn_7.py in the Github repository. Table 16 shows the structure of the multi-output model, with some of the values taken from Keras documentation [37].

Table 16 Structure of the feedforward multi-output DNN that predicts age and gender of the speaker.

Feature	**Property**	**Value**
1. hidden layer	Type of layer	Dense
	Size	3000 neurons
	Activation function	relu
	Dropout	0.6
2. hidden layer	Type of layer	Dense
	Size	1500 neurons
	Activation function	relu
	Dropout	0.4
Gender output layer (follows second hidden layer)	Type of layer	Dense
	Size	2 neurons
	Activation function	softmax
	Loss function	categorical_crossentropy
3. hidden layer	Type of layer	Dense
	Size	100 neurons
	Activation function	sigmoid
Age output layer (follows third hidden layer)	Type of layer	Dense
	Size	1 neuron
	Activation function	sigmoid
	Loss function	mse
Optimizer		sgd

Feature	**Property**	**Value**
Learning rate		0.01
Epochs		50
Batch size		32

Appendix X – The Most Accurate Multi-Output RNN for Age Group Identification

This appendix brings out the details about a multi-output RNN model, that predicts the age and gender of a subject and has the highest accuracy on test data out of the built multi-output RNN models for both of the outputs. The model can be found from file model_94.py in the Github repository Table 17 shows the structure of the RNN, with some of the values taken from Keras documentation [37].

Table 17 Structure of the multi-output RNN that predicts age and gender of the speaker.

Feature	Property	Value
1. layer	Type of layer	Masking
1. hidden layer	Type of layer	Bidirectional LSTM
	Size	128
	return_sequences	True
2. hidden layer	Type of layer	Bidirectional LSTM
	Size	128
	return_sequences	True
3. hidden layer	Type of layer	AttentionWithContext
Gender output layer (follows third hidden layer)	Type of layer	Dense
	Size	2 neurons
	Activation function	softmax
	Loss function	categorical_crossentropy
4. hidden layer	Type of layer	Dense
	Size	100 neurons
	Activation function	sigmoid
Age output layer (follows fourth hidden layer)	Type of layer	Dense
	Size	1 neuron
	Activation function	sigmoid
	Loss function	mse
Optimizer		sgd
Learning rate		0.01
Epochs		100

Feature	**Property**	**Value**
Batch size		256

Appendix XI – The Most Accurate RNN for Males' Age Group Identification

In this appendix, the details about the most accurate model for identifying males' age group is brought out. The model can be found from file model_43.py in the Github repository. Table 18 shows the structure of the RNN, with some of the values taken from Keras documentation [37].

Table 18 Structure of the most accurate RNN for males' age group identification.

Feature	**Property**	**Value**
1. hidden layer	Type of layer	Conv2D
	Filters	128
	Size of convolutional window	3x20
	Activation function	relu
	border_mode	valid
	kernel_regularizer	l2(0.0001)
2. hidden layer	Type of layer	Conv2D
	Filters	128
	Size of convolutional window	5x1
	Strides	(3,1)
	Activation function	relu
	border_mode	valid
	kernel_regularizer	l2(0.0001)
3. hidden layer	Type of layer	Conv2D
	Filters	128
	Size of convolutional window	5x1
	Strides	(3,1)
	Activation function	relu
	border_mode	valid
	kernel_regularizer	l2(0.0001)
4. hidden layer	Type of layer	Conv2D
	Filters	128
	Size of convolutional window	5x1

Feature	**Property**	**Value**
	Strides	(3,1)
	Activation function	relu
	border_mode	valid
	kernel_regularizer	l2(0.0001)
5. hidden layer	Type of layer	Conv2D
	Filters	128
	Size of convolutional window	5x1
	Strides	(3,1)
	Activation function	relu
	border_mode	valid
	kernel_regularizer	l2(0.0001)
6. hidden layer Reshape layer	Type of layer	Reshape
	Target shape	(-1, 128)
7. hidden layer	Type of layer	Bidirectional LSTM
	Size	128
	return_sequences	True
8. hidden layer	Type of layer	AttentionWithContext
Output layer	Type of layer	Dense
	Size	3 neurons
	Activation function	softmax
	Loss function	categorical_crossentropy
Optimizer		sgd
Learning rate		0.01
Epochs		50
Batch size		64

Appendix XII – Ensemble Model for Males' Age Group Identification

The models that were used to boost the accuracy on separated males' test data can be found in the following files from the provided Github repository brought out in Appendix I: model_43_males.py, dnn_2_males.py, model_57_males.py and model_65_males.py. The corresponding weights for the predictions of each of the models are 0.3, 0.3, 0.3 and 0.2. All of the used models were trained on separated males data and the code for the built ensemble model can be found in merge_predictions.py.

Appendix XIII – Ensemble Model for Females' Age Group Identification

The models that were used to boost the accuracy on the separated females' test data can be found in the following files from the provided Github repository brought out in Appendix I: model_63_females_original.py, dnn_6_females_original.py, model_94_females_original.py, model_65_females_original.py. None of the used models were trained on separated females' dataset. The code for the ensemble model can be found in merge_predictions.py.

Appendix XIV – Ensemble Model for Age Group Identification

The models that were used to boost the age group identification accuracy on test data through averaging can be found in the following files from the provided Github repository brought out in Appendix 1: model_63.py, dnn_7.py, dnn_2.py and dnn_6.py. The corresponding weights for the predictions of each of the models are 0.5, 0.5, 0.3 and 0.1. The built ensemble model can be found in file combined_predictions_2.py.