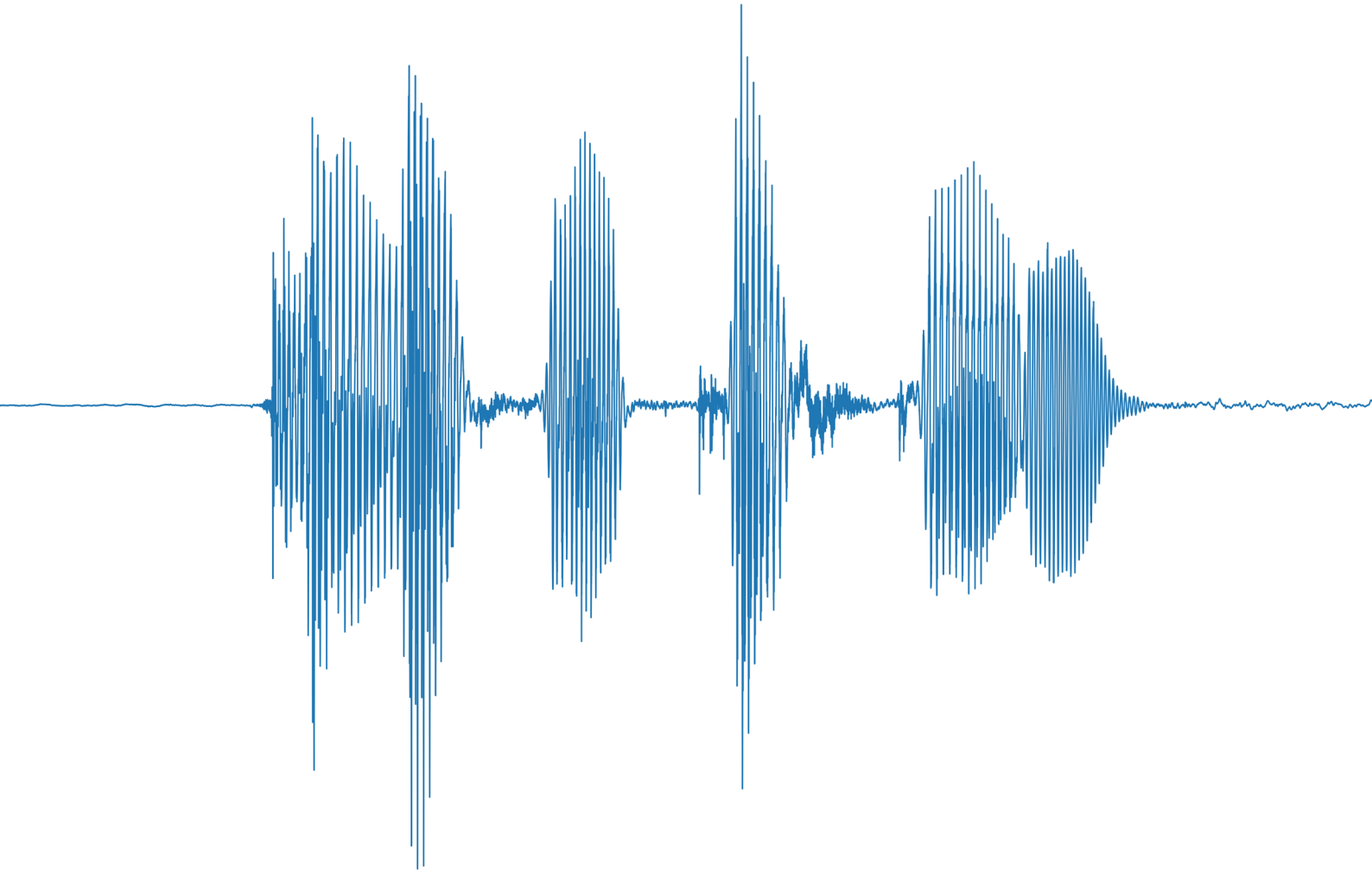
Department of People and Technology
AUTUMN 2021

SimoNN

Author
Halim **Rahman**

Teachers
Henning **Christiansen**
Mikkel **Pedersen**



Automatic speaker recognition using convolutional neural networks trained upon non-specific speeches

A mini-project for the course *Artificial Intelligence: Deep Learning*

22 November 2021

Mini-Project

Department of People and Technology

Roskilde University, Autumn 2021

Research Title

Automatic speaker recognition using convolutional neural networks trained upon non-specific speeches

Research Question

Using a bespoke dataset, how to train minimal neural network architectures to perform automatic speaker recognition?

Authored by

Abdul Halim **bin Abdul Rahman**
63870 (NIB 2017)
ahbar@ruc.dk

MSc. candidate in
Mathematical Computer Modelling
Dept. of Science and Environment

Taught by

Henning **Christiansen**
henning@ruc.dk

Mikkel **Pedersen**
mikped@ruc.dk

NOTICE & REMARKS

1. Figures (charts, diagrams, graphs and/or any other graphics composite) that appear in this report, are credited with their respective source; otherwise, they are originally plotted, compiled or recreated. Figures produced were based on the literature understanding.

2. Every facial image and voice signal that appears in this report is produced with explicit permission from its respective volunteer with all the rules and laws observed. None of the personal information of any of the volunteers for this project has been reproduced.

CITATION

The volunteers' permissions given were given only for this report and therefore, FIGURES CONTAINING FACIAL IMAGES AND/OR VOICE SIGNALS **ARE NOT PERMITTED** FOR REPRODUCTION.

To cite this report,

@MISC { rahman_2021,
TITLE = "Automatic speaker recognition using convolutional neural networks trained upon non-specific speeches",
AUTHOR = "Abdul Halim bin Abdul Rahman {Rahman et al.}",
INSTITUTION = "Roskilde University, Denmark",
HOWPUBLISHED = "Mini project",
YEAR = "2021",
NOTE = "Artificial Intelligence: Deep Learning (autumn 2021)",
}

FEEDBACK

Feedbacks are welcome! I did my best to be as concise and as precise as possible, informative and close to the point. However, there might be errors or wrongly explained or certain topics may be better rephrased to be more precise or comprehensible, please don't hesitate to contact me. It would benefit me for future reports as more works I am heading towards the implementation of neural networks with different experiments.

Please write to `halim@ebyx.net`

Abstract

Several works of literature documented their finding on automatic speech recognition and also automatic speaker recognition using Convolutional Neural Networks (CNN). The specific part was about using convolution 1-dimensional in CNN to train a model for automatic speaker recognition. This mini-project embarks first on collecting bespoke data from several volunteers and then trained these data using different data treatments and two different CNN architectures. Volunteers were not constrained in producing specific sounds and were asked to read/speak naturally. Four experiments took place with one showing a failure as it could not produce any classification. Three other experiments were deemed successful with all three managing to rake between 95 to 99 per cent accuracies. Experimenting with convolution 1-dimensional side by side with convolution 2-dimensional, it was found that convolution 2-dimensional yields better results.

Keywords: 1-dimensional convolution, 2-dimensional convolution, audio classification, automatic speaker recognition, automatic speaker verification, biometric identification, classification learning, convolutional neural networks, sound classification, speech synthesis, supervised learning, time-series signal.

TABLE OF CONTENTS

Abstract

Section 1 Background
1.1 Problem Formulation 1
1.2 Research Question 1
1.3 Introduction 1
1.3.1 Speaker Recognition 1
1.3.2 Speech Signals 3
1.3.3 Convolutional Neural Networks 6
1.4 Method 7

Section 2 Data Collection
2.0 Volunteer and Voice Collection 8
2.1 Data Cleaning and Management 9
2.2 Data Conversion 9
2.3 Data Samples 11

Section 3 Experiments
3.0 Python Libraries and Frameworks 12
3.1 Architectures 12
3.2 Experiment 1 - STFT (Spectrogram) using Conv1D 13
3.3 Experiment 2 - MFCC (Spectrogram) using Conv1D 14
3.4 Experiment 3 - MFCC (Spectrogram) using Conv2D 16
3.5 Experiment 4 - MFCC (Spectrogram Image) using Conv2D 18
3.6 Predictions 20

Section 4 Analysis, Results and Discussion
4.0 Analysing Experiment Results 21
4.1 Best Performance 22
4.2 Discussion 22
4.3 Conclusion 23
4.4 Future Works 23

Bibliography 24

Appendix 1 Training Results 26

SECTION 1: BACKGROUND

This section provides the background for this report and introduction into the problem domain, namely speaker recognition and 1-dimensional convolutional neural networks (Conv1D). It also entails the research question and the method used to solve the problem. This section is however not going in-depth on the physics and mathematical aspects other than a brief introduction to certain terms.

1.1 Problem Formulation

The purpose of this report is as a mini-project for a class assignment, *Artificial Intelligence: Deep Learning* (autumn semester 2021), in which students are permitted to explore certain domains using knowledge gained from the class. The author chooses this particular topic as an exploration into voice recognition and also to explore the specific domain of 1-dimensional and 2-dimensional convolutional neural networks. The main inspiration for this project is drawn from a report by *Casanova et al., Speech2Phone: A Multilingual and Text Independent Speaker Identification Model* [1].

1.2 Research Question

In gearing this report towards *Automatic speaker recognition using convolutional neural networks trained upon non-specific speeches*, a research question has been formulated as follows,

Using a bespoke dataset, how to train minimal neural network architectures to perform automatic speaker recognition?

1.3 Introduction

This section provides the background and introduction to the problem domains for this particular mini-project. It first introduces the speaker recognition problem, before moving towards the speech signal representation and then into the convolution part. Due to time and requirement constraints, the introduction only provides condensed information without the mechanics of mathematics or physics.

1.3.1 Speaker Recognition

Speaker recognition is defined as recognizing the identity of an individual based on the voice [2] or speech utterance [3]. *Bai et al.* [2] referred to the term, *automatic speaker recognition,* as the task being handled by machines as opposed to recognition by humans. *Liu et al.* [4] used the term *Automatic speaker verification (ASV)*[1] for voice comparison, personalization of voice-

[1] For the purpose of distinction, this report is using the term ***ASV*** for *automatic speaker recognition* tasks to differentiate from *automatic speech recognition (**ASR**)* tasks.

based services and devices. The task of automatic speaker recognition is different from *automatic speech recognition (ASR)*, as the latter is interested in converting given speech audio snipped into text in real-time [5]. Feature extraction from a voice can be done using machine learning [6], [7], [8]. Voice recognition is based on personal traits of the speaker which is produced by the vocal tract and larynx organs which are unique as well as one's personal accent [9] and can be extracted as a unique feature.

A biometric identification aims in recognizing a person based on physiological or behavioural characteristics [10] that includes face and voice[2]. Speaker recognition can be treated as a biometric identification as voice is characteristic data [10] that is unique for individuals. For face recognition, a set of facial images are fed into a neural network for classification. The input can be formatted to share similar dimensions, for example in terms of height, width and also the colour depth. The facial features can also be observed from different angles as in *Figure 1* and simple observations can be used to conclude as to who the person was.

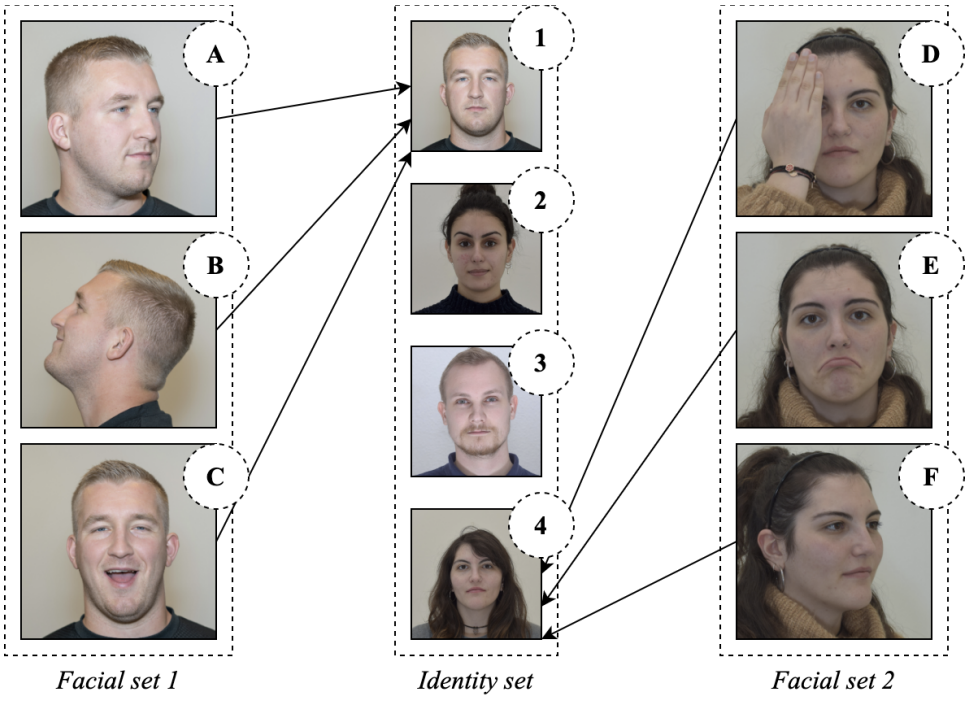


Figure 1: Three different facial images from different angles and also with different expressions. The identification is trivial as one can easily distinguish the facial features and conclude who the person was. For example, the "facial set 1" and "facial set 2" can be easily pointed towards the correct person listed in the "identity set". A, B and C are from two different angles with B having the most extreme angle. C and E are showing different emotions while D has the face partially covered.

[2] *H. Mandalapu et al.* [10] were interested in a system where both audio and visual biometrics recognition to be used for authentication that includes identification and verification.

However, audio as an input for a neural network uses a different dimension, that is time. Another problem with audio is that there are almost unlimited utterances and phonemes that one human can make and it is therefore impossible to identify a person based on one audio snippet alone. It has been mentioned that each person has a unique vocal tract and larynx and these produce different levels of utterances. Therefore, most of the projects [1], [11], [12], [13] and [14] are based on identifying certain unique *noises*. A second problem is that audio can vary in length. The time-domain source has to be converted to a different measurable domain as an input with a congruent dimension that can be accepted by a proposed neural network.

Different covariates influence the utterance of an individual, for example, languages, genders, ages, timbres, accents, and background noises [3]. A proper identification system should be trained using samples with different demographic characteristics, languages, genders and age-group [3].

1.3.2 Speech Signals

Sound is the changes of pressure waves created through vibrations [15]. A sound wave is described through the number of waves in a second and its size. **Measuring the number of waves passing in a second[3] produces frequency** [15]. **Observing the amplitude of a sound wave allows the measurement of loudness intensity** [15]. The speech signal is a sound wave that has variation in the number of air pressures that varies over time. The information is captured by collecting the samples of the air pressure over time. The rate of data sampling for frequency is measured in Hertz (Hz), for example, 48,000 Hz means there were 48,000 samples collected per second [15]. The signal is captured as a waveform and as the signal's frequency varies over time, it is known as a *non-periodic signal*. **An audio signal is a time-series data which in this context is a speech signal made of several single-frequency sound waves, recorded over a time span**. During a sampling for signals over a spanned time, the changes in air pressures' (amplitudes) are captured [16]. The captured signal can be decomposed using Fourier transform into separate frequencies and also into the frequency's amplitude, resulting in the signal being converted from the time domain to the frequency domain. The result of this is called a *spectrum*. An algorithm known as the Fast Fourier Transform (FFT) is used to compute the Fourier transform [17]. Using FFT, the frequency content of a signal can be analysed and by computing several spectrums by performing FFT on several windowed segments of the signal as is known as short-time Fourier transform [17]. By calculating the FFT on overlapping windowed segments of the signal, it produces a spectrogram [17].

[3] This report chose to be explicit about the unit time, that is second.

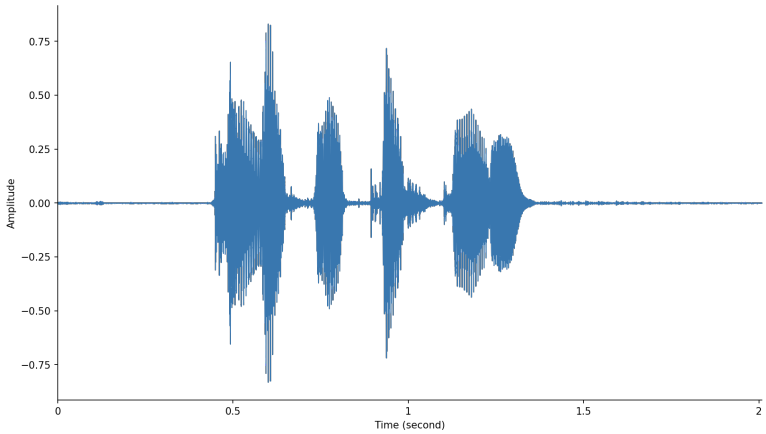


Figure 2: Oscillograph is a digital representation of an audio signal visualized as a time-domain that is by plotting amplitude over time. The amplitude shows the loudness of sound waves that is changing with time. The amplitude represents the air particles that are oscillating because of the pressure change in the atmosphere due to sound.

From the oscillogram in *Figure 2*, the amplitudes are not very useful as it only provides a glimpse of the loudness of a recorded sound as it varies over time at different frequencies. More information can be obtained if it is transformed into the frequency domain. It is a representation of a signal that provides information on the different frequencies that are present in the signal [18]. An audio signal is composed of multiple single-frequency sound waves that travel together and are captured through the resultant amplitude of those multiple waves [18]. A commonly used algorithm is the Fourier transform as it can decompose a signal into its constituent frequencies along with the magnitude of each frequency as shown in *Figure 3*.

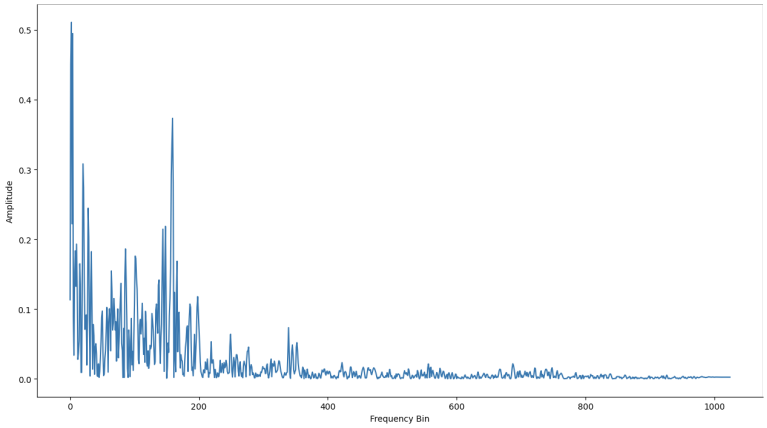


Figure 3: FFT is a process of binning the amplitude occurrences into the frequency's bin.

At an arbitrary spot of point in time, a specific sound is made of different frequency combinations and each frequency has different intensity. As seen in an oscillograph as in *Figure 2* that only mimics the amplitude, a different visualization is needed to represent the variation of frequency intensity over a given period. This visualization is known as a spectrogram as in *Figure 4* and shows the acoustic features from audio [19]. It is produced from computation from Fourier transform by converting the amplitude values into decibel scaled values.

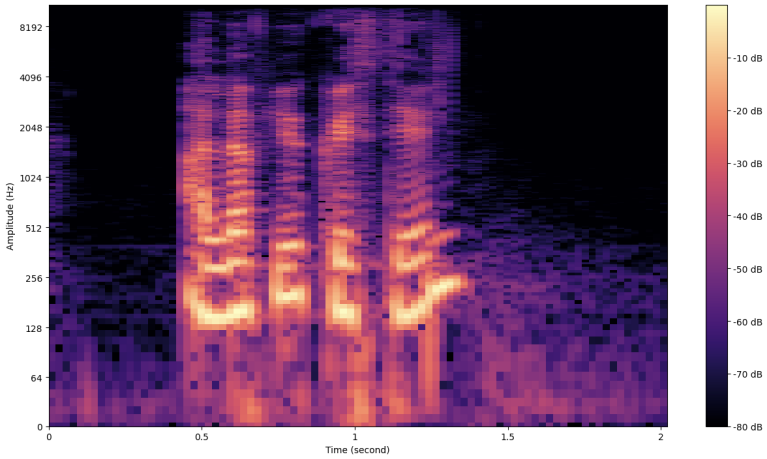


Figure 4: Spectrograms is a visual representation of an audio signal that plots the intensity of frequencies against time. Yellow is for very low intensity and dark purple is for very high intensity. Simple observation shows that certain frequency spectrums do have near-zero intensity and vice versa.

Voice can be collected in the form of audio samples that are a one-dimensional time-series signal. Audio samples tend to be transformed into two-dimensional time-frequency. However, the axes, time-vs-frequency are not homogeneous to the axes for images [20] that represented the horizontal *(width)* and vertical *(height)*. Audio samples require certain processing in order to extract certain feature representations. One of the well-used processing is *Mel-Frequency Cepstral Coefficients (MFCCs)* [20] to extract dominant acoustic feature representations as in *Figure 5*. According to *Davis et al.* [21], the Mel filter bank was inspired by the human auditory system.

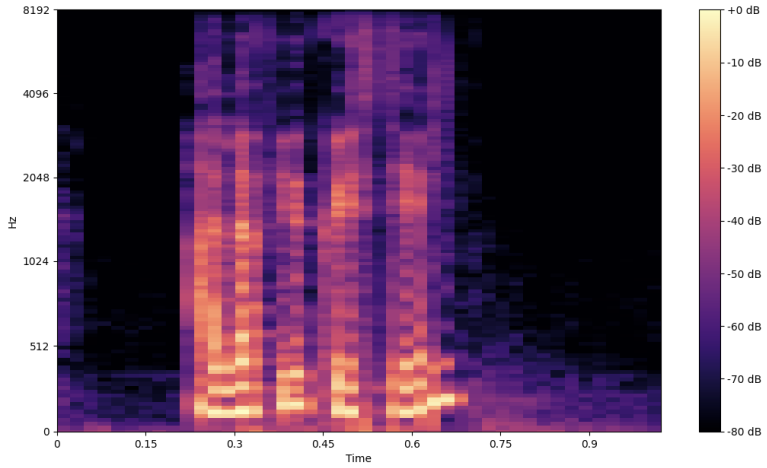


Figure 5: Mel-scale spectrogram is produced when frequencies are converted to mel scale. The audio signal for this is similar to the source for Figure 4 and when comparing the two, it can be seen that the mel-scale spectrogram is slightly cleaner and less noisy especially between 0 and 512 Hz.

1.3.3 Convolutional Neural Networks (CNN)

Pattern recognition is a process of identifying trends from a cluster of patterns. A pattern exhibits certain regularity. Patterns help the identification or recognition of objects from varying distances or angles. Using machine learning, pattern recognition can be automated to be an efficient solution to solve problems in real-time [22]. Machine learning with decent number of samples has exhibited the ability to identify minute details and use the gained knowledge to recognize or classify new unseen data.

Time series signals are different from two-dimensional images. For example, facial images can be instantaneously captured from a source and then analysed without much alteration. At most, the captured input is normally cropped to the facial region. Audio signals which are fundamentally different from facial images have to be studied sequentially in chronological order [20]. As discussed in *section 1.3.2*, a voice signal can be transformed into a spectrogram that generates unique features in a 2-dimensional matrix for frequency magnitudes along the time length for a given signal. The 2D matrix can be visualized as a figure and therefore can be used as an image classification problem. The image can represent the direction of speech from left to right on the time axis. Therefore, the problem can be solved using 2-dimensional convolution (Conv2D). A simplified understanding of convolution 2D is as in *Figure 6*.

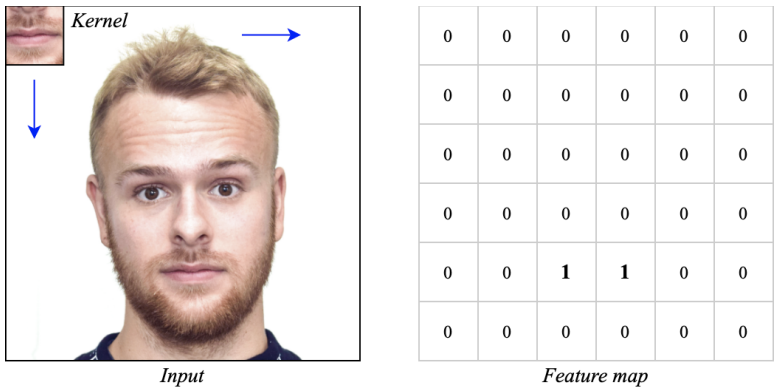


Figure 6[4]: Without being technically precise, a Conv2D refers to the process of sliding an m×n kernel over an input (in this example, a face) from left to right and top to bottom until the kernel traverses the whole input. If the kernel found something similar to its content, it marked the location in the feature map. Each kernel only produces one feature map, which can then be passed to another set of kernels to traverse.

As voice signal is a time-length problem, it has been suggested [23] that 1-dimensional convolution (Conv1D) is the most likely candidate to yield the best result after training. A simplified understanding of convolution 1D is as in *Figure 7*.

[4] This figure originally appeared in the author's previous report, *Rahman et al., "Probing the Mathematical Model for Multinomial Logistic Regression," Bachelor Thesis, Roskilde University, 2021 [22].*

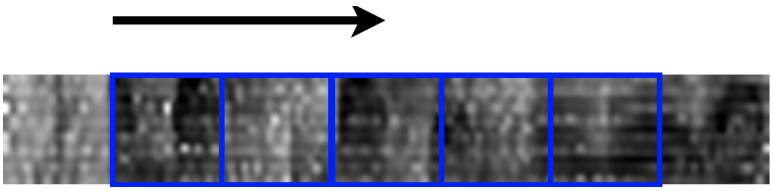


Figure 7[5]: Without being technically precise, a Conv1D refers to the process of sliding an m×1 kernel (in this example, 5×1) over an input (in this example, feature-vector for an arbitrary time, t) only in one direction as shown by the arrow. If the kernel found something similar to its content, it marked the location in the feature map similar to the example in Figure 6, except that the feature map is a vector. Each kernel only produces one feature map, which can then be passed to another set of kernels to traverse.

1.4 Method

This report is based on data collected from several volunteers. The data is in the form of recorded voices. Each volunteer was asked for explicit permission and explained regarding his/her rights on the *General Data Protection Regulation (GDPR)*[6]. Further discussion on this is in *section 2*. The dataset from the collection is used for neural network training using certain architectures which are explained in *section 3*.

The experiments are based on supervised learning therefore it is predicting a single global class label, which is a sequence classification [20]. The predefined set of possible classes are going to be only the volunteers identification number. The experiments were using CNN Conv-1D and CNN Conv-2D. The reason for this is to compare which network has a better performance. Another reason is that, as mentioned in *section 1.3.3*, since a spectrogram can be represented as a matrix and also visualized as a figure, the experiments can yield additional clues on the performance.

[5] This figure was simplified from an understanding from [24].
[6] Further information regarding the law is available at *https://gdpr-info.eu/*

SECTION 2: DATA COLLECTION

This section discusses the method for collecting data to be used for the experiment. It also entails the method used for organizing the data and the conversion methods on data treatment before using them for the experiments.

2.0 Volunteer and Voice Collection

As mentioned in *section 1.4*, volunteers were asked to sign a permission form to explicitly permit the use of their voice recordings for this report. The recording took place at various locations and at various times of the day. Volunteers were asked to speak over a microphone and the speech was recorded. Volunteers were asked to speak in their mother tongue. If their mother tongue is not English, a separate session is made to repeat the same process. Each speech was recorded at 48,000 Hz at 768kbps and saved as WAV PCM (.wav) in uncompressed/lossless 16-bit[7] format.

One session has two recordings. The **first recording** is made for around 2 minutes and 5 seconds. As it was difficult for volunteers to maintain talking for 2 minutes without stopping, they were asked to read an article from the internet or a book. The reason for the additional 5 seconds was to allow for 2.5 seconds clipping at the beginning and end of the recording to avoid silence. A **second recording** was made for 25 seconds for a *"freestyle"* speech to collect samples for a normal speech tempo and rhythm. The setup for the data collection environment can be seen in *Figure 8*. The main equipment is a RØDE *VideoMicro* that was put on a stand facing a volunteer at a distance of around 30 cm from the mouth. A windjammer is also used to reduce hissing when talking.

Figure 8: A volunteer was asked to sit down and read from his mobile phone. His voice is recorded using a RØDE VideoMicro microphone which is connected to a mobile phone.

[7] A 16-bit depth audio means that the sound pressure values are mapped to integer range between -2^{15} and $(2^{15})-1$.

2.1 Data Cleaning and Management

Each recording is grouped by volunteer and each session is uniquely identified by the date of recording. For volunteers who can speak more than one language, the session is annotated using ISO 639-1 code[8]. For example, for a volunteer who speaks both English and Danish, each session is annotated accordingly. The data organisation is summarized in *Table 1*.

Level 1	Volunteer 1
Level 2	2021-10-15 (EN)
Level 3	recording_1.wav
Level 3	recording_2.wav
Level 2	2021-10-15 (DA)
Level 3	recording_1.wav
Level 3	recording_2.wav

Table 1: Dataset management. ***Level 1*** *describes the unique label for the training.* ***Level 2*** *describes the unique session taken, denoted with the date a session took place as well as the spoken language.* ***Level 3*** *describes the files taken during a recording session and renamed accordingly. Recording_1.wav is for reading and recording_2.wav is for freestyle.*

2.2 Data Conversion

The first step is to remove the silence from a recording. This report treats silence as noise because it contains less or no useful information that can be used for training purposes. Then followed by the second step, to remove the starting and ending of a recording so that it can be split equally. The third step is to split the original recording into a mini-clip with equal length. A recording is split into a 5-second[9] sequence clip and saved in the format,

Filename: **volunteer-id_session-date_language_recording-type_split-sequence**.wav

[8] ISO 639 is a standardized nomenclature used to classify languages. Each language is assigned a two-letter (639-1). For example; DA for Danish, EL for Greek, EN for English, etc. More information is available at *https://www.loc.gov/standards/iso639-2/php/English_list.php*

[9] The choice for 5-second splitting is arbitrary. The author assumes that 5-second duration has enough information for a person to speak and deliver a short useful sentence. For example, an arbitrary test for a sentence, *"I like artificial intelligence"* only has 2.2-second in length.

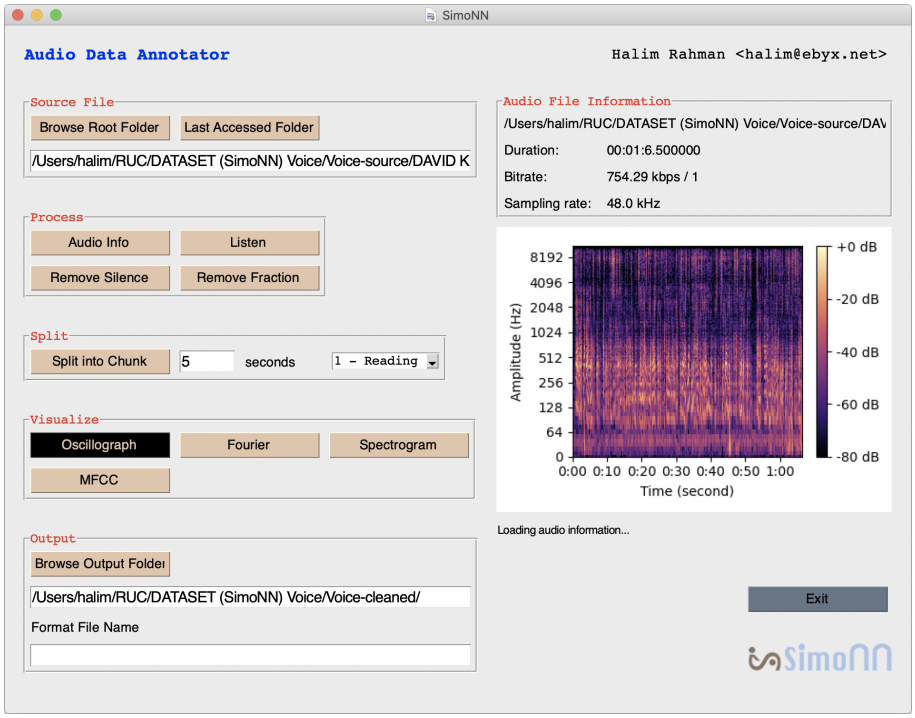


Figure 9: A screenshot from Audio Data Annotator which was written by the author specifically to expedite the process of audio cleaning, visualization and most importantly the audio splitting.

For the experiments, each data is converted into four different formats:

a. Short-time Fourier transform (STFT) in NumPy format,
b. Mel Frequency Cepstral Coefficient (MFCC) in NumPy format,
c. Mel Frequency Cepstral Coefficient (MFCC) converted to an image.

The images are saved as a PNG file to avoid having an artefact that is frequently observed when saved as a JPEG file [25]. Instead of saving the file in a 3-channel format, images are saved as a single channel file.

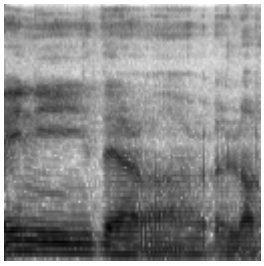


Figure 10: An audio that has been passed through a mel-spectrogram filter and then converted into a PNG file. The choice for a PNG file is to retain the compression and avoid image artefacts.

2.3 Data Samples

Voice recording samples were collected from several volunteers and only six of them yielded close or more than 250 sub-samples after their recordings were cleaned and split. For the experiment, the data for *recording type 1 (reading)* and *recording type 2 (freestyle)* were mixed and randomized. Before being used in the experiment, the data were split into two parts, 80 per cent for training and 20 per cent for validation. The split was done equally for each volunteer.

Index	Volunteer ID	Recording Type 1	Recording Type 2	Total
0	91	203	69 / 25.36%	272
1	200	191	60 / 23.90%	251
2	27	238	38 / 13.77%	276
3	9	215	33 / 13.31%	248
4	84	203	50 / 19.76%	253
5	72	220	28 / 11.29%	248
			Total Samples	**1,548**

Table 2: This table summarizes the number of total recordings collected from six volunteers and the distribution of recording type 1 (reading) and recording type 2 (freestyle) for each volunteer.

SECTION 3: EXPERIMENTS

This section entails the experiment formulated to answer the research question which was posed to find a minimal neural network architecture that may work well for the defined problem, that is automatic speaker recognition. The analysis and discussion regarding the results obtained are in section 4.

3.0 Python Libraries and Frameworks

To keep in line with the requirements for the course, the experiments were conducted using Keras and Tensorflow. For the mathematical aspect, the library is NumPy and for the audio, the library is Librosa. The project has been designed to use as fewer libraries as possible.

3.1 Architectures

The design for the architecture was inspired by VGG-16 structures. It similarly follows the architecture but Conv2D layers were replaced with Conv1D. However, due to the dataset's small size and without the use of transfer learning, the training from scratch did not produce useful results. Therefore, trial and error were done by altering the layers to find a smaller architecture that produces better results. In the final architecture, only 2 blocks from the convolutional segment were kept. In the classifier segment, one dense layer was removed and the other remaining layer was resized from 4,096 to 256. Also based on trial-and-error, two architectures as in *Figure 11* were finalized for experiments. Both architectures share similarities in terms of layers depth, dropout and classifier segment. The only difference is that architecture 1 uses convolution 1D while architecture 2 uses convolution 2D.

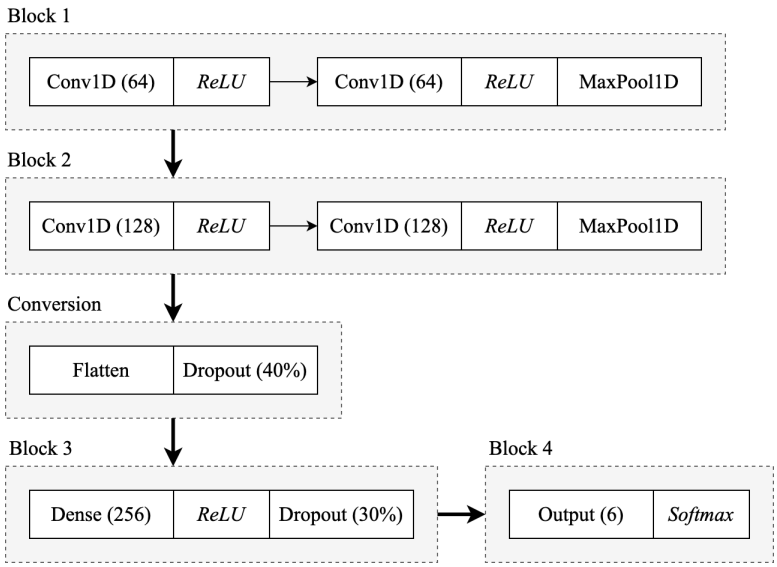


Figure 11: Architecture 1 showing the building block of the CNN architecture used for experiment 1, experiment 2 and experiment 3. For experiment 4, the Conv1D layers in Block 1 and Block 2 were substituted with Conv2D.

The hyperparameters are set as in *Table 3* below. The number of epochs was chosen based on the previous trial-and-error before finalizing the suitable architecture for the experiments. It was chosen to use Adam (Adaptive Momentum) as the optimizer as it is less volatile when compared to RMSProp considering that the number of samples was relatively small. The learning rate is an arbitrary value, chosen to be as minimal as possible.

Epoch	20
Optimizer	Adam
Learning rate	0.0007

Table 3: All of the experiments in this report are using these hyperparameter values for the training process.

3.2 Experiment 1 - STFT (Spectrogram) using Conv1D

The experiment was done using STFT spectrograms in a NumPy format. The result for each epoch is in Appendix 1 and plotted as in *Figure 11*.

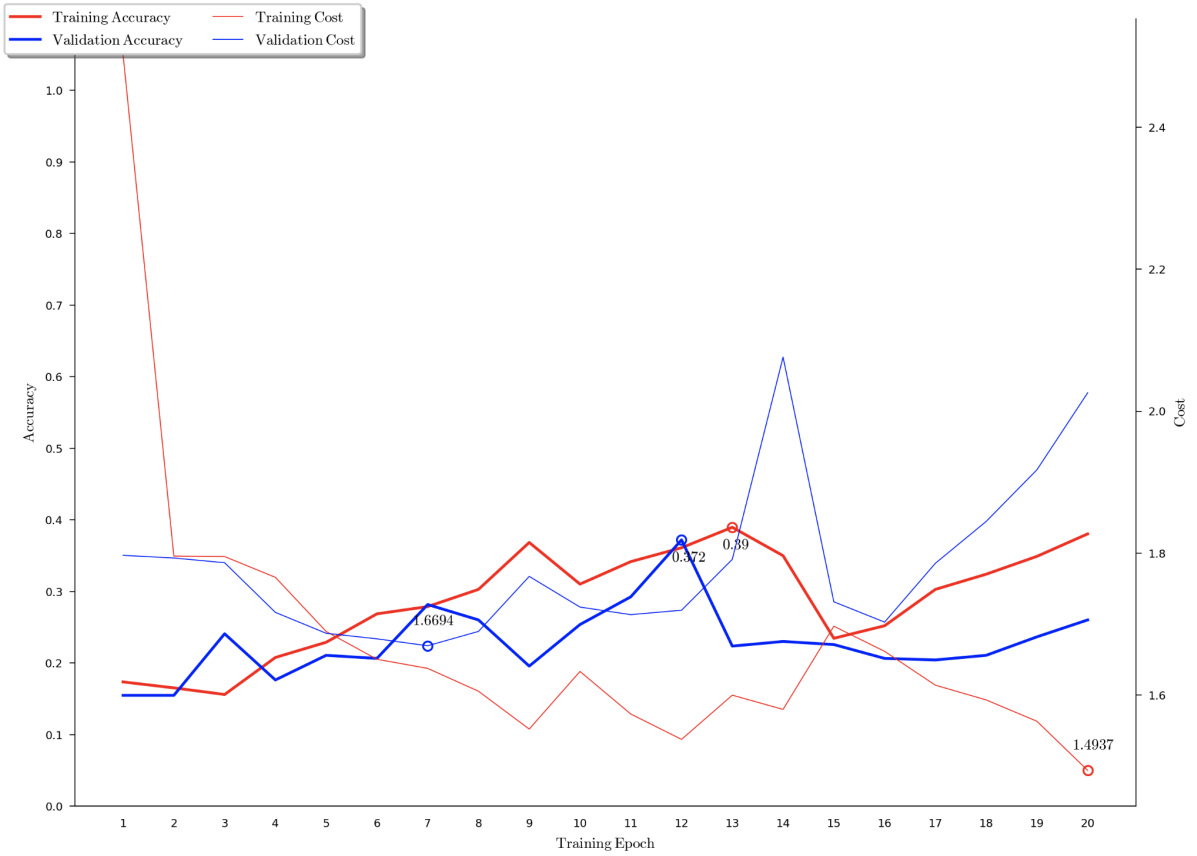


Figure 12: The history for training and the validation from experiment 1 are plotted showing the accuracy on the y-axis on the left and the cost on the y-axis on the right. Neither accuracy for training nor validation passed 50 per cent.

3.3 Experiment 2 - MFCC (Spectrogram) using Conv1D

When audio is converted into an MFCC format, the output given is in the format MFCC feature dimension on the row (*y*-axis) and the time dimension on the column (*x*-axis) [26]. As the convolution process must take place on the time dimension (that is iteration of the convolution process is run over on rows), the format must be flipped so that the time dimension is on the row (*y*-axis). Therefore, each timestep is now a row that holds 1 feature vector.

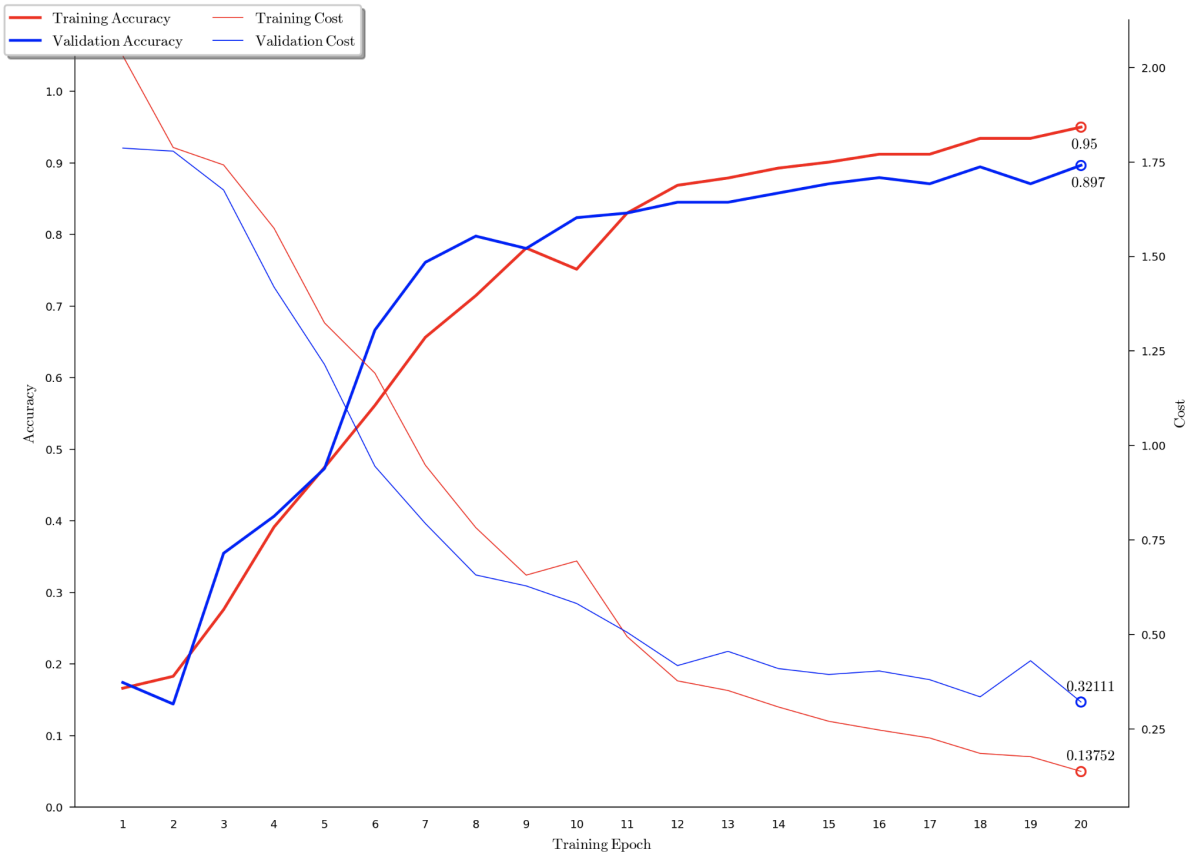


Figure 13: The history for training and the validation from the experiment are plotted showing the accuracy on the y-axis on the left and the cost on the y-axis on the right.

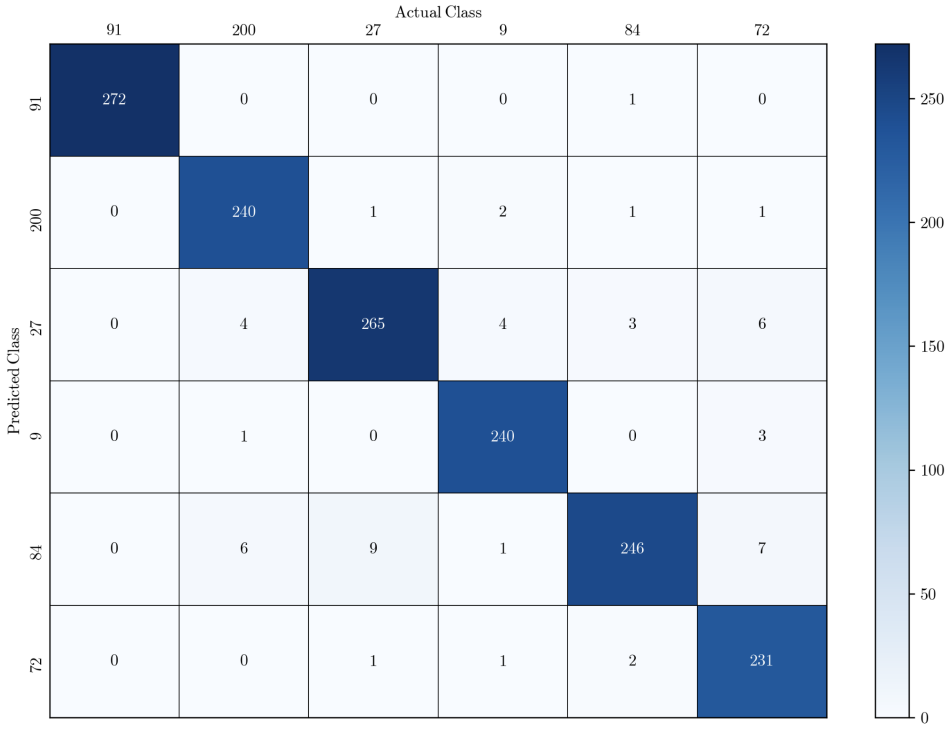


Figure 14: A confusion matrix was produced using the trained model from experiment 2 and was used to revalidate the whole dataset.

```
Classification Report:
                      Precision   Recall   F2-score
                      =========   ======   ========
Volunteer 91             1.0000   0.9963     0.9971
Volunteer 200            0.9562   0.9796     0.9748
Volunteer 27             0.9601   0.9397     0.9437
Volunteer 9              0.9677   0.9836     0.9804
Volunteer 84             0.9723   0.9145     0.9255
Volunteer 72             0.9315   0.9830     0.9722

Accuracy (Overall)                           0.9884
Macro-averaged           0.9646   0.9661     0.9656
Weighted-averaged        0.9651   0.9660     0.9656
```

Table 4: The metrics for precision, recall and F2-score for the prediction for the whole dataset using the trained model from experiment 2.

3.4 Experiment 3 - MFCC (Spectrogram) using Conv2D

Using similar data as in experiment 2, the flattened data were reshaped into *(128, 128, 1)* matrix where the columns represent features and rows represent time and as the data is not an actual photograph, it has no depth.

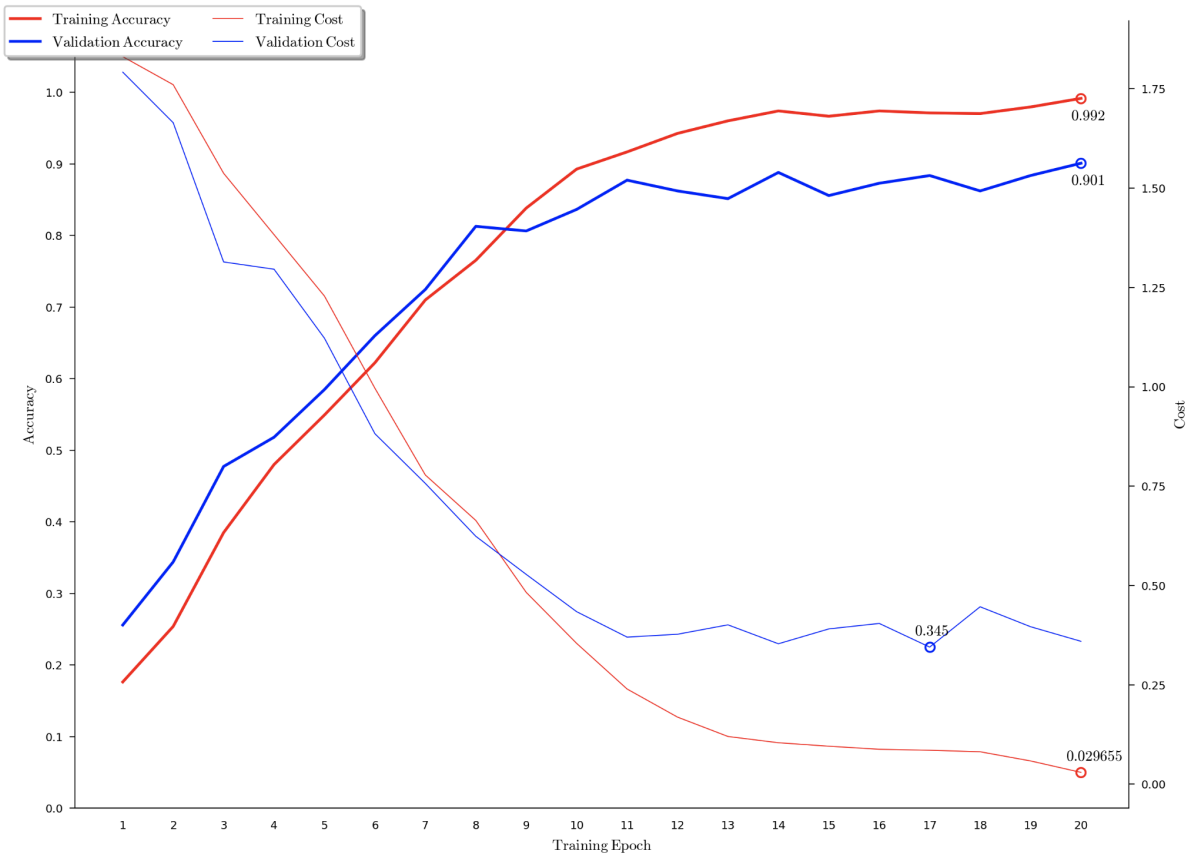


Figure 15: The history for training and the validation from the experiment are plotted showing the accuracy on the y-axis on the left and the cost on the y-axis on the right. A quick observation here tells that the training overfit.

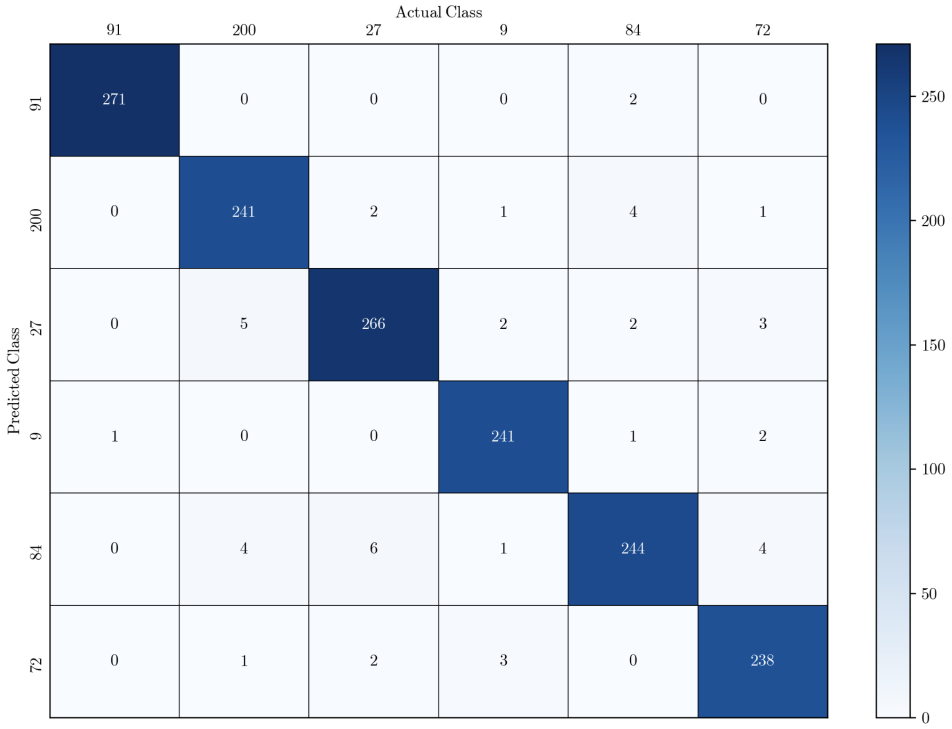


Figure 16: A confusion matrix was produced using the trained model from experiment 3 and was used to revalidate the whole dataset.

```
Classification Report:
                      Precision   Recall   F2-score
                      =========   ======   ========
Volunteer 91             0.9963   0.9927     0.9934
Volunteer 200            0.9602   0.9679     0.9663
Volunteer 27             0.9638   0.9568     0.9582
Volunteer 9              0.9718   0.9837     0.9813
Volunteer 84             0.9644   0.9421     0.9465
Volunteer 72             0.9597   0.9754     0.9722

Accuracy (Overall)                           0.9899
Macro-averaged           0.9694   0.9698     0.9696
Weighted-averaged        0.9696   0.9698     0.9697
```

Table 5: The metrics for precision, recall and F2-score for the prediction for the whole dataset using the trained model from experiment 3.

3.5 Experiment 4 - MFCC (Spectrogram Image) using Conv2D

The dataset was converted similarly in experiment 3. But the final data was scaled between *[0, 254]* and saved as a single channel PNG image. Each data has 16,384 data points. The result from the experiment is recorded in *Table A-4* in the appendix and plotted as in *Figure 17*. Before being passed for the training, the data has been rescaled to *[0, 1]*.

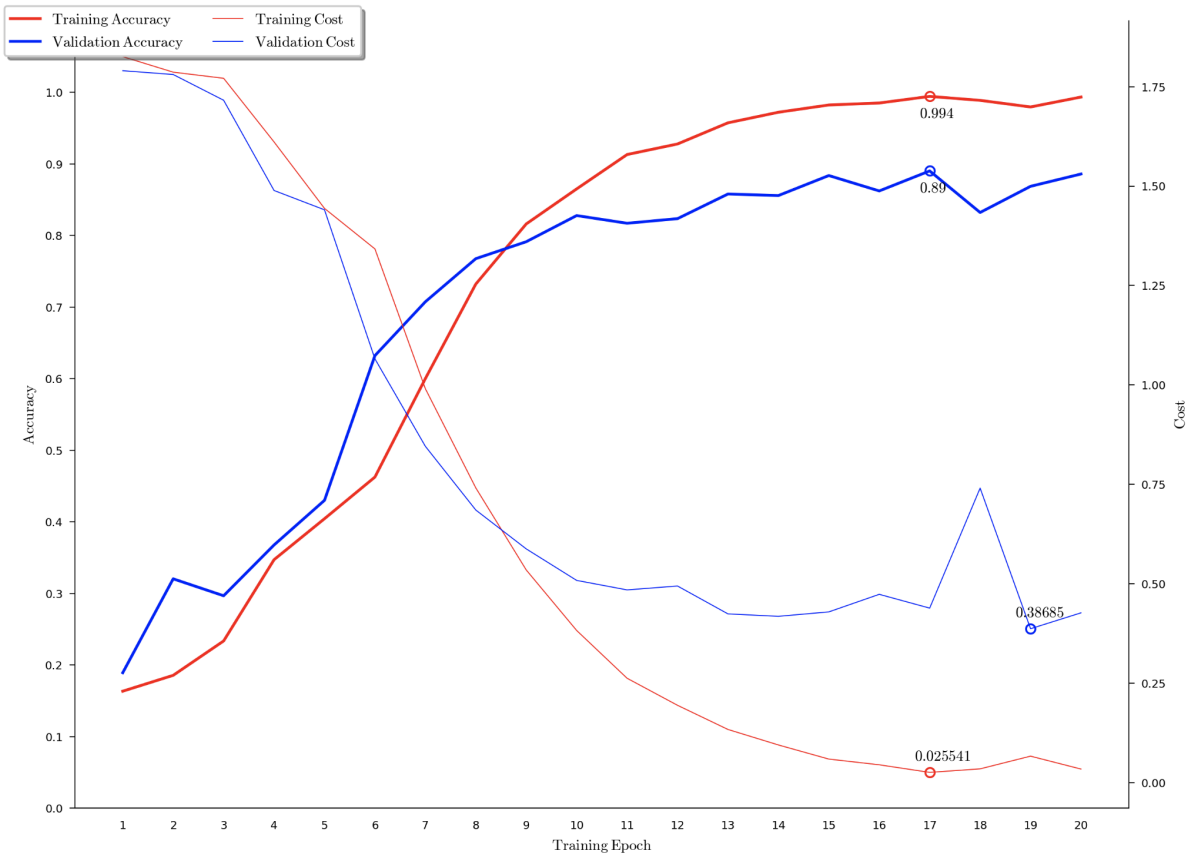


Figure 17: The history for training and the validation from the experiment are plotted showing the accuracy on the y-axis on the left and the cost on the y-axis on the right. A quick observation here tells that the training overfit.

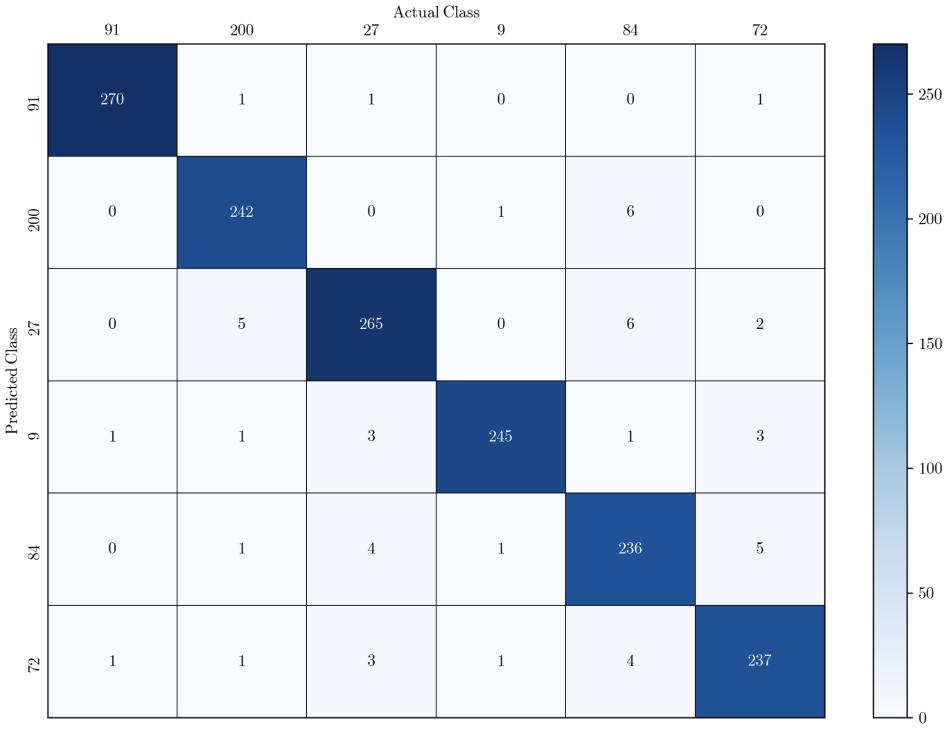


Figure 18: A confusion matrix was produced using the trained model from experiment 2 and was used to revalidate the whole dataset. Confusion matrix for experiment 4. It can be seen that Volunteer 91 has the lowest false-positive and also false-negative.

```
Classification Report:
                      Precision   Recall   F2-score
                      =========   ======   ========
Volunteer 91             0.9926   0.9890     0.9897
Volunteer 200            0.9641   0.9719     0.9703
Volunteer 27             0.9601   0.9532     0.9546
Volunteer 9              0.9879   0.9646     0.9691
Volunteer 84             0.9328   0.9555     0.9508
Volunteer 72             0.9556   0.9595     0.9587

Accuracy (Overall)                           0.9886
Macro-averaged           0.9655   0.9656     0.9656
Weighted-averaged        0.9658   0.9657     0.9657
```

Table 6: The metrics for precision, recall and F2-score for the prediction for the whole dataset using the trained model from experiment 4.

3.6 Predictions

Newly collected samples were cleaned and split. Only samples from four volunteers were successfully collected. For each volunteer, only 10 samples were used for prediction. The correct predictions are summarized in *Table 7*.

Volunteer	Experiment 2	Experiment 3	Experiment 4
200	0	4	1
27	10	7	9
9	3	5	8
84	8	6	5
Variance	*15.6875*	*1.25*	*9.6875*
Standard Deviation	*3.9607*	*1.1180*	*3.1125*

Table 7: Predictions were made using the trained model from each experiment. The samples were never seen neither during training nor validation. The variance and standard deviation for each result were also calculated.

SECTION 4: ANALYSIS, RESULTS AND DISCUSSION

This section analyses and summarizes the results from the experiments and uses the arguments as the ground to choose which model performs the best. In the end, it concludes and answers the research question for this mini-project.

4.0 Analysing Experiment Results

Despite the earlier plan to use raw audio files for the experiment, it was however halted because the training parameters were getting too large to be trained using architecture 1. Therefore, only three different data formats were used; STFT spectrogram, Mel-spectrogram and Mel-spectrogram that was converted into PNG.

Experiment	Training Accuracy	Validation Accuracy	Difference
2	0.950139	0.896774	0.053365
3	0.991690	0.901075	0.090615
4	0.994460	0.890323	0.10123

Table 8: Summarizing the results from experiments 2, 3 and 4 from Section 3. Experiment 1 was not included because it has a very poor training result.

Experiment 1, which was using STFT-spectrogram, had the worst performance as it was unable to make any meaningful learning. Experiment 2 and experiment 3, were using Mel-spectrogram original signal without any scaling. Experiment 4 was done using a mel-spectrogram signal which was scaled to a value between *[0, 254]*.

From *Table 8*, **comparing experiment 2 and experiment 3**, it was found that experiment 2 has a lower difference between training accuracy and validation accuracy, hence it is assumed to have a lower overfitting. However, the accuracy from experiment 3 is better than in experiment 2. **Comparing experiment 2 and experiment 4** in terms of validation accuracy, it seems that experiment 2 fared better and also in terms of overfitting, experiment 2 also fared better when compared to experiment 4. **Comparing experiment 3 and experiment 4**, it can also be seen that experiment 3 has less overfitting and the training accuracy for both experiments are 99 per cent.

From *Figures 13*, *15* and *17*, a quick observation tells that experiment 3 has the largest overfitting. From *Figures 14*, *16* and *18*, another observation can be made by calculating the number of false positives of which there are 54, 47 and 50 for experiments 2, 3 and 4 respectively. These two observations can easily yield experiment 3 as the one with the best-trained model.

Observing the metrics in *Tables 4*, *5* and *6*; the first observation is to check the difference for the values for each metric as produced in *Table 9*. It can be seen that experiment 4 has two metrics with the lowest difference. But when comparing the weighted values for each metric, it can be seen that experiment 3 has the highest values. The weighted values make more sense in considering the number of samples for each volunteer.

Experiment	Precision			Recall			F2-Score		
	Lowest	Highest	Difference	Lowest	Highest	Difference	Lowest	Highest	Difference
2	0.9315	1.0000	0.0685	0.9145	0.9963	0.0818	0.9255	0.9971	0.0716
3	0.9597	0.9963	0.0366	0.9421	0.9927	0.0506	0.9465	0.9934	0.0469
4	0.9327	0.9926	0.0609	0.9512	0.9890	0.0378	0.9508	0.9897	0.0389

Table 9: Summarizing the metrics for precisions, recalls and F2-scores for experiments 2, 3 and 4 by filtering the highest and lowest for each metric. The blue text marked the lowest difference for each metric. A good metric is the one that has lowest difference.

4.1 Best Performance

From the discussion, it was quite clear that experiment 3 and experiment 4 are having the best metrics. However, when taking into consideration that experiment 4 has the highest overfitting, which is an undesirable trait for a trained model, **the best model is concluded to be from experiment 3**. This is also supported by the predictions made in *section 3.6* and *Table 7*, it shows experiment 3 has the lowest variance and standard deviation.[10] This is probably because the model was trained using the original MFCC signal, instead of using the image-converted signal that may have lost a certain amount of data.

4.2 Discussion

This mini-project was an experiment using bespoke, independent and collected from volunteers specifically with certain needs in mind. The reason for collecting instead of using the dataset available from the internet is to allow the dataset expansion when needed and based on what is required. Using a bespoke dataset helps the author to learn first-hand about planning and managing volunteers as well as data organizations. The author also learnt about data cleaning which is also an important part of any data science project. The author has properly answered on how to collect voice samples, organizing and cleaning the data before being used as a dataset. Despite having mentioned in *section 1.3.1* that a proper identification system should have been trained with samples from diverse backgrounds including languages, genders and age-group, this report however did not focus on this due to time and volunteer constraints.

[10] Because of the small number of samples used during prediction, the conclusion is only valid within the discussion of this report. A proper test should take place by enlarging the samples during the training and also a reasonably large independent data for post-training verification.

Secondly, the experiments conducted shows that both Conv1D and Conv2D can perform automatic speaker recognition quite well. Conv1D has the lowest overfitting, however, Conv2D has the best overall performance. It has to be taken into account that the trained dataset was rather small and this may not reflect an ideal situation. Besides, the size of the layers is rather trivial. For example, a complex system should be able to discern the input and produce a robust model with higher correct predictions for each class. Perhaps, increasing the number of training samples per class and also increasing the number of the class itself can solve this triviality.

Thirdly, a problem for automatic speaker recognition using convolutional neural networks is a different problem domain when compared to image recognition. Given an image, patterns can be explained and discerned even by the human eye. For voice signals, there are no specific patterns that can be explained by evaluating a spectrogram. However, certain patterns do exist and can be learned by a simple architecture, as demonstrated from the experiments. Perhaps, this mini-project should have started with a specific sound (phonemes) instead of training using non-specific speeches.

4.3 Conclusion

In answering the research question, *"using a bespoke dataset, how to train minimal neural network architectures to perform automatic speaker recognition?"*, it was concluded that convolution 2D seems to have the most significant result using the original MFCC signal.

The project embarks on exploring convolutional 1D which was suggested to be a suitable convolutional neural network for a problem involving sounds. However, through the experiment, it was shown that convolutional 2D produces better results. In addition to that, it was observed that the original MFCC signal is almost at par when the signals are converted to an image. STFT, which is an almost raw signal, is perhaps too noisy for a network to find any useful pattern.

4.4 Future Works

Several future works are possible. Firstly, the experiments can be extended to increase more samples per volunteer and also to increase the number of volunteers. By having more samples and volunteers (i.e.: classes) may produce a better-trained model. Secondly, from the understanding that signal-to-image conversion seems to produce a better result, perhaps it can be used for a different problem within the same domain. Last but not least, to build an application with a user interface to deploy a trained model to recognize a speaker on a live model instead of a pre-recorded audio signal.

BIBLIOGRAPHY

[1] "Speech2Phone: A Multilingual and Text Independent Speaker Identification Model - Paper Detail." [Online]. Available: https://deeplearn.org/arxiv/118314/speech2phone:-a-multilingual-and-text-independent-speaker-identification-model. [Accessed: 01-Nov-2021]

[2] Z. Bai and X.-L. Zhang, "Speaker Recognition Based on Deep Learning: An Overview," 02-Dec-2020 [Online]. Available: http://arxiv.org/abs/2012.00931. [Accessed: 26-Oct-2021]

[3] G. Fenu, G. Medda, M. Marras, and G. Meloni, "Improving Fairness in Speaker Recognition," 29-Apr-2021 [Online]. Available: http://arxiv.org/abs/2104.14067. [Accessed: 27-Oct-2021]

[4] X. Liu, Sahidullah, and T. Kinnunen, "Learnable MFCCs for Speaker Verification," 20-Feb-2021 [Online]. Available: http://arxiv.org/abs/2102.10322. [Accessed: 27-Oct-2021]

[5] R. Masumura, M. Ihori, A. Takashima, T. Tanaka, and T. Ashihara, "End-to-End Automatic Speech Recognition with Deep Mutual Learning," 16-Feb-2021 [Online]. Available: http://arxiv.org/abs/2102.08154. [Accessed: 26-Oct-2021]

[6] Y. Lei, N. Scheffer, L. Ferrer, and M. McLaren, "A Novel Scheme for Speaker Recognition Using a Phonetically-Aware Deep Neural Network." 2014 [Online]. Available: http://dx.doi.org/10.21236/ada613971

[7] E. Variani, X. Lei, E. McDermott, I. L. Moreno, and J. Gonzalez-Dominguez, "Deep neural networks for small footprint text-dependent speaker verification," *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2014 [Online]. Available: http://dx.doi.org/10.1109/icassp.2014.6854363

[8] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-Vectors: Robust DNN Embeddings for Speaker Recognition," *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2018 [Online]. Available: http://dx.doi.org/10.1109/icassp.2018.8461375

[9] T. Kinnunen and H. Li, "An overview of text-independent speaker recognition: From features to supervectors," *Speech Communication*, vol. 52, no. 1. pp. 12–40, 2010 [Online]. Available: http://dx.doi.org/10.1016/j.specom.2009.08.009

[10] H. Mandalapu *et al.*, "Audio-Visual Biometric Recognition and Presentation Attack Detection: A Comprehensive Survey," *IEEE Access*, vol. 9. pp. 37431–37455, 2021 [Online]. Available: http://dx.doi.org/10.1109/access.2021.3063031

[11] E. Casanova *et al.*, "Speech2Phone: A Multilingual and Text Independent Speaker Identification Model," 25-Feb-2020 [Online]. Available: https://www.arxiv-vanity.com/papers/2002.11213/. [Accessed: 01-Nov-2021]

[12] J. S. Chung *et al.*, "In defence of metric learning for speaker recognition," Mar. 2020, doi: 10.21437/Interspeech.2020-1064. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2020-1064. [Accessed: 01-Nov-2021]

[13] L. Wan, Q. Wang, A. Papir, and I. L. Moreno, "Generalized End-to-End Loss for Speaker Verification," 28-Oct-2017 [Online]. Available: http://arxiv.org/abs/1710.10467. [Accessed: 01-Nov-2021]

[14] S. Yun, J. Cho, J. Eum, W. Chang, and K. Hwang, "An End-to-End Text-independent Speaker Verification Framework with a Keyword Adversarial Network," 06-Aug-2019 [Online]. Available: http://arxiv.org/abs/1908.02612. [Accessed: 01-Nov-2021]

[15] G. Penn, "What is Sound?," 2010 [Online]. Available: http://www.cs.toronto.edu/~gpenn/csc401/soundASR.pdf. [Accessed: 01-Nov-2021]

[16] "Nondestructive Evaluation Physics : Sound." [Online]. Available: https://www.nde-

ed.org/Physics/Sound/components.xhtml. [Accessed: 21-Nov-2021]
[17] J. Nunez-Iglesias, S. van der Walt, and H. Dashnow, *Elegant SciPy*. O'Reilly Media, Inc.
[18] D. Gerhard, "Audio Signal Classification," Jul. 2000 [Online]. Available: http://dx.doi.org/. [Accessed: 21-Nov-2021]
[19] T. N. Sainath, R. J. Weiss, A. Senior, K. W. Wilson, and O. Vinyals, "Learning the speech front-end with raw waveform CLDNNs," *Interspeech 2015*. 2015 [Online]. Available: http://dx.doi.org/10.21437/interspeech.2015-1
[20] H. Purwins, B. Li, T. Virtanen, J. Schluter, S.-Y. Chang, and T. Sainath, "Deep Learning for Audio Signal Processing," *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, no. 2. pp. 206–219, 2019 [Online]. Available: http://dx.doi.org/10.1109/jstsp.2019.2908700
[21] "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences." [Online]. Available: https://ieeexplore.ieee.org/document/1163420. [Accessed: 27-Oct-2021]
[22] A. H. B. Rahman and S. Nielsen, "Probing the Mathematical Model for Multinomial Logistic Regression," Roskilde University, 2021.
[23] Y. Yao *et al.*, "INT8 Winograd Acceleration for Conv1D Equipped ASR Models Deployed on Mobile Devices," 28-Oct-2020 [Online]. Available: http://arxiv.org/abs/2010.14841. [Accessed: 21-Nov-2021]
[24] M. Saini, U. Satija, and M. D. Upadhayay, "Light-Weight 1-D Convolutional Neural Network Architecture for Mental Task Identification and Classification Based on Single-Channel EEG," 12-Dec-2020 [Online]. Available: http://arxiv.org/abs/2012.06782. [Accessed: 21-Nov-2021]
[25] D. Barina, "Comparison of Lossless Image Formats," 25-Jun-2021 [Online]. Available: http://arxiv.org/abs/2108.02557. [Accessed: 05-Nov-2021]
[26] R. Charan, A. Manisha, K. Ravichandran, and R. Muthu, "A text-independent speaker verification model: A comparative analysis," *2017 IEEE International Conference on Intelligent Computing and Control (I2C2)*, Jun. 2017, doi: 10.1109/I2C2.2017.8321794. [Online]. Available: https://www.researchgate.net/publication/321510709_A_text-independent_speaker_verification_model_A_comparative_analysis. [Accessed: 21-Nov-2021]

BIBLIOGRAPHY

[1] "Speech2Phone: A Multilingual and Text Independent Speaker Identification Model - Paper Detail." [Online]. Available: https://deeplearn.org/arxiv/118314/speech2phone:-a-multilingual-and-text-independent-speaker-identification-model. [Accessed: 01-Nov-2021]

[2] Z. Bai and X.-L. Zhang, "Speaker Recognition Based on Deep Learning: An Overview," 02-Dec-2020 [Online]. Available: http://arxiv.org/abs/2012.00931. [Accessed: 26-Oct-2021]

[3] G. Fenu, G. Medda, M. Marras, and G. Meloni, "Improving Fairness in Speaker Recognition," 29-Apr-2021 [Online]. Available: http://arxiv.org/abs/2104.14067. [Accessed: 27-Oct-2021]

[4] X. Liu, Sahidullah, and T. Kinnunen, "Learnable MFCCs for Speaker Verification," 20-Feb-2021 [Online]. Available: http://arxiv.org/abs/2102.10322. [Accessed: 27-Oct-2021]

[5] R. Masumura, M. Ihori, A. Takashima, T. Tanaka, and T. Ashihara, "End-to-End Automatic Speech Recognition with Deep Mutual Learning," 16-Feb-2021 [Online]. Available: http://arxiv.org/abs/2102.08154. [Accessed: 26-Oct-2021]

[6] Y. Lei, N. Scheffer, L. Ferrer, and M. McLaren, "A Novel Scheme for Speaker Recognition Using a Phonetically-Aware Deep Neural Network." 2014 [Online]. Available: http://dx.doi.org/10.21236/ada613971

[7] E. Variani, X. Lei, E. McDermott, I. L. Moreno, and J. Gonzalez-Dominguez, "Deep neural networks for small footprint text-dependent speaker verification," *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2014 [Online]. Available: http://dx.doi.org/10.1109/icassp.2014.6854363

[8] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-Vectors: Robust DNN Embeddings for Speaker Recognition," *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2018 [Online]. Available: http://dx.doi.org/10.1109/icassp.2018.8461375

[9] T. Kinnunen and H. Li, "An overview of text-independent speaker recognition: From features to supervectors," *Speech Communication*, vol. 52, no. 1. pp. 12–40, 2010 [Online]. Available: http://dx.doi.org/10.1016/j.specom.2009.08.009

[10] H. Mandalapu *et al.*, "Audio-Visual Biometric Recognition and Presentation Attack Detection: A Comprehensive Survey," *IEEE Access*, vol. 9. pp. 37431–37455, 2021 [Online]. Available: http://dx.doi.org/10.1109/access.2021.3063031

[11] E. Casanova *et al.*, "Speech2Phone: A Multilingual and Text Independent Speaker Identification Model," 25-Feb-2020 [Online]. Available: https://www.arxiv-vanity.com/papers/2002.11213/. [Accessed: 01-Nov-2021]

[12] J. S. Chung *et al.*, "In defence of metric learning for speaker recognition," Mar. 2020, doi: 10.21437/Interspeech.2020-1064. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2020-1064. [Accessed: 01-Nov-2021]

[13] L. Wan, Q. Wang, A. Papir, and I. L. Moreno, "Generalized End-to-End Loss for Speaker Verification," 28-Oct-2017 [Online]. Available: http://arxiv.org/abs/1710.10467. [Accessed: 01-Nov-2021]

[14] S. Yun, J. Cho, J. Eum, W. Chang, and K. Hwang, "An End-to-End Text-independent Speaker Verification Framework with a Keyword Adversarial Network," 06-Aug-2019 [Online]. Available: http://arxiv.org/abs/1908.02612. [Accessed: 01-Nov-2021]

[15] G. Penn, "What is Sound?," 2010 [Online]. Available: http://www.cs.toronto.edu/~gpenn/csc401/soundASR.pdf. [Accessed: 01-Nov-2021]

[16] "Nondestructive Evaluation Physics : Sound." [Online]. Available: https://www.nde-

ed.org/Physics/Sound/components.xhtml. [Accessed: 21-Nov-2021]
[17] J. Nunez-Iglesias, S. van der Walt, and H. Dashnow, *Elegant SciPy*. O'Reilly Media, Inc.
[18] D. Gerhard, "Audio Signal Classification," Jul. 2000 [Online]. Available: http://dx.doi.org/. [Accessed: 21-Nov-2021]
[19] T. N. Sainath, R. J. Weiss, A. Senior, K. W. Wilson, and O. Vinyals, "Learning the speech front-end with raw waveform CLDNNs," *Interspeech 2015*. 2015 [Online]. Available: http://dx.doi.org/10.21437/interspeech.2015-1
[20] H. Purwins, B. Li, T. Virtanen, J. Schluter, S.-Y. Chang, and T. Sainath, "Deep Learning for Audio Signal Processing," *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, no. 2. pp. 206–219, 2019 [Online]. Available: http://dx.doi.org/10.1109/jstsp.2019.2908700
[21] "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences." [Online]. Available: https://ieeexplore.ieee.org/document/1163420. [Accessed: 27-Oct-2021]
[22] A. H. B. Rahman and S. Nielsen, "Probing the Mathematical Model for Multinomial Logistic Regression," Roskilde University, 2021.
[23] Y. Yao *et al.*, "INT8 Winograd Acceleration for Conv1D Equipped ASR Models Deployed on Mobile Devices," 28-Oct-2020 [Online]. Available: http://arxiv.org/abs/2010.14841. [Accessed: 21-Nov-2021]
[24] M. Saini, U. Satija, and M. D. Upadhayay, "Light-Weight 1-D Convolutional Neural Network Architecture for Mental Task Identification and Classification Based on Single-Channel EEG," 12-Dec-2020 [Online]. Available: http://arxiv.org/abs/2012.06782. [Accessed: 21-Nov-2021]
[25] D. Barina, "Comparison of Lossless Image Formats," 25-Jun-2021 [Online]. Available: http://arxiv.org/abs/2108.02557. [Accessed: 05-Nov-2021]
[26] R. Charan, A. Manisha, K. Ravichandran, and R. Muthu, "A text-independent speaker verification model: A comparative analysis," *2017 IEEE International Conference on Intelligent Computing and Control (I2C2)*, Jun. 2017, doi: 10.1109/I2C2.2017.8321794. [Online]. Available: https://www.researchgate.net/publication/321510709_A_text-independent_speaker_verification_model_A_comparative_analysis. [Accessed: 21-Nov-2021]

APPENDIX 1: TRAINING RESULTS

A-1: Results from Experiment 1

Training Accuracy	Validation Accuracy	Training Cost	Validation Cost
0.17359186708927155	0.1548387110233307	2.5024142265319824	1.7967803478240967
0.16528162360191345	0.1548387110233307	1.7957298755645752	1.7930524349212646
0.1560480147600174	0.2408602088689804	1.7950879335403442	1.786411166191101
0.20775623619556427	0.17634408175945282	1.7658274173736572	1.7164133787155151
0.22899353504180908	0.2107526808977127	1.6895442008972168	1.6869875192642212
0.2686980664730072	0.20645160973072052	1.650346279144287	1.6791434288024902
0.27885502576828003	0.28172042965888977	1.6374939680099487	1.6693812608718872
0.30286240577697754	0.26021504402160645	1.6053173542022705	1.6895935535430908
0.3684210479259491	0.19569893181324005	1.5519777536392212	1.7671263217926025
0.31024929881095886	0.25376343727111816	1.6332030296325684	1.7239041328430176
0.3416435718536377	0.29247310757637024	1.5731984376907349	1.7132271528244019
0.3610341548919678	0.37204301357269287	1.5374480485916138	1.719454050064087
0.3896583616733551	0.22365590929985046	1.5997495651245117	1.7911769151687622
0.349953830242157	0.23010753095149994	1.5796915292739868	2.0761396884918213
0.23453369736671448	0.22580644488334656	1.6969268321990967	1.7313263416290283
0.25207754969596863	0.20645160973072052	1.6614972352981567	1.7027029991149902
0.30286240577697754	0.20430107414722443	1.6138150691986084	1.7856262922286987
0.32409971952438354	0.2107526808977127	1.5930665731430054	1.8445533514022827
0.349030464887619	0.23655913770198822	1.5629888772964478	1.9173469543457031
0.38042473793029785	0.26021504402160645	1.4937498569488525	2.0259015560150146

Table A-1: The recorded training and validation accuracies as well as training and validation costs for experiment 1.

A-2: Results from Experiment 2

Training Accuracy	Validation Accuracy	Training Cost	Validation Cost
0.16620498895645142	0.17419354617595673	2.031454563140869	1.786756992340088
0.1828254908323288	0.14408601820468903	1.7884165048599243	1.7786675691604614
0.2760849595069885	0.35483869910240173	1.7421584129333496	1.6758956909179688
0.39150509238243103	0.40645161271095276	1.574633002281189	1.4190149307250977
0.47460755705833435	0.47311827540397644	1.3243716955184937	1.2142705917358398
0.5614035129547119	0.6666666865348816	1.191212773323059	0.9448384046554565
0.6565096974372864	0.7612903118133545	0.9477810859680176	0.7935056686401367
0.7146814465522766	0.7978494763374329	0.7825384736061096	0.6573032140731812
0.7811634540557861	0.7806451320648193	0.6568336486816406	0.6282601356506348
0.7516158819198608	0.823655903339386	0.6942753195762634	0.5817884802818298
0.8301015496253967	0.8301075100898743	0.4943067133426666	0.5056200623512268
0.8688827157020569	0.8451613187789917	0.377014696598053	0.41746076941490173
0.8790397047996521	0.8451613187789917	0.35156795382499695	0.4551532566547394
0.8928900957107544	0.8580645322799683	0.3079644739627838	0.4095504879951477
0.9012003540992737	0.8709677457809448	0.2700614035129547	0.3939726650714874
0.9122806787490845	0.8795698881149292	0.24691639840602875	0.40318426489830017
0.9122806787490845	0.8709677457809448	0.2259184718132019	0.3801648020744324
0.9344413876533508	0.8946236371994019	0.1851116269826889	0.33476364612579346
0.9344413876533508	0.8709677457809448	0.17636241018772125	0.4303582012653351
0.950138509273529	0.896774172782898	0.13751842081546783	0.3211125135421753

Table A-2: The recorded training and validation accuracies as well as training and validation costs for experiment 2.

A-3: Results from Experiment 3

Training Accuracy	Validation Accuracy	Training Cost	Validation Cost
0.17636196315288544	0.25591397285461426	1.8310339450836182	1.7920273542404175
0.25392428040504456	0.34408602118492126	1.7604656219482422	1.66477632522583
0.38504156470298767	0.47741934657096863	1.5374611616134644	1.3143196105957031
0.48014774918556213	0.5182795524597168	1.3829762935638428	1.296012282371521
0.5493997931480408	0.5849462151527405	1.2283403873443604	1.122246503829956
0.6223453283309937	0.6602150797843933	0.9964065551757812	0.8820152878761292
0.7100646495819092	0.7247312068939209	0.7781066298484802	0.7568922638893127
0.7654662728309631	0.8129032254219055	0.6638185977935791	0.6238830089569092
0.838411808013916	0.8064516186714172	0.48271116614341736	0.527973473072052
0.8928900957107544	0.8365591168403625	0.35432684421539307	0.43411189317703247
0.9168975353240967	0.8774193525314331	0.2394752949476242	0.3700931668281555
0.9427515864372253	0.8623656034469604	0.1687755137681961	0.37732094526290894
0.9602954983711243	0.85161292552948	0.12004192173480988	0.4010041654109955
0.9741458892822266	0.8881720304489136	0.10425132513046265	0.35334062576293945
0.9667590260505676	0.8559139966964722	0.09554847329854965	0.3907422721385956
0.9741458892822266	0.8731182813644409	0.0878167524933815	0.40454384684562683
0.9713758230209351	0.8838709592819214	0.0853293165564537	0.3449953496456146
0.9704524278640747	0.8623656034469604	0.08155883103609085	0.4464591145515442
0.9796860814094543	0.8838709592819214	0.05849064141511917	0.3962807059288025
0.9916897416114807	0.9010752439498901	0.029655488207936287	0.35953497886657715

Table A-3: The recorded training and validation accuracies as well as training and validation costs for experiment 3.

A-4: Results from Experiment 4

Training	Validation	Training Cost	Validation Cost
0.16343490779399872	0.18924731016159058	1.8256384134292603	1.7902758121490479
0.1855955719947815	0.32043009996414185	1.7866135835647583	1.780698537826538
0.2336103469133377	0.2967741787433624	1.7712167501449585	1.7159982919692993
0.3471837341785431	0.3677419424057007	1.6111059188842773	1.489024043083191
0.40443214774131775	0.4301075339317322	1.4439584016799927	1.4404237270355225
0.4626038670539856	0.6322580575942993	1.3421062231063843	1.0656096935272217
0.6001846790313721	0.7075268626213074	0.9904987812042236	0.84563809633255
0.7322252988815308	0.7677419185638428	0.741083025932312	0.6853950023651123
0.8162511587142944	0.7913978695869446	0.5346901416778564	0.5876592993736267
0.8651893138885498	0.8279569745063782	0.3824532628059387	0.508383572101593
0.9132040739059448	0.8172042965888977	0.2622700035572052	0.4844874143600464
0.9279778599739075	0.823655903339386	0.19426797330379486	0.4942777156829834
0.957525372505188	0.8580645322799683	0.1335093230009079	0.42418172955513
0.9722991585731506	0.8559139966964722	0.09453204274177551	0.41801872849464417
0.9824561476707458	0.8838709592819214	0.05900990217924118	0.4291740953922272
0.9852262139320374	0.8623656034469604	0.04463599622249603	0.47335419058799744
0.9944598078727722	0.8903225660324097	0.02554064616560936	0.43866077065467834
0.9889196753501892	0.8322580456733704	0.03438803181052208	0.7405208349227905
0.9796860814094543	0.8688172101974487	0.06637159734964371	0.3868517577648163
0.9935364723205566	0.8860214948654175	0.033907003700733185	0.4267832636833191

Table A-4: The recorded training and validation accuracies as well as training and validation costs for experiment 4.