TheThirdEye

Audio identification with headphones

Final project in Software Engineering

sce

המכללה האקדמית להנדסה ע"ש סמי שמעון

מהנדסים לעולם טוב יותר!

בסביבת PROJECT ORIENTED

The Third Eye
Audio identification with headphones

Project number: 50

Submitted in the Software Engineering Department
Shamoon College of Engineering (SCE)

By:
Tal gradus
Olga lapovsky
Alex Skatkov

Academic advisor approval: _____
Head of department approval: _____

Hebrew date
Sivan 5780

Civil date
June 2020

Beer Sheva

We would like to express our very great appreciation to our academic advisors
Natalia Vanetik Ph.D,Marina Litvak Ph.D
for their valuable and constructive suggestions during the planning and development of this final project. Their willingness to give their time so generously has been very much appreciated.

Content

1. Introduction

In today's work space environment exists a fundamental problem. Most of the work environments are designed as an open space which has some disadvantages, such as less productivity and low morale of the employees , and the main issue is the constant noise due to lack of privacy.

Most employees try to combat this problem by using headphones to listen to music or using the noise cancelling function. This solution creates another problem: a lack of awareness to the environment which causes communication problems between the team. [32]

Currently there is only one known attempt in this field that is published that was made by amazon to create noise canceling headphones that will recognize when a person is being called, upon recognition the headphones temporarily stop canceling noise so that the headphone wearer could hear outside sounds [35]. Currently this product did not launch and was officially approved as a patent only in the last month [34] (December 2019). If amazon did launch the headphones it is still not a perfect solution for all because not everyone can afford a new pair of headphones.

Our system tries to offer a solution that allows the employee to work without distraction or noise and yet to be aware of his environment and his team mates. Our solution is to create a tool which will inform the employee when he is called by name and specifies who called him, thus lowers the communication issues.

The benefits of our system are:

- Our solution does not require special operation while in use.
- Allows communication while using headphones.
- Creates a pleasant work environment.

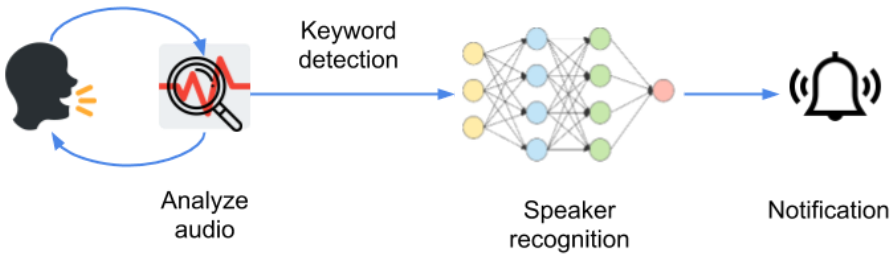


Figure 1
Pipeline of the system.

2. Literature review

2.1 Audio processing

Audio processing covers many diverse fields, all involved in presenting sound to human listeners.
It is the discipline of manipulating audio signals, can be represented as a digital or an analog format in both the time and frequency domains and typically measured in decibels.
Audio processing has three main application areas: high fidelity music reproduction(enhanced music experiences),voice telecommunications(communication of sound over a distance), and synthetic speech(has many applications such as Text-to-speech systems, Speaker verification systems and so on). [9]
Thanks to those applications we arrived to a point where technology has the ability to understand us in some manner , due to that automating daily tasks became easier.

2.1.1 Speech recognition

Speech recognition is the ability of a system to identify words and phrases in spoken language and convert them to text.It uses acoustic modeling and language modeling algorithms. Speech recognition is a very complex problem because speech vary in terms of accent,pronunciation, articulation, roughness, nasality, pitch, volume, and speed. Speech can also be distorted by background noise and echoes. Some applications of speech recognition are home automation,interactive voice response and hands free computing. [12]
Examples of those applications are google home,alexa and google translate voice features, all of these tools are just a little piece of how speech recognition plays an important part in our daily life.

2.1.2 Speaker recognition

Speaker recognition is a process of identifying a speaker by its vocal features.
It operates by identifying the same keyword or a wide variety of keywords by extracting features and comparing with voice samples.It can also be used for verification and classification of an individual speaker, this allows improved security measures in many systems by identifying the person with a more accurate and simplified verification. In current days we need this more than ever since almost everything in our life is virtual. [2]

2.1.3 Techniques of implantation for speaker and speech recognition

The following models can be implemented in both speaker and speech recognition due to the similarity between these areas.

2.1.3.1 Gaussian mixture models

The Gaussian mixture model(GMM)[2] is a model that expresses the probability density function of a random variable in terms of a weighted sum of its components, each of which is described by a Gaussian(normal) density function:

$$p(x|\varphi) = \sum_{y=1}^{\Gamma} p(x|\theta\gamma)P(\theta\gamma)$$

GMM is a powerful algorithm for *clustering*[1], it can be used to cluster unlabeled data.
The applications of GMMs are feature extraction from speech data and object tracking of multiple objects.

2.1.3.2 Hidden Markov Model(HMM)

A hidden Markov model (HMM) is a statistical model that can be used to describe the evolution of observable events that depend on internal factors, which are not directly observable.
An HMM consists of two stochastic processes, namely, an invisible process of hidden states and a visible process of observable symbols. [13]
HMMs are a formal foundation for making probabilistic models of linear sequence labeling problems, they are known for their application in reinforcement learning and temporal pattern recognition such as speech, handwriting, gesture recognition, part of speech tagging, musical score following, partial discharges and bioinformatics.

[1] clustering is grouping similar data points together, based on their attributes or features

2.1.3.3 Neural networks

A neural network is a series of algorithms that endeavors to recognize underlying relationships in a set of data through a process that mimics the way the human brain operates.
ANNs have the ability to learn and model non-linear and complex relationships,after learning from the initial inputs and their connections, it can conclude a connection on an unknown data. The general architecture of ANN is composed of an input layer,hidden layers and output layer,each layer is built from artificial neurons called nodes.In the node the calculation occurs,the input with the weight is calculated and the sum passes through activation function that determines if the signal will be passed to the next layer. [8]

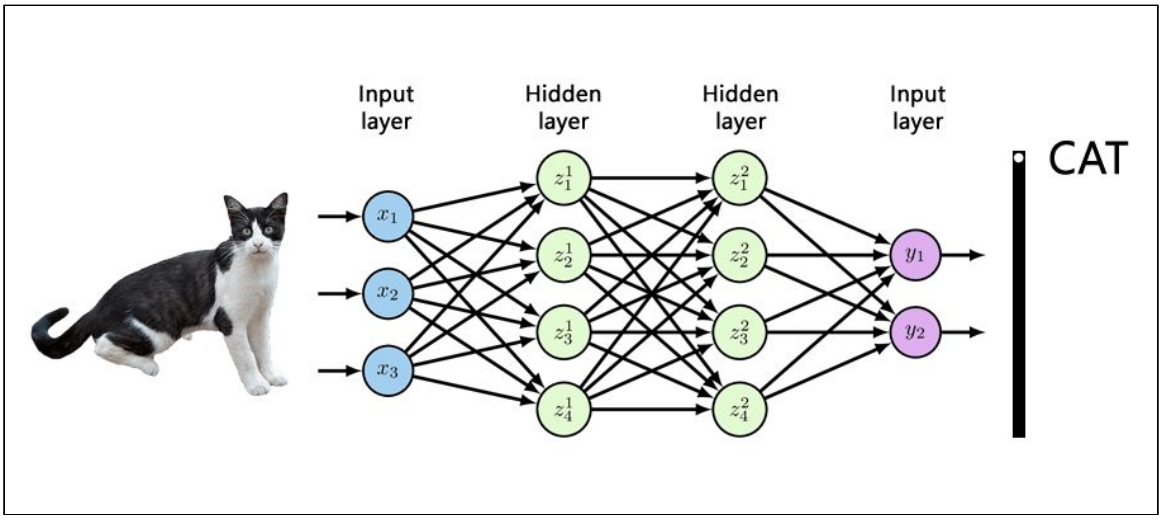


Figure 2
General structure of ANN

There are many ways to implement neural networks.

2.1.3.3.1 Feedforward neural network

Feedforward neural network is the simplest type of artificial neural network,it is primarily used for supervised learning in cases where the data learned is neither sequential nor time dependent.
In a feedforward neural network, the signal moves in only one direction, forward, from the input nodes, through the hidden nodes and the output nodes.In this type of neural network there aren't cycles or loops.
The simplest type of feedforward neural network is the perceptron, a feedforward neural network with no hidden units. [6]

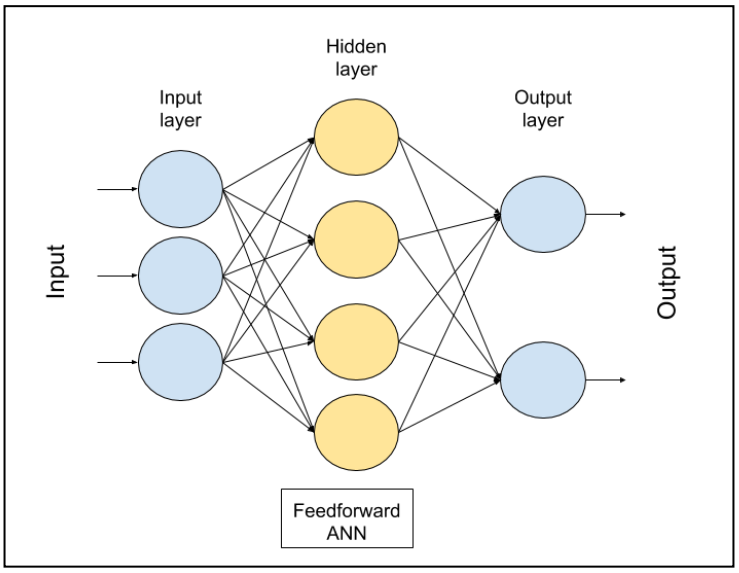


Figure 3
General structure of feedforward ANN

2.1.3.3.2 Recurrent neural network

In traditional neural networks, all the inputs and outputs are independent of each other, but in cases where it is required to predict the next step in a sequence, for example the next word of a sentence, there's a need for the previous words therefore its required to remember them, thus recurrent neural networks came into existence.
Recurrent neural network is a type of ANN where the output from the previous step is transmitting an input into the current step.
The main and most important feature of RNN is hidden state which is a representation of previous inputs.
Today it's commonly used in speech recognition and natural language processing. [7]

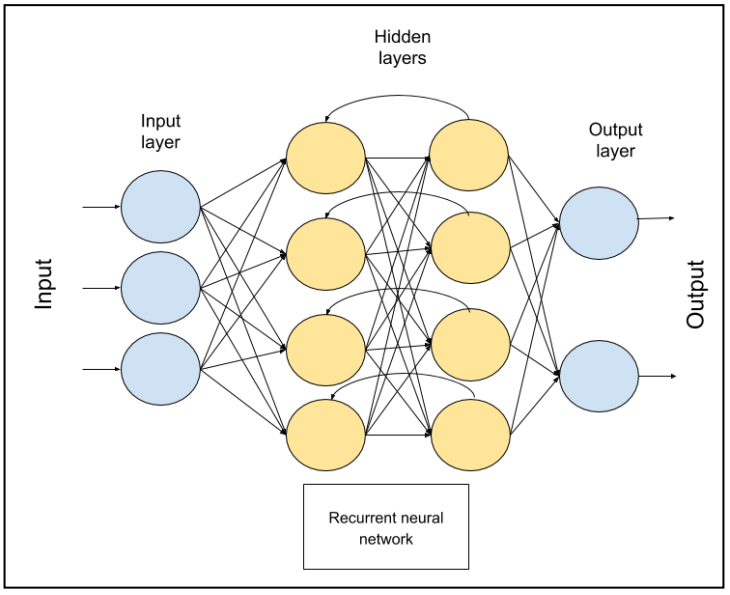


Figure 4
General structure of RNN

Even though RNNs helped in many ways it still had some issues, one of the problems was the vanishing gradient which is essentially the lower the *gradient*[2] is, the harder it is for the network to update the weights and the longer it takes to get to the final result.
The solutions to the vanishing gradient are:

- LSTM
  LSTM is a type of RNN, capable of learning long-term dependencies.
  It has three *gates*[3] that update and control the cell *states*[4], the forget gate that decides what information should be removed or kept, input gate that decides which values will be updated by transforming the values to be between 0 and 1, and the output gate that decides what the next hidden state should be.
  The gates use sigmoid activation and hyperbolic tangent functions.
  LSTMs solve the vanishing gradient problem by creating a connection between the forget gate activations and the gradients computation, this connection creates a path for information to flow through the forget gate this way the LSTM doesn't forget desired information. [14]

- GRU
  The GRU is the newer generation of RNN, it got rid of the cell state and used the hidden state to transfer information. It has only two gates, a reset gate that decides how much past information to forget, and update gate that decides what information to remove and what new information to add.
  GRU eliminates the vanishing gradient problem since the model is not removing the new input every single time but keeps the relevant information and passes it down to the next steps of the network. [14]

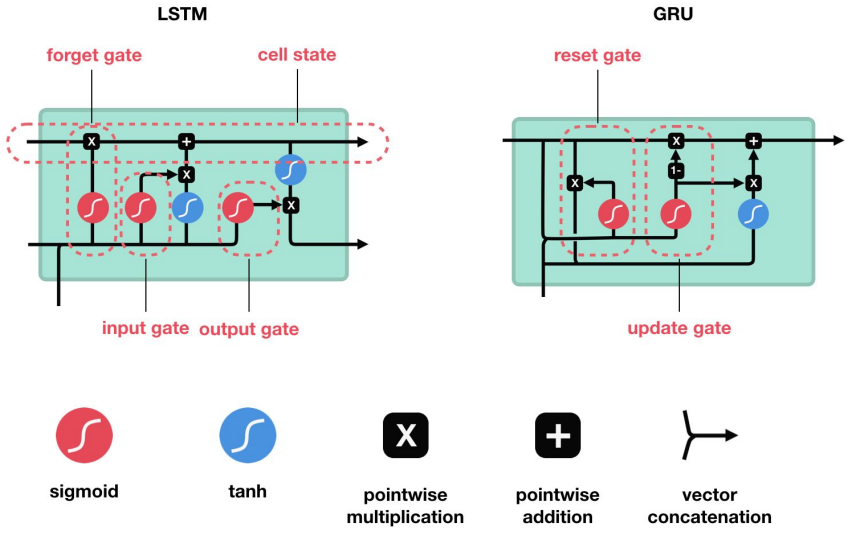


Figure 5
LSTM AND GRU Cells and their Operations

[2] direction and magnitude calculated during the training of a neural network that is used to update the network weights in the right direction and by the right amount
[3] internal mechanisms that can regulate the flow of information. They are composed out of a sigmoid neural net layer and a pointwise multiplication operation.
[4] The cell state is kind of like a conveyor belt. It runs straight down the entire chain, with only some minor linear interactions. It's very easy for information to just flow along it unchanged

2.1.3.3.3 CNN

A convolutional neural network is one of the basic neural networks architectures.It usually consists of an input and an output layer, as well several hidden layers such as:convolutional layers,pooling layers,fully-connected layers. The activation function is commonly RELU.

Mostly applied to image analyzing,video recognition and language analyzing,it has the ability to differentiate one item from another, giving us the capability to categorize an audio sample.[36]

2.1.3.4 Neural networks for small datasets

Most neural network models today can't handle a small amount of dataset due to overfitting. But there are several ANN types that were created to handle this problem.

2.1.3.4.1 Siamese Networks

Siamese neural network is an advancement in neural networks. It is one of the simplest and most popularly used one-shot learning algorithms which is a learning technique where we learn from only one training example per class.
This neural network contains two or more identical subnetwork components that share parameters and weights,both joined together at the end using an energy function which can be any similarity measure, such as Euclidean distance and cosine similarity.
They became popular among tasks that involve finding similarity or a relationship between two comparable things.
The applications of siamese networks are endless, they've been stacked with various architectures for performing various tasks such as human action recognition, scene change detection, and machine translation. [5]

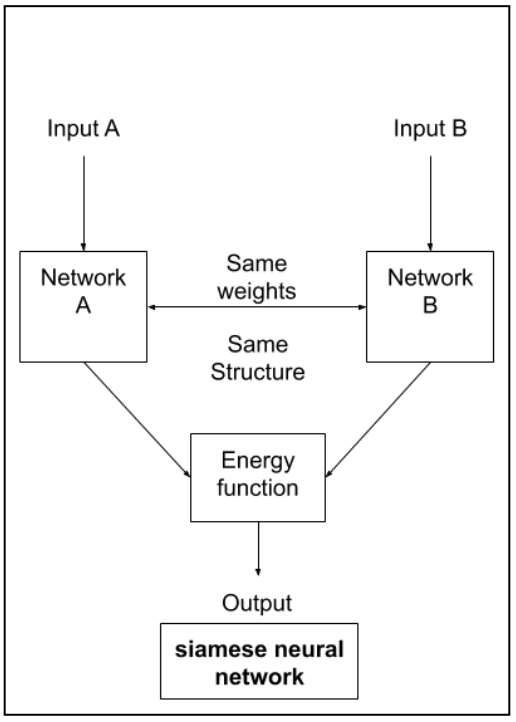


Figure 6
General structure of a siamese neural network

2.1.3.4.2 Prototypical networks

Prototypical network is mainly used when there are small amounts of data for classification problems.
It creates a prototypical representation of each class and classifies a query point based on the distance between the class prototype and the query point.
Prototypical networks compute an M-dimensional representation or prototype, of each class through an embedding function with learnable parameters. Each prototype is the mean vector of the embedded support points belonging to its class. [10]

$$Class\ Prototype\ (C) = \frac{1}{S} \sum_{(Xi,Yi) \in s} F\phi(Xi)$$

mean embeddings for each class

$$P\phi(Y = K|X) = \frac{exp(-d(F\phi(X),C))}{\sum_{K} exp(-d(F\phi(X),C))}$$

the probability of the class of a query set of points by applying softmax over the distance d to determine the classification

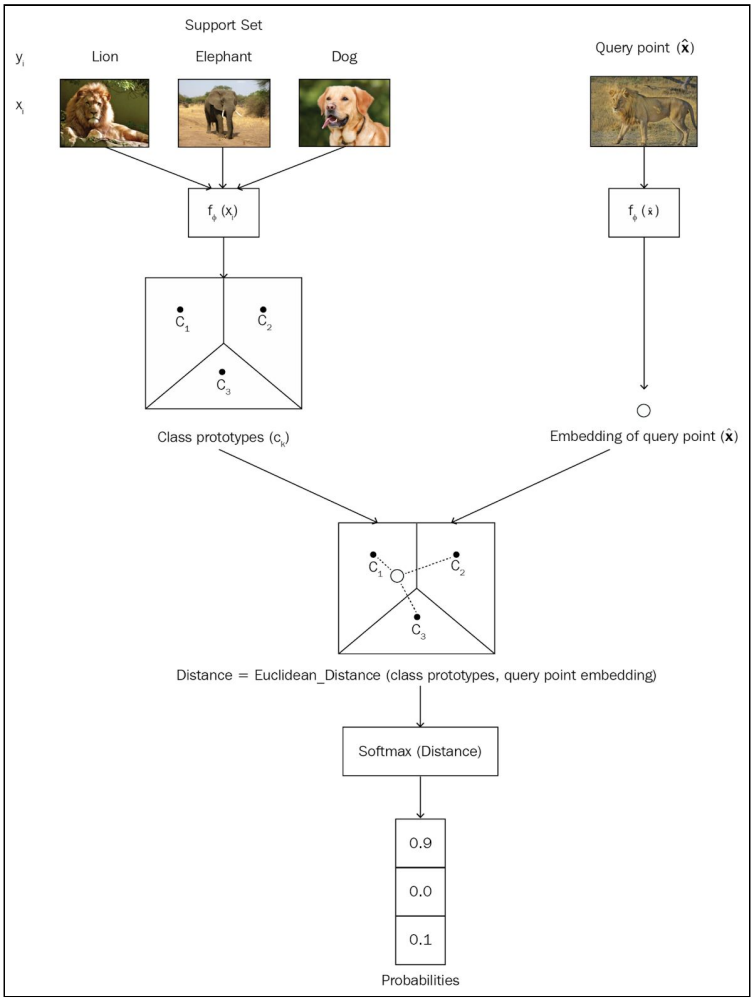


Figure 7
General structure of a prototypical neural network

There are a few variants of prototypical networks:

◄ Gaussian prototypical network

Gaussian prototypical network is mainly used for classifying noisy and less homogeneous data.
It generates embeddings for the data points, and adds confidence region around them characterized by a Gaussian covariance matrix. [4]
The confidence region contributes in characterizing the quality of individual data points.
The output of the encoder will be embeddings, as well as the covariance matrix.

◄ Semi-prototypical network

There are several occasions where the data does not belong to any category currently defined in the network therefore it gets classified incorrectly.Semi prototypical network is used for handling unlabeled data.
It uses k-means in order to assign all the unlabeled data in a distractor class.
K-means is a clustering method that groups the data based on their closeness to each other according to the Euclidean distance. [1]
The distractor class handles the data by computing a threshold which concludes if the unlabeled data is added or ignored. [3]

2.2 Software And Hardware

2.2.1 Hardware

2.2.1.1 Computer

A standard mobile or immobile computer which includes a windows operating system version 7 and up, With a memory of minimum 4 GB RAM.

2.2.1.2 Microphone

A standard microphone built in or external with Wide Frequency Response and High Sensitivity such as Dynamic Microphones which are cheap, durable and sound good, Large Diaphragm Condenser Microphones which are used mostly in professional recording studios, Small Diaphragm Condenser Microphones which have great transient response, extended top end, and consistent pickup patterns, and Ribbon Microphones with warm, vintage tones.

2.2.2 Software

The following libraries are free.

2.2.2.1 Neural network

- TensorFlow

An open source artificial intelligence library,that is using data flow graphs to build models. It allows developers to create large-scale neural networks with many layers. [15]

◀ Keras

Keras[16] is a high-level neural networks API, written in Python[33] and capable of running on top of TensorFlow[15], CNTK[24], or Theano[25].

2.2.2.2 Audio recording

◀ pyaudio

PyAudio[20] provides Python[33] bindings for PortAudio[29], the cross-platform audio I/O library. With PyAudio, you can easily use Python to play and record audio on a variety of platforms.
Advantages: records and plays audio,low-level control.
Disadvantages: playing an audio is complex,records in a format of bytes objects.

◀ python-sounddevice

This Python[33] module provides bindings for the PortAudio[29] library and a few convenience functions to play and record NumPy[17] arrays containing audio signals. [28]
Advantages: records in a format of NumPy arrays.
Disadvantages: only records audio.

2.2.2.3 Audio manipulation:

◀ LibROSA

It's a python package for music and audio analysis. [19]
Advantages: ease of implementation,ease of use,ease of interoperability with other libraries,compatible with Python 3,more commonly used.
Disadvantages: librosa's functionality does not always apply easily to large audio files.

◀ pyAudioAnalysis

It's a Python[33] library covering a wide range of audio analysis tasks, including: feature extraction, classification, segmentation and visualization.
Advantages: compatible with Python3, can convert mp3 format to wav format.
Disadvantages: rarely used in software implantation. [31]

2.2.2.4 Data representation

◀ Numpy

NumPy is the fundamental package for scientific computing with Python[33]. NumPy can also be used as an efficient multi-dimensional container of generic data.It replaces the use of built-in list data structures. [17]
advantages: higher computing performance,frequently used by many libraries.

2.2.2.5 Data mining libraries

◀ scikit-learn package

scikit-learn[18] is a Simple and efficient tool for data mining and data analysis used to build a machine learning model, built on NumPy[17], SciPy[26], and matplotlib[27].
Advantages: accessible to everybody, and reusable in various contexts, good and clean documentation.
Disadvantages: doesn't have a lot of flexibility,can be slow.

◀ mlpy

mlpy is a Python[33] module for Machine Learning built on top of NumPy[17] and SciPy[26]. It provides a wide range of state-of-the-art machine learning methods. [30]
Advantages: provides a wide range of state of the art machine learning methods,compatible with python 3.
Disadvantages: outdated library.

2.3 Data

The data is a set of audio files of a number of speakers with several recordings of themselves speaking. The following datasets fit this criteria.

2.3.1 Free spoken digit dataset

An audio dataset consisting of recordings of spoken digits, each digit has 50 recordings with English pronunciations.
In total there is 2000 recordings and each recording is 1s long, with 4 different speakers (500 recordings each). It's in a format of wav files at 8kHz at total size of 12MB.
The recordings are with minimal silence at the beginning and at the end. [21]

2.3.2 Chime

The Chime dataset is a collection of annotated domestic environment audio recordings. [22]
It has 4 different speakers with English pronunciations in different noisy locations but also includes clean recordings.
In total there are 12274 recordings each recording is 4s long with an approximate size of 4GB in the format of wav files.

2.3.3 TIMIT Corpus

The dataset contains recordings of 630 speakers with 8 different dialects of american english. Each speaker reads ten phonetically rich sentences in an average size of 4s with a total of 6300 recordings,in a format of wav files at 16 kHz with a total size of 440MB. [23]

3. Initiation & Characterization

3.1 Initiation

0. Administration

- This system was developed to improve the working environment in open space complexes.
- With this system headphones can be used while being aware of the surroundings.
- The system provides an efficient and clean interface.
- The system pledges to comply with the conditions and objectives set out in this document.
- Developers: Tal Gradus, Olga Lapovsky, Alex Skatkov.

1. Goals

1.1 Client / application specialist

1.1.1 Users

Software developers:
- Tal Gradus
Email: talgr3@ac.sce.ac.il.
- Olga Lapovsky
Email: olgala1@ac.sce.ac.il.
- Alex Skatkov
Email: alexsk@ac.sce.ac.il.
Software developers will be the system maintainers, responsible for the ongoing management of the software.

1.1.2 Application specialist

Natalia Vanetik Ph.D
Email: natalyav@sce.ac.il

Marina Litvak Ph.D
Email: marinal@sce.ac.il

When the project begins, the application specialists will participate and accompany the characterization of the system.

1.2 Goals and Objectives

Objective 1: Convenient and clean user interface

current state	**Desirable condition**	**How long from the activation system**	**Priority**
Basic GUI	Clean and convenient	Week	Low

Objective 2: Quick speaker recognition

current state	**Desirable condition**	**How long from the activation system**	**Priority**
Slow	Fast	Month	High

Objective 3: High detection percentage

current state	**Desirable condition**	**How long from the activation system**	**Priority**
Low percentage	High percentage	Month	High

1.3 Problems

The problem	**Cause**	**Outcome**
Technology problems: hardware problems, software problems.		
There is no software or hardware that provides a complete solution to the problem of communication in a work environment in an open space while isolating the noise with headphones.	Complicated algorithm for realization.	Lack of communication between employees.

1.4 Corporate / Business Context

General organizational structure:
Irrelevant.

1.5. Annual Work Plan

Executive Meetings - Meetings will be held once every two weeks with the application specialists where progress will be shown in building the system.
At these meetings, it is possible to track desired progress and ensure proper understanding of system requirements.
In the meetings will be present the application specialists, and the users.
Equipment - A number of laptops.

1.6 Applicability and cost-effectiveness

1.6.1.System applicability

The system will work on Windows 7 and above.
With built-in or external microphone.

1.6.2 Benefit

- Improving communication in open work environments.
- Better quality work.

1.7 The timeline

Milestones - Executive meetings every two weeks.
End date - 1/06/2020.

Number	Task	Start Date	End Date	Duration	Prerequisites
1	project proposal	01/07/2019	01/08/2019	31	-
2	submit literature review format	07/11/2019	17/11/2019	10	1
3	literature review	17/11/2019	24/11/2019	7	2
4	market research	24/11/2019	01/12/2019	7	1
5	Initiation and specification files	01/12/2019	08/12/2019	7	1
6	software development	08/12/2019	23/04/2020	137	5
7	submit first draft	07/11/2019	15/12/2019	38	3,4,5
8	first follow committee meeting	22/12/2019	26/12/2019	4	7
9	final submission of initial report	15/12/2019	24/01/2020	40	8
10	second follow committee meeting	19/04/2020	23/04/2020	4	6,9
11	submit an abstract draft of the project	23/04/2020	10/05/2020	17	10
12	submit an approved final summary of the project	10/05/2020	25/05/2020	15	11
13	submit the project poster	10/05/2020	25/05/2020	15	12
14	final submission of approved poster	25/05/2020	01/06/2020	7	13
15	final submission of report	01/06/2020	19/06/2020	18	14
16	project conference	19/06/2020	19/06/2020	0	15

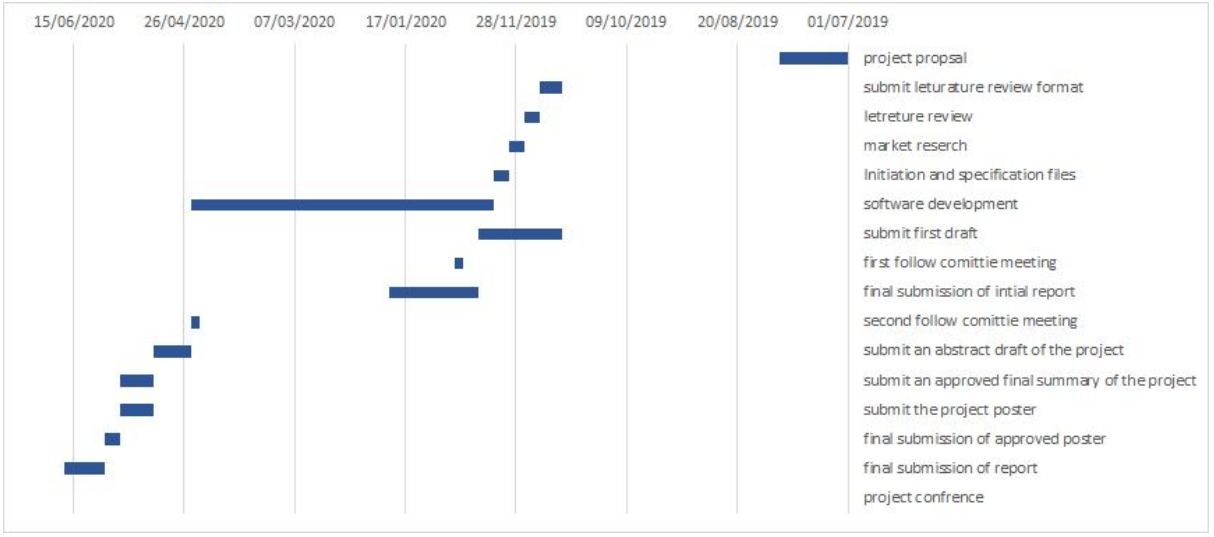


Figure 8
Gantt Chart

2. Application

2.1 Nature and general state of the application

2.1.1 The existing system

Today there is a partial solution to the problem that it is noise-canceling headphones but blocks the ability to hear the environment.

2.1.2 The nature of the new system

- The new system will work together with the use of noise-canceling headphones.
- The system will be built from scratch as per detailed requirements (listed in this document).
- The system will have a convenient user interface.
- The system will run in the background of the operating system.

2.2 External delimitation - Users and tangent systems

Details of system users and their classification:
Primary user - full control of the system.
- No additional users except the above.

2.3 User / Operational Interface

Listing potential user skills:
Basic knowledge of English.
No additional knowledge or additional work frames required.
The interface must be comfortable and clear.

2.4 Processes

List of main processes for the system:

Process	**Description**
opening the program	Clicking on the system icon will open the system.
Set the username	Clicking on settings and inserting the username in the appropriate line and clicking Save.
Add people to the repository ID	Clicking on settings and writing the person's information in the appropriate line and making recordings with the click of a button.
Setting the volume	Enter the setting volume meter and set to the desired volume.
Enable caller identification	Clicking the power button will start recording background noise and analyze whether the user is named.
Caller Identification	When the system detects the user name , it will analyze who the caller is.
Lowering the volume	After all identification processes, the system will lower the master volume.
Caller Alert	After lowering, the system will alert who called the user.

2.5 Glossary

Term	Explanation
json	type of file.
GUI	graphic user interface.
Audio signals	representation of sound.
Model	representation of a system using mathematical concepts and language.
Framework	a reusable set of libraries or classes for a software system or subsystem.
Dataset	a collection of data.

2.6 Reports (Queries)

Reports (queries) desired in the system:
User queries:
Displays the person's identification that called the user.
Displays user settings such as name and labels.

2.7 Data Security

Data security is minimal,the system is local so there is no issue with unwanted information leakage. Any security issue is in the responsibility of the user to keep their computer up to date with the latest security standards.

2.8 Load volumes and performance

Average number of users at the same time:
There is no case of system overload because there is only one user.
System performance depends on the user's computer hardware.

2.9 Interfaces and Links

Information relationships with external systems:
The system has links with the default speakers and microphone.
The system is local.

3.2 Characterization

1. Goals

1.1 Client / application specialist

1.1.1 Users

Software developers:
- Tal Gradus
Email: talgr3@ac.sce.ac.il .
- Olga Lapovsky
Email: olgala1@ac.sce.ac.il.
- Alex Skatkov
Email: alexsk@ac.sce.ac.il.
Software developers will be the system maintainers, responsible for the ongoing management of the software.

1.1.2 Application specialist

Natalia Vanetik Ph.D
Email: natalyav@sce.ac.il

Marina Litvak Ph.D
Email: marinal@sce.ac.il

- When the project begins, the application specialists will participate and accompany the characterization of the system.

1.2 Goals and Objectives

1.2.1 General Objectives

Long-term planning.
High detection percentages.
Fast response time.

1.2.2 Practical Objectives

Goals	**Objectives**
High detection percentages.	High detection percentages.
Fast response time.	Fast response time.
Long-term planning.	Work with a large database of people who can effectively call the user.

1.2.3 Future Goals

Objective 1: Convenient and clean user interface

current state	**Desirable condition**	**How long from the activation system**	**Priority**
Basic GUI	Clean and convenient	Week	Low

Objective 2: Quick speaker recognition

current state	**Desirable condition**	**How long from the activation system**	**Priority**
Slow	Fast	Month	High

Objective 3: High detection percentage

current state	Desirable condition	How long from the activation system	Priority
Low percentage	High percentage	Month	High

1.3 Problems

1.3.0 Summary of the problems in the existing situation

Waste of time.
Damage efficiency.
Lack of communication.

1.3.1 Problems that the system solves / is supposed to solve

The problem	Cause	Outcome
Technology problems: hardware problems, software problems.		
There is no software or hardware that provides a complete solution to the problem of communication in a work environment in an open space while isolating the noise with headphones.	Complicated algorithm for realization.	Lack of communication between employees

1.3.2 Problems that the system creates / may create

High reliance on the system which can create misunderstandings in case of inaccuracy.

1.3.3 Problems Rejected

The system can only work on Windows 7 and above.

2. Application - System essence

2.1 General Features

2.1.1 Existing condition

Today there is a partial solution to the problem that it is noise-canceling headphones but blocks the ability to hear the environment.

2.1.2 The nature and type of the system

- The new system will work together with the use of noise-canceling headphones.
- The system will be built from scratch as per detailed requirements (listed in this document).
- The system will have a convenient user interface.
- The system will run in the background of the operating system.

2.1.3 Constraints

Ethics: The information sitting on a local computer will not be passed on to a third party.
Technology: there isn't a 100% guarantee to detect a speaker.
Schedule: The system must be ready by 01.06.20.

2.1.4 Glossary

Term	**Explanation**
json	type of file.
GUI	graphic user interface.
Audio signals	representation of sound.
Model	representation of a system using mathematical concepts and language.
Framework	a reusable set of libraries or classes for a software system or subsystem.
Dataset	a collection of data.

2.2 External delimitation

2.2.0 General delimitation

This system only works on Windows 7 and above with only one user.

2.2.1 Users

Single primary user.

2.2.2 Tangent systems

Noise-canceling headphones,microphone.

2.3 Internal delimitation

2.3.0 General description of the system

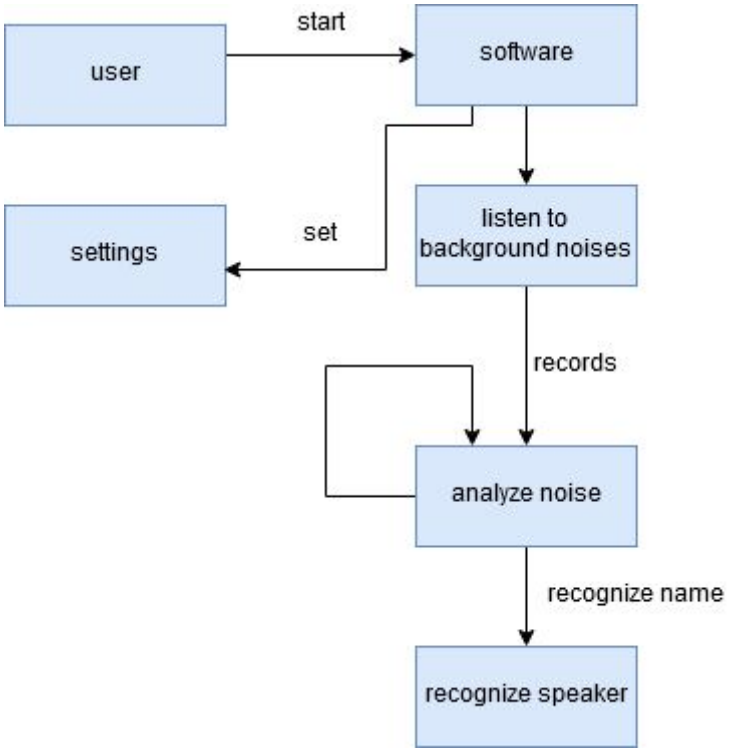


Figure 9
General description of the system

2.4 User interface

2.4.0 Human Engineering Rules

Irrelevant.

2.4.1 Menu Screens - The Screens Tree

Main page.
Settings.

2.4.2 Action Screens

Main page: start process.
Settings: set name,set volume,set labels.

2.5 Processes

Process	**Description**
Opening the program	Clicking on the system icon will open the system.
Set the username	Clicking on settings and inserting the username in the appropriate line and clicking Save.
Add people to the repository ID	Clicking on settings and writing the person's information in the appropriate line and making recordings with the click of a button.
Setting the volume down	Enter the setting volume meter and set the to the desired volume.
Enable caller identification	Clicking the power button will start recording background noise and analyze whether the user is named.
Caller Identification	When the system detects the user name , it will analyze who is the caller.
Lowering the volume	After all identification processes, the system will lower the computer volume.
Caller Alert	After lowering, the system will alert who called the user.

2.6 Physical Files - DATABASE

Local database.

2.7 Reports (Queries)

Reports (queries) desired in the system:
User queries:
Displays the person's identity that called the user.
Displays user settings such as name and labels.

2.8 Inputs (Forms)

Audio from the environment, Tags.

2.9 Data Security

Data security is minimal,the system is local so there is no issue with unwanted information leakage. Any security issue is in the responsibility of the user to keep their computer up to date with the latest security standards.

2.10 Crosses

Data Crossing: Recorded audio has been tested against previous samples.

2.11 Load volumes and performance

Number of workstations: 3 development positions.
Special Load periods: end of academic year.
Storage space: the system needs to be built to handle large databases. Up to 1000 details. There should be no problem of deviating from an assigned location in the databases.

2.12 Special Requirements

Identification times should be as short as possible.

2.13 Future Requirements

Development of the software in additional operating systems.

3. Technology and infrastructure

3.1 Central Hardware

The system will work locally on Windows 7 and above.

3.2 Centralized Data Storage

local database.

3.3 Edge Equipment

Computer with built-in or external microphone.

3.4 Development and Maintenance Tools

Pycharm-python framework.
Google drive-storing the software in google drive and constant updates.

4. Implementation

4.1 Work Plan

4.1.0 Development Method

The method of development and management of the project is serial.

4.1.1 General Development Plan

The design phase must be completed by 30.11.2019, the construction phase must be completed by 24.5.2020.

4.1.2 Individual Plan

Number	Task	Start Date	End Date	Duration	Prerequisites
1	project proposal	01/07/2019	01/08/2019	31	-
2	submit literature review format	07/11/2019	17/11/2019	10	1
3	literature review	17/11/2019	24/11/2019	7	2
4	market research	24/11/2019	01/12/2019	7	1
5	Initiation and specification files	01/12/2019	08/12/2019	7	1
6	software development	08/12/2019	23/04/2020	137	5
7	submit first draft	07/11/2019	15/12/2019	38	3,4,5
8	first follow committee meeting	22/12/2019	26/12/2019	4	7
9	final submission of initial report	15/12/2019	24/01/2020	40	8
10	second follow committee meeting	19/04/2020	23/04/2020	4	6,9
11	submit an abstract draft of the project	23/04/2020	10/05/2020	17	10
12	submit an approved final summary of the project	10/05/2020	25/05/2020	15	11
13	submit the project poster	10/05/2020	25/05/2020	15	12
14	final submission of approved poster	25/05/2020	01/06/2020	7	13
15	final submission of report	01/06/2020	19/06/2020	18	14
16	project conference	19/06/2020	19/06/2020	0	15

4.2 Next / Immediate Step

Design the system using Uml.

4.3 Current Operations

Project advisor.

4.4 Resilience and Reliability

4.4.1 Test Plan

The encompassing expected system tests are unit tests for each requirement and integration test.

4.4.2 Availability and survivability

The level of reliability and system resilience is exceptionally high.

4.5 Configurations

4.5.0 List of Configurations (Installations)

Computer Program.

4.5.1 Development Configuration (and Experiments)

Integrated development testing.

4.5.2 Main Configuration (Central, Main Server)

Local software.

5. Cost - Resources

5.1 Establishment cost (development and installation)

5.1.1 First Edition (Upcoming)

Depends on the number of requirements that will be fulfilled.

5.2 Current cost

5.2.1 First Edition (Upcoming)

Depends on the number of requirements that will be fulfilled.

4. Risk Management

Characterization of risk	**Risk analysis**	**Probability**	**Influence**	**Ways of treatment and coping**
Network realization	Inability to build networks due to lack of information in the field of speaker identification with a small amount of information.	high	The project will not be implemented.	Implementing a network that is less suited and making an effort not to let it damage the software quality.
Hardware problem	Hardware that is not compatible with the software such as a microphone or computer that fails to run the program.	medium	Will result in incorrect results or software crash.	Cheap hardware replacement.
Low success rates	The network will produce low detection results	high	Causes low quality software	Spend a lot of time training and experimenting to improve network quality.
File corruption issues	An external factor will cause the files to be corrupted.	low	Disruption of software processes.	Create automatic backups.

5. Requirements analysis

ID number	01	**Name of requirement**	Enable recording		
Date of request	27/11/2019	**System name**	The Third Ear		
Telephone	08-6475621	**Unit**	Software engineer	**Name of the requester**	Sami Shamoon College
Reference	**1001**	**End date required**	----------	**Urgency of execution**	Supreme

Analyst

Date	27/11/2019	**Role**	Requirements analyst	**Name**	Tal Gradus
Email	talgr3@ac.sce.ac.il	**Telephone**	054-7432637	**Organizational affiliation**	Development team

Application Analysis

Detailed description of how to apply	This occurs after logging in to the software when the user presses the record button.

Estimate development costs

	Initial evaluation	Final assessment
Implementation cost ($,NIS)	Irrelevant	Irrelevant
Schedule for implementation	Week	Week

ID number	02	**Name of requirement**	Define a keyword		
Date of request	27/11/2019	**System name**	The Third Ear		
Telephone	08-6475621	**Unit**	Software engineer	**Name of the requester**	Sami Shamoon College
Reference	**1002**	**End date required**	----------	**Urgency of execution**	Supreme

Analyst

Date	27/11/2019	**Role**	Requirements analyst	**Name**	olga lapovsky
Email	olgala1@ac.sce.ac.il	**Telephone**	054-7432637	**Organizational affiliation**	Development team

Application Analysis

Detailed description of how to apply	This occurs after logging in to the software when the user presses the settings tab and enters the required key word in the end he presses the save button to save.

Estimate development costs

	Initial evaluation	Final assessment
Implementation cost ($,NIS)	Irrelevant	Irrelevant
Schedule for implementation	Week	Week

ID number	03	**Name of requirement**	Define labels		
Date of request	27/11/2019	**System name**	The Third Ear		
Telephone	08-6475621	**Unit**	Software engineer	**Name of the requester**	Sami Shamoon College
Reference	**1003**	**End date required**	----------	**Urgency of execution**	Supreme

Analyst

Date	27/11/2019	**Role**	Requirements analyst	**Name**	alex skatkov
Email	alexsk@ac.sce.ac.il	**Telephone**	054-7432637	**Organizational affiliation**	Development team

Application Analysis

Detailed description of how to apply	This occurs after entering the software when the user presses the settings tab and enters the required label in the end he presses the save button to save.

Estimate development costs

	Initial evaluation	Final assessment
Implementation cost ($,NIS)	Irrelevant	Irrelevant
Schedule for implementation	Week	Week

ID number	04	**Name of requirement**	set volume		
Date of request	27/11/2019	**System name**	The Third Ear		
Telephone	08-6475621	**Unit**	Software engineer	**Name of the requester**	Sami Shamoon College
Reference	**1004**	**End date required**	----------	**Urgency of execution**	Supreme

Analyst

Date	27/11/2019	**Role**	Requirements analyst	**Name**	olga lapovsky
Email	olgala1@ac.sce.ac.il	**Telephone**	054-7432637	**Organizational affiliation**	Development team

Application Analysis

Detailed description of how to apply	This occurs after entering the software when the user presses the settings tab and sets the required volume.

Estimate development costs

	Initial evaluation	Final assessment
Implementation cost ($,NIS)	Irrelevant	Irrelevant
Schedule for implementation	Week	Week

ID number	05	**Name of requirement**	stop recording		
Date of request	27/11/2019	**System name**	The Third Ear		
Telephone	08-6475621	**Unit**	Software engineer	**Name of the requester**	Sami Shamoon College
Reference	**1005**	**End date required**	----------	**Urgency of execution**	Supreme

Analyst

Date	27/11/2019	**Role**	Requirements analyst	**Name**	Tal Gradus
Email	talgr3@ac.sce.ac.il	**Telephone**	054-7432637	**Organizational affiliation**	Development team

Application Analysis

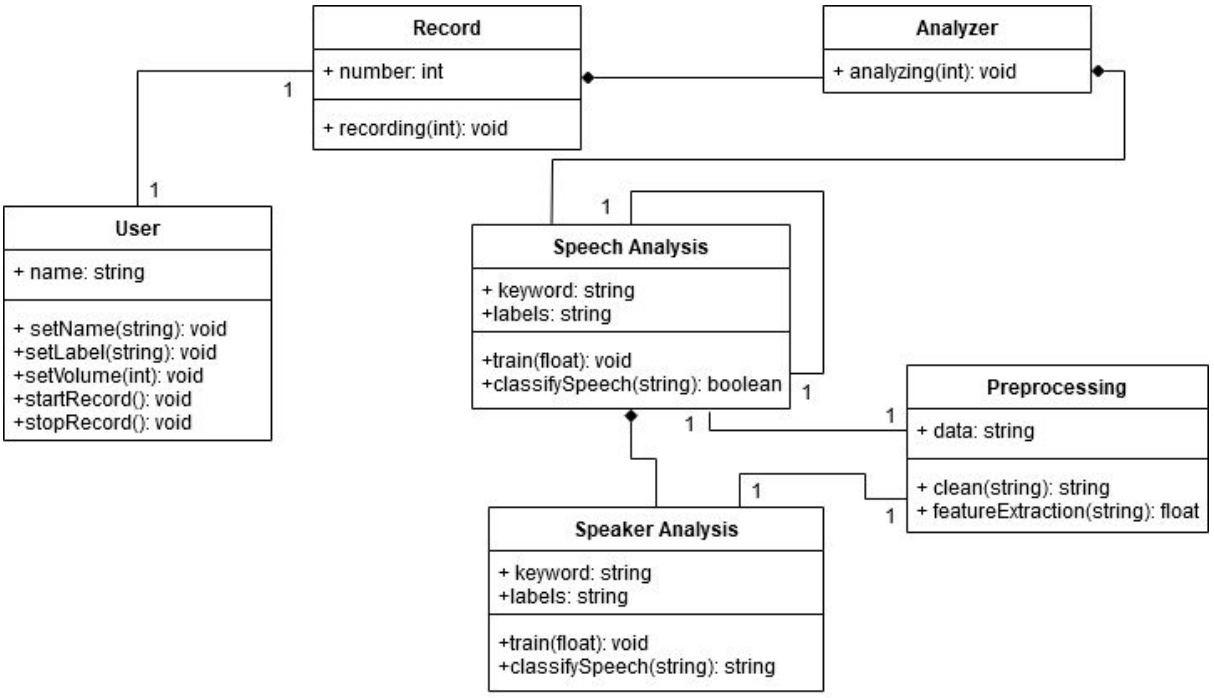
Detailed description of how to apply	This occurs after entering the software when the user presses the record button again to turn off..

Estimate development costs

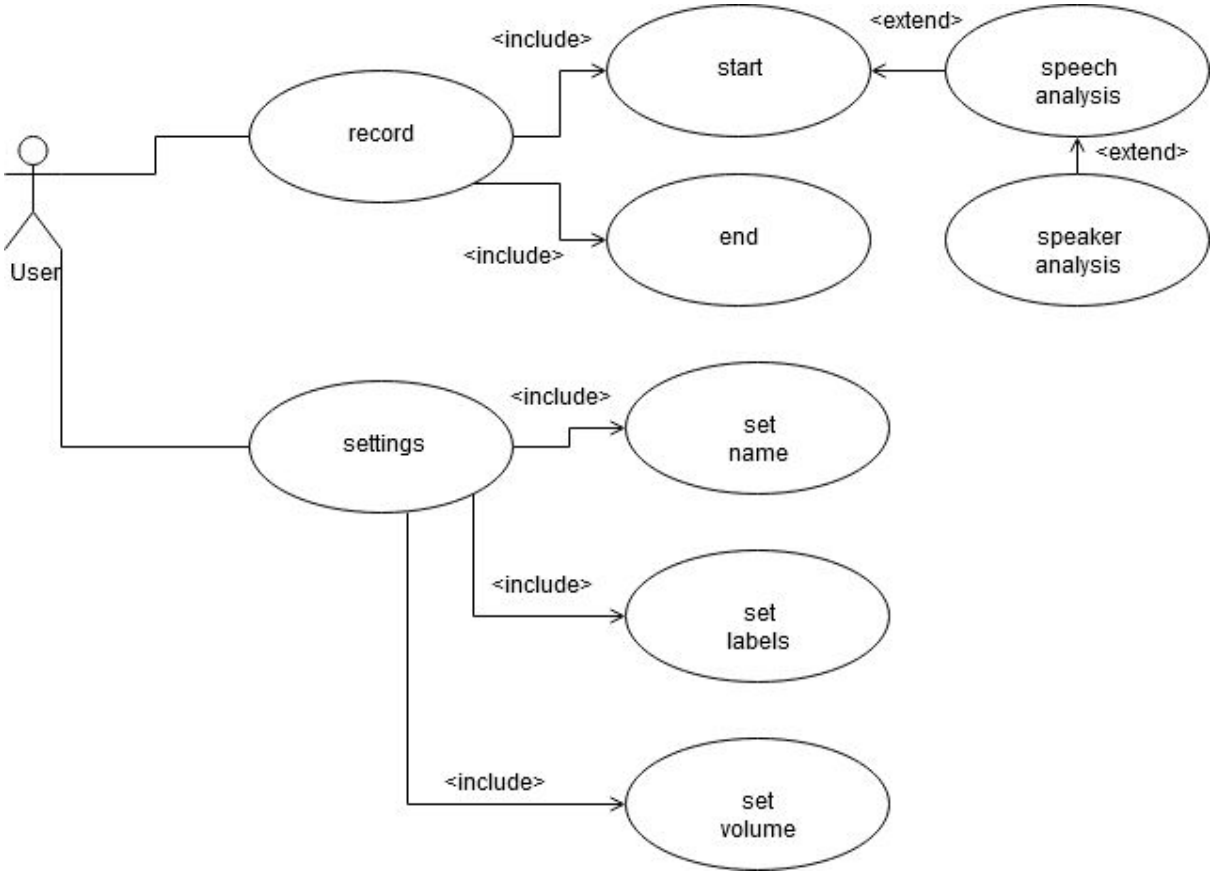
	Initial evaluation	Final assessment
Implementation cost ($,NIS)	Irrelevant	Irrelevant
Schedule for implementation	Week	Week

6. Design

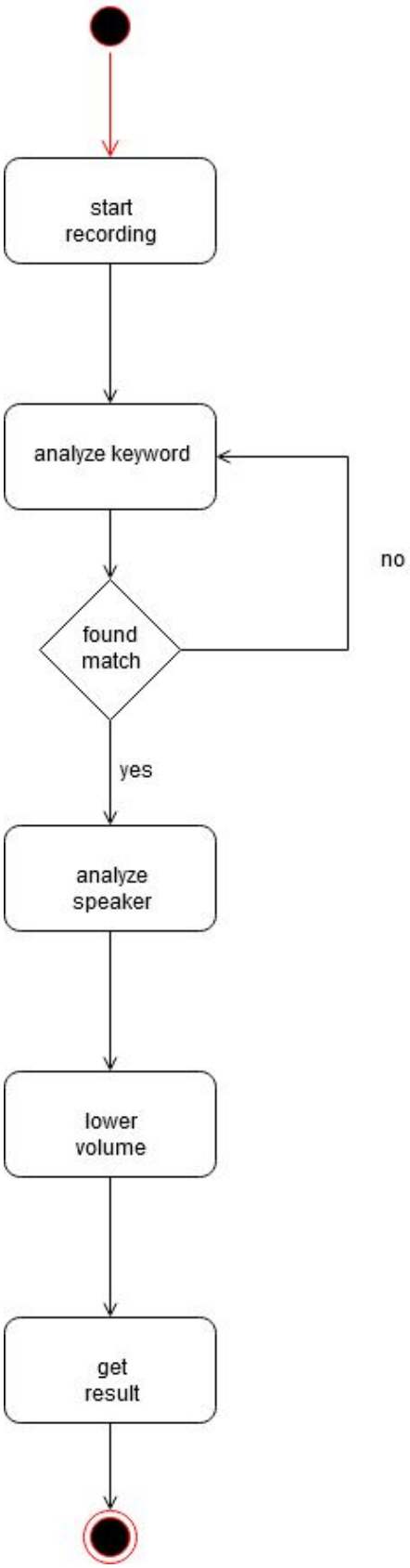
6.1 Class diagram

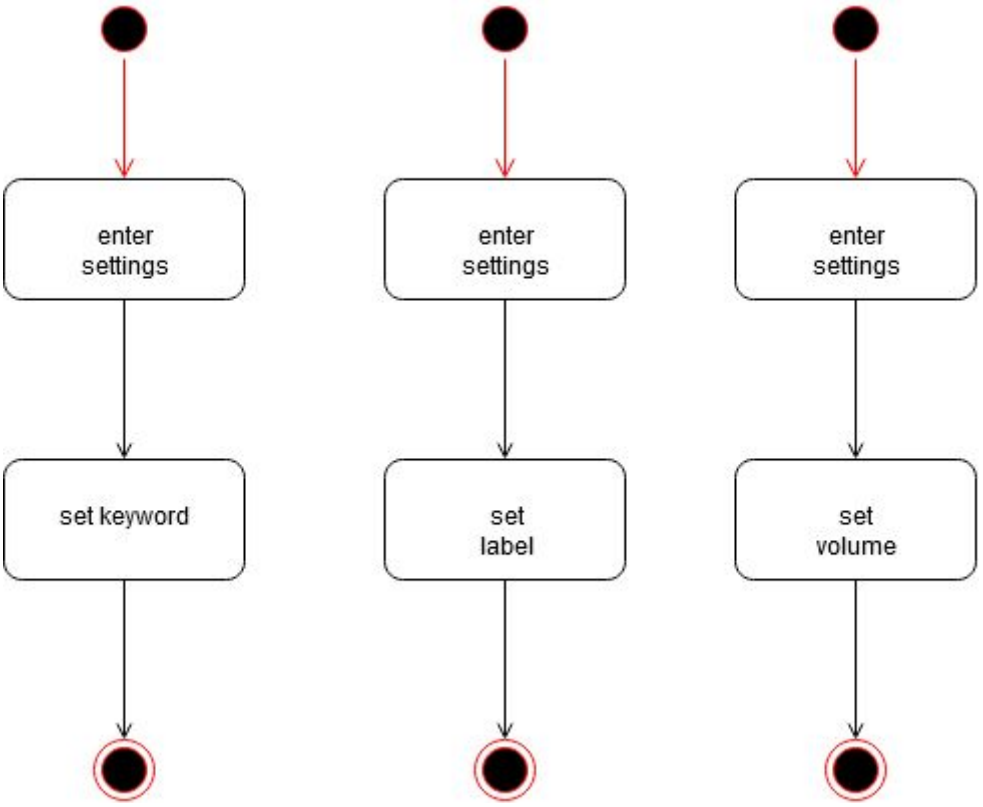


6.2 Use case


<include>
start
<extend>
speech analysis
record
<extend>
<include>
end
speaker analysis
User
<include>
set name
settings
<include>
set labels
<include>
set volume

6.3 Activity diagrams

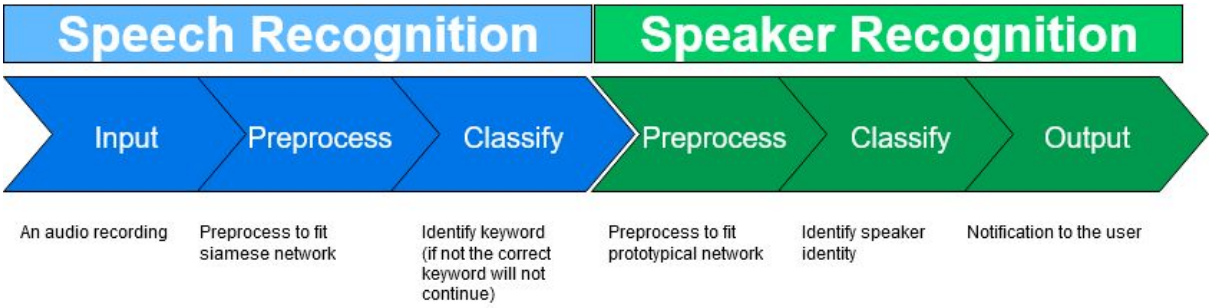



enter settings
set keyword
enter settings
set label
enter settings
set volume

7.Methodology

7.1 Preface

We want to address in a practical way the issue of noise and distractions in an open space environment, by using the power of two neural networks that can adapt and learn.
The first neural network is a siamese network for keyword detection see Section 2.1.3.4.1 for more details.
The second neural network is a prototypical network for speaker identification see section 2.1.3.4.2 for more details.
In this section we will explain on the dataset, preprocessing and training the model.



7.2 Data Collection

Our dataset consists of pre existing data and manually created data fit for both networks.
When we started to look for the pre existing data we wanted it to be audio files that includes different speakers with variations of different words. We found a couple of options for more details about the data see section 2.3 for more details.
We selected the TIMID dataset because it has the most variation in words and speakers, 630 speakers with 8 different dialects of american english.
In addition we manually created voice samples by recording various speakers, each speaker pronounced a specific keyword hadlist five times.

7.3 Preprocessing

The main point to understand about the speech is that the sounds generated by a human are manipulated by the shape of the vocal tract. This shape determines what sound comes out. by determining the shape accurately, we get an accurate representation of the *phoneme*[5] being produced. The shape of the vocal tract manifests itself in the waves of the short time power spectrum, and the job of MFCCs is to accurately represent this wave.

[5] a unit of sound that distinguishes one word from another in a particular language.

The first step in any audio processing systems is to extract features:

- Frame the signal into short frames.
- For each frame calculate the periodogram estimate of the *power spectrum*[6].
- Apply the mel *filterbank*[7] to the power spectrum, sum the energy in each filter.
- Take the logarithm of all filterbank energies.
- Take the *DCT(discrete cosine transform)*[8] of the log filterbank energies.
- Keep DCT(discrete cosine transform) coefficients 2-13, discard the rest.
- Combine *Deltas and Deltas-Deltas*[9] features to the feature vector

Each neural network requires different data representation:

- In the siamese network we take the data and catalog it to 3 different folders: “origin”,”good” and ”bad”. then we extract features for each category and create pairs from the folders such as (“origin”,”good”) is categorized as positive and (“origin”,”bad”) categorized as negative.
- In the prototypical network we catalog the audio files to classes and extract features to each one and generate the mean embeddings for the classes.

[6] describes the distribution of power into frequency components composing that signal.
[7] an array of band-pass filters that separates the input signal into multiple components, each one carrying a single frequency sub-band of the original signal.
[8] expresses a finite sequence of data points in terms of a sum of cosine functions oscillating at different frequencies.
[9] approximate first and second derivatives of the signal

7.4 Training

We divided our data as such: 70% training,25% testing and 5% validation
Each neural network requires different training:

- siamese network

  We define our base network with three dense layers and we feed the audio pair to the base network, then feed these feature vectors to the energy function to compute the distance between them, and we use Euclidean distance as our energy function. We define our loss function as Adam optimizer and train our model.

- prototypical network

  We created a query set and generated the embeddings and defined a distance function that gives us the distance between the class prototypes and query set embeddings. Then we calculated the distance between the class prototype and query set embeddings, to get the probability for each class as a softmax to the distance then, we compute the loss and calculate the accuracy.
  followed using the Adam optimizer for minimizing the loss and starting our TensorFlow session with episodic training, for each episode we sample data points, build the support and query sets, and train the model.

8.SRS

8.1 Introduction

8.1.1 Purpose

This SRS describes the software functional and nonfunctional requirements for TheThirdEar 1.0.
TheThirdEar 1.0 will permit users to use headphones and listen to music while the software will monitor if they have been called by a specifically defined keyword and if so it will notify the user and specify by whom.

8.1.2 Document Conventions

No document conventions are being used at this time.

8.1.3 Intended Audience and Reading Suggestions

This document is intended to be used by members of the project team that will implement the system and the academic advisers that will verify the correct functioning of the system.

8.1.4 Product Scope

TheThirdEar 1.0 is a windows application that will allow users to use headphones while being aware of the environment, with an easy-to-use interface.
The project scope can be found in section 3.

8.1.5 References

No references have been identified.

8.2 Overall Description

8.2.1 Product Perspective

TheThirdEar 1.0 is a new system that interacts with existing hardware.
Figure 1 illustrates the external entities and system interfaces.

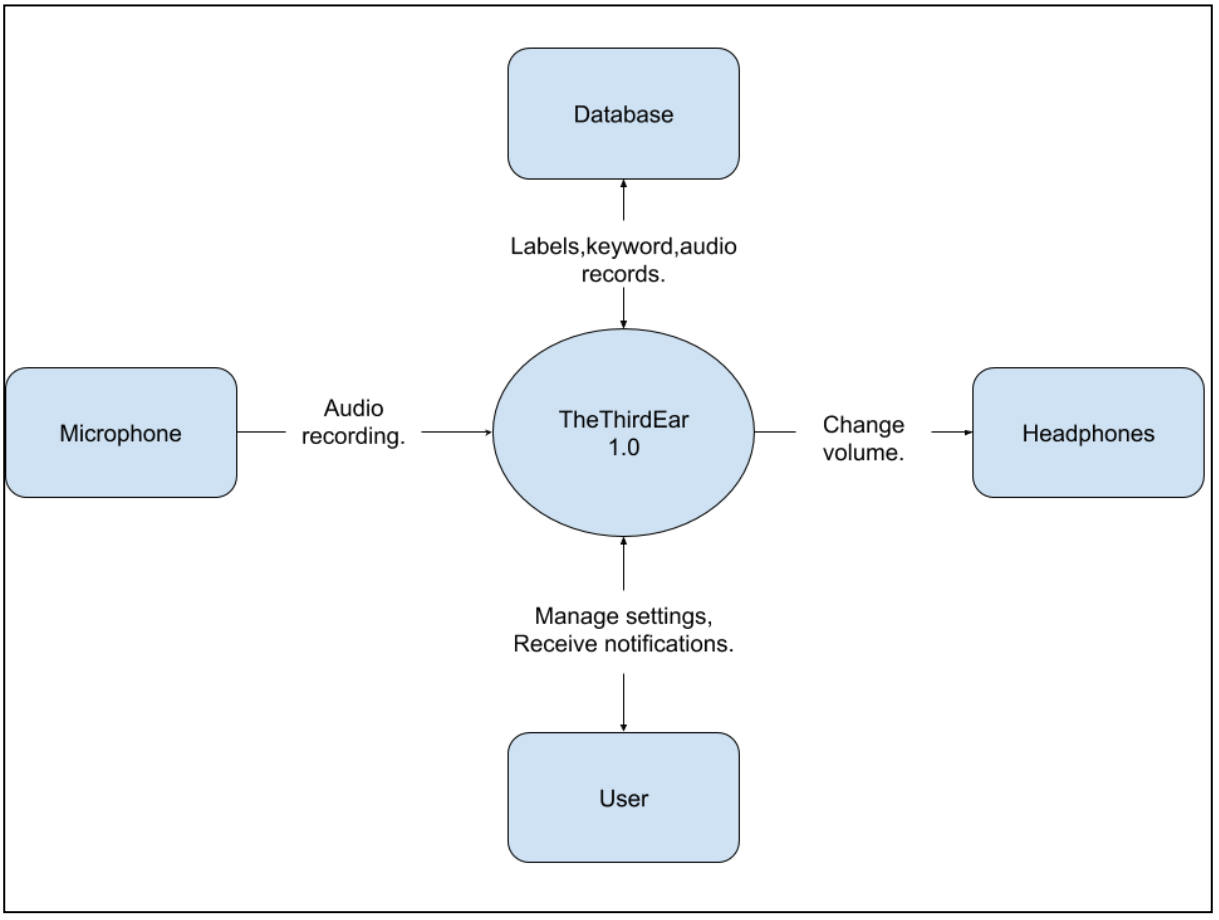


Figure 10
Context diagram for TheThirdEar 1.0.

8.2.2 Product Functions

FE-1: Customize keyword and labels
FE-2: Active listening
FE-3: Keyword recognition
FE-4: Speaker recognition
FE-5: Volume control
FE-6: User notification

See Figure 11 for the relationship between these features.

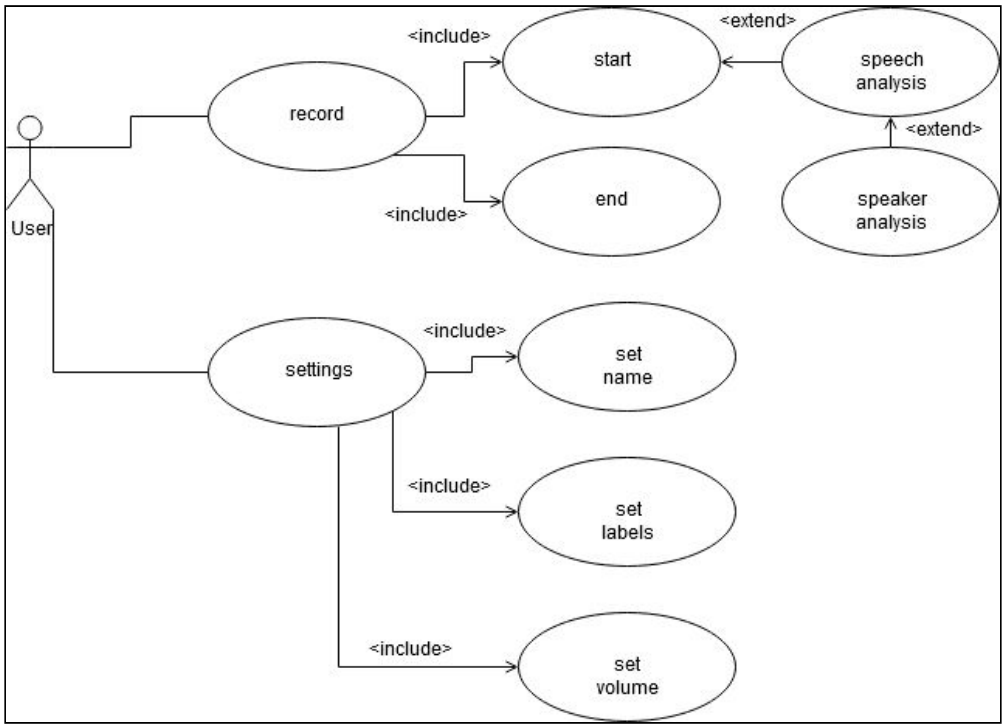


Figure 11
Major features and how they relate.

8.2.3 User Classes and Characteristics

User
The User is the only person who can access the system in the local computer that the system is installed upon.
The User has access to all the functions and settings.

8.2.4 Operating Environment

OE-1: The system shall operate in the newest versions of windows OS.
OE-2: There should be no constraint on users being able to access the system at a given time.
OE-3: Data is generated by a prior dataset and future data will be generated by microphone recordings and stored in a local database.

OE-4: All data will be stored in a local database with security measures

8.2.5 Design and Implementation Constraints

CO-1: All data shall be stored on a local database.
CO-2: Due to small amount of data the developers have a limited choice of models.
CO-3: Number of recordings will be limited due to memory restrictions.

8.2.6 User Documentation

No user documentation information at this time.

8.2.6 Assumptions and Dependencies

AS-1: No more than 1 GB of data stored on local database.
AS-2: Users use headphones while using the system.
AS-3: Microphone is able to catch high quality audio.
DE-1: Computer is open to installing external software.
DE-2: Users have a connected microphone.

8.3. External Interface Requirements

8.3.1 User Interfaces

UI-1: The software shall permit complete navigation through a standard computer mouse and keyboard.

8.3.2 Hardware Interfaces

HI-1: The software will receive input from a connected microphone, which records the surround audio.

8.3.3 Software Interfaces

SI-1: The System
SI-1.1: The system shall receive audio input and will analyze it.
SI-1.2: Upon keyword recognition, the system will analyze the audio fingerprint
SI-1.3: On recognition,the system will lower the volume and notify the User.

SI-2: Database - The system shall communicate with a database through a programmatic interface for the following operations:
SI-2.1: To store new audio recordings.
SI-2.2: To store user settings.
SI-2.3: To take the audio samples and to use for training the model.
SI-2.4: To take the audio samples for comparison.

8.3.4 Communications Interfaces

CI-1: The system shall send a notification to the user to inform them of keyword detection.

8.4. System Features

8.4.1 Customize keyword and labels

7.4.1.1 Description and Priority

The user will be required to enter a keyword of his choice and labels of people for recognition.
Priority: high.

7.4.1.2 Stimulus/Response Sequences

Stimulus: User sets a keyword.
Response: The speech recognition model starts the training process.

Stimulus: User sets labels.
Response: The speaker recognition model starts the training process.

8.4.1.3 Functional Requirements

Set a keyword	The system will allow the user to define a keyword.
Set labels	The system will allow the user to define labels .
Edit the keyword	The system will allow the user to edit the keyword.
Edit labels	The system will allow the user to edit the labels .

8.4.2 Active listening

7.4.2.1 Description and Priority

The software will be able to connect to an internal or external microphone and will listen and record the environment.
Priority: high.

7.4.2.2 Stimulus/Response Sequences

Stimulus: The user activates the active listening feature.
Response: The system will listen and record the environment.

Stimulus: The user deactivates the active listening feature.
Response: The system will stop listening and recording.

8.4.2.3 Functional Requirements

Turn on the active listening feature	The system will listen and record the environment.
Turn off the active listening feature	The system will stop listening and recording.

8.4.3 Keyword recognition

7.4.3.1 Description and Priority
The software will analyze the audio samples and will attempt to recognize if the defined keyword was spoken.
Priority: high.

7.4.3.2 Stimulus/Response Sequences
Not applicable.

7.4.3.3 Functional Requirements
Not applicable.

8.4.4 Speaker recognition

7.4.4.1 Description and Priority
The software will analyze the audio samples and will attempt to recognize the speaker identity.
Priority: high.

7.4.4.2 Stimulus/Response Sequences
Not applicable.

7.4.4.3 Functional Requirements
Not applicable.

8.4.5 Volume control

7.4.5.1 Description and Priority
The user can set the desired volume that the software will lower to it when there's a notification.
Priority: medium.

7.4.5.2 Stimulus/Response Sequences
Stimulus: The user sets the desired volume.
Response: The volume is set for future use of the system.

8.4.5.3 Functional Requirements

Set volume	The user can set the desired volume for future use of the system.

8.4.6 User notification

7.4.6.1 Description and Priority
The software will send a notification to the user when it recognizes a speaker identity who spoke the keyword.
Priority: high.

7.4.6.2 Stimulus/Response Sequences
Not applicable.

7.4.6.3 Functional Requirements
Not applicable.

8.5 Other Nonfunctional Requirements

8.5.1 Performance Requirements

PE-1: All keyword detection and speaker identification must be less than 10 seconds.
PE-2: The system threads will not overload the computer.

8.5.2 Safety Requirements

No safety requirements have been identified.

8.5.3 Security Requirements

SE-1: Users shall be required to log in to the system for all operations.

8.5.4 Software Quality Attributes

SQA-1: The software will be able to adapt to changing keyword and labels
SQA-2: The software will have high identification percentages

8.5.5 Business Rules

No business rules have been identified.

8.6 Other Requirements

No other requirements have been identified.

8.7 Appendix A: Glossary

7.7.1 Labels: labels are the names of the people which the model will try to identify.

8.8 Appendix B: Analysis Models

See section 6 in this document.

8.9 Appendix C: To Be Determined List

No issues have been identified.

9. Overview

In our project we implemented:

Preprocessing:

In the beginning we attempted to implement the same preprocessing strategy for both neural networks, we extracted the mfccs of each of our audio samples and calculated the mean embeddings of each sample in the form of a numpy array and with this we trained our networks. Due to bad predictions of our networks we realized each network needs a different strategy of preprocessing.

For the word recognition network we kept the strategy the same for the preprocessing but instead decided to expand each sample by manipulating the audio samples by adding noise layers,rotating,slicing,speeding and slowing down.

For the speaker recognition network we decided to take the audio samples and convert each sample spectrogram to an image. We performed image augmentation such as random rotation, shifts, shear and flips which allowed us to train our model with fewer images.

The following changes of the preprocessing improved both of the neural networks.

Word recognition:

In the beginning we attempted to implement a siamese network for recognizing a word for a small amount of data.
We managed to implement the siamese network with 60% accuracy and then improved 80% accuracy but even with a high accuracy the siamese network didn't produce a reliable result. While testing the network we started with a small amount of data but as we expanded, we realised that the siamese network is only suited for small amounts of data and we needed more data to reach our desired result, which was still considered small.
Thus we decided to attempt to build a CNN network with three Convolution layers and one Dense layer, that can handle a bigger amount of data in order to compare between the two. We ended up choosing the CNN network due to better accuracy of 93% and results.

Speaker recognition:

In the beginning we wanted to implement a prototypical neural network but after a few unsuccessful attempts of trying to implement it and experiments on the word recognition model we decided to build a cnn network with three Convolution layers and two Dense layers.
We successfully built the cnn network with accuracy of 90% with reliable results.

10. Experiments

10.1. Dataset

We have two types of datasets:

Word recognition data:
source: The data is a combination of recordings that has been generated by text to speech program in different accents and manual recordings.
Labels: In total there are 41 labels, 40 of them are people's names and the last one is a noise label which helps us prevent false predictions.
Description: The samples are in wav format in average length of 1 second.
There are 13 women samples and 4 men samples for each label.
Number of samples:
The number of the original samples is 680, for each label we expend the data and add 391 samples thus each label has 408 samples.
In addition we added a noise label that has 2000 samples.
The total amount of samples is 18320.

Speaker recognition data:

source: The data is a collection of manual recordings made by 2 women and 2 men and TIMID database with mixed genders.
Labels: In total there are 5 labels, 4 of them are people's names and the last one is an unidentified label for people that are not in the system.
Description: The samples are in PNG format with an average size of 10.5 KB.
Number of samples:
For each label there are 15 samples, in total there are 75 samples.

10.2. Experiments

We have two types of experiments:

Word recognition experiments:

We run experiments on different amount of labels:

- 21 labels with different epochs.
- 41 labels with different epochs.

Speaker recognition experiments:
We run the experiments on 5 labels:

- same words and amount of samples for each person with different epochs.
- different words and same amount of samples for each person with different epochs.
- same words and different amount of samples for each person with different epochs.
- different words and different amounts of samples for each person with different epochs.

The experiments are detailed in section 11.3.

10.3. Results

Word recognition:

20 users and noise:

Number of epochs	**Precision**	**Recall**	**F1-score**	**Accuracy**	**MSE**
400	0.96	0.96	0.96	95.66929340362549	0.004707114305347204
600	0.96	0.96	0.96	96.0629940032959	0.0032452207524329424
800	0.98	0.98	0.98	97.63779640197754	0.0022526760585606098
1000	0.98	0.98	0.98	97.83464670	0.002055486

				181274	6641759872

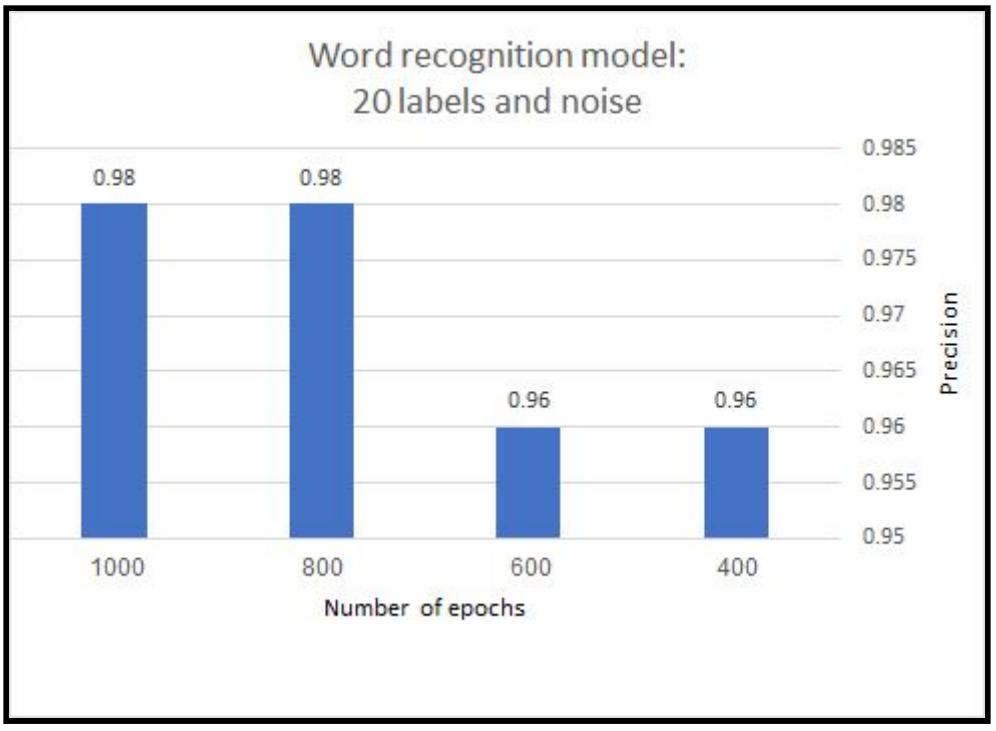


Figure 12
Precision in word recognition model with 21 labels.

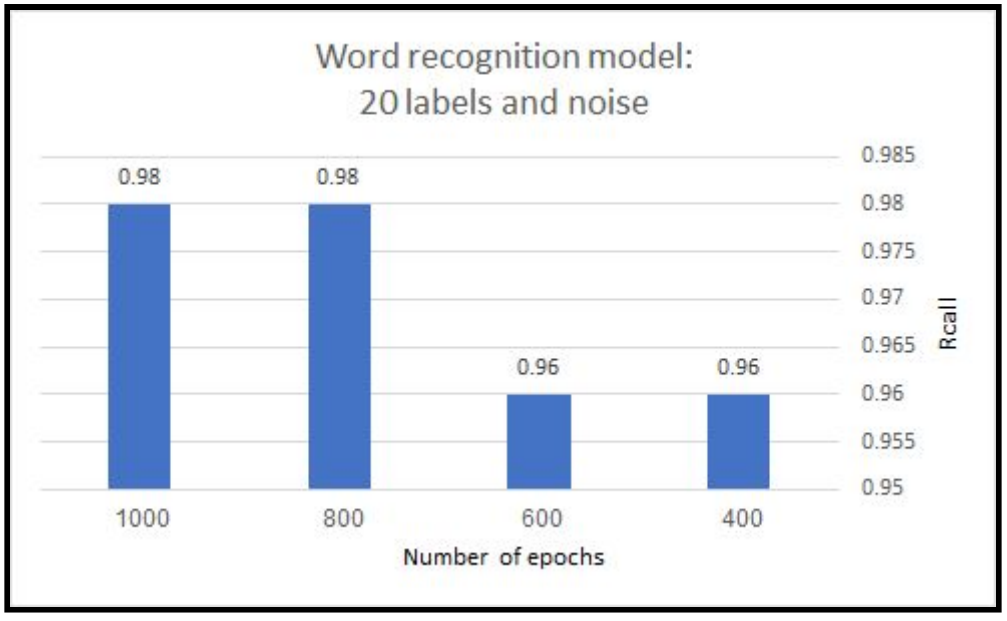


Figure 13
Recall in word recognition model with 21 labels.

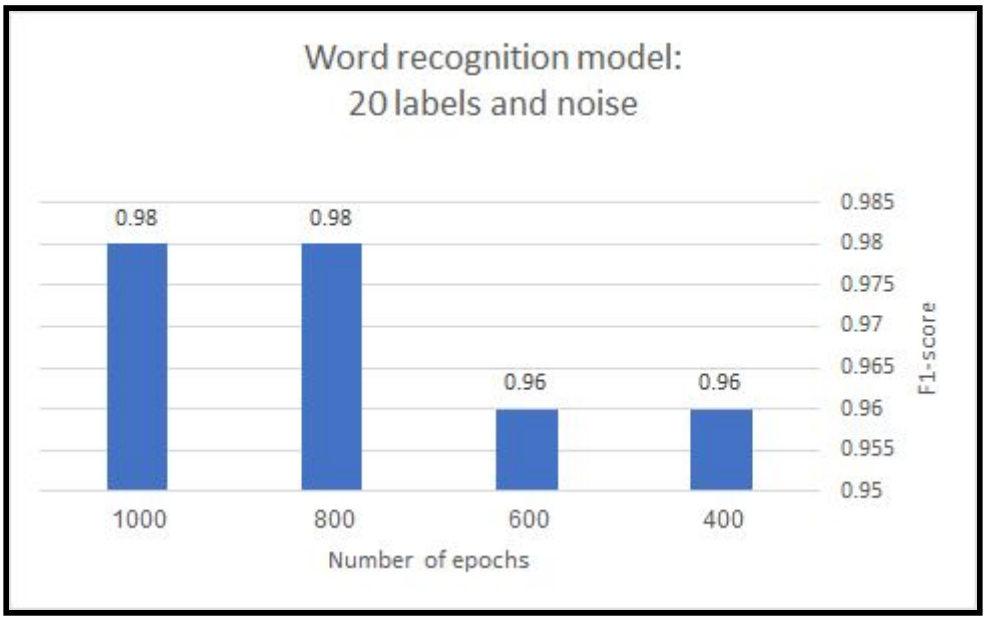


Figure 14
F1-score in word recognition model with 21 labels.

Figures 12,13 and 14 are a comparison between 400,600,800 and 1000 epochs with precision,recall and F1-score.
From the figures we can see that the precision,recall and F1-score are identical.
Looking at the epochs there is no change between epochs 400 and 600 but after we raise to 800 epochs there is a jump of 2% and then it continues to stay the same.

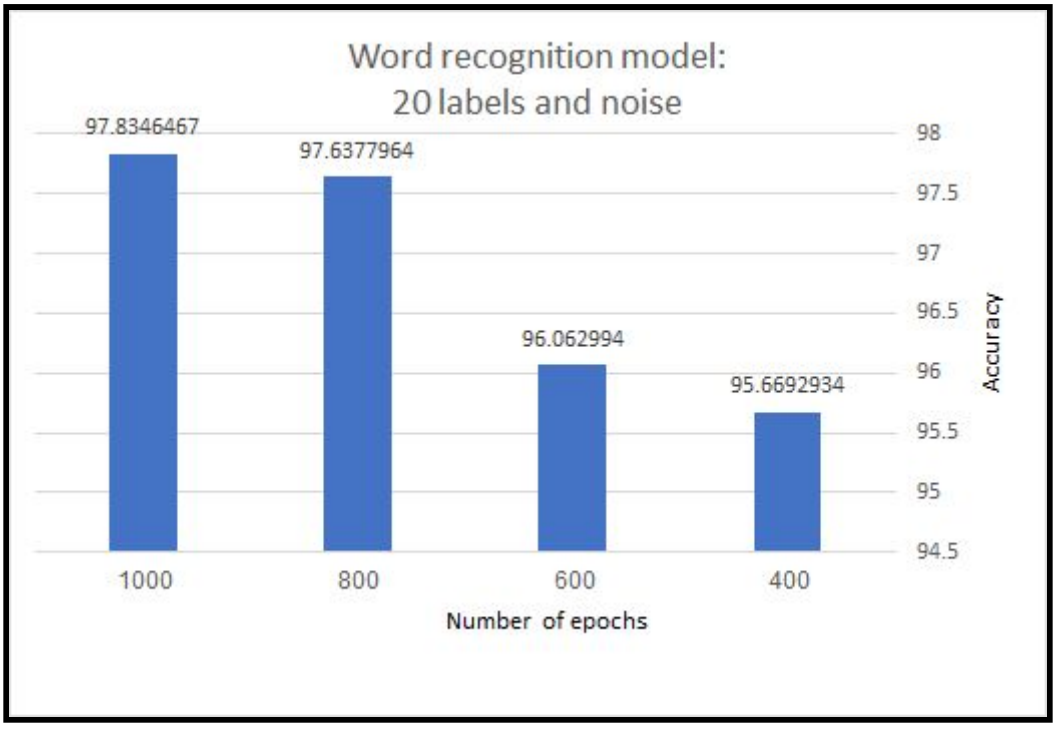


Figure 15
Accuracy in word recognition model with 21 labels.

Figure 15 is a comparison between 400,600,800 and 1000 epochs with accuracy.
The figure shows an increase of accuracy as we add more epochs, but the incline is starting to be negligible from the 800th epoch.

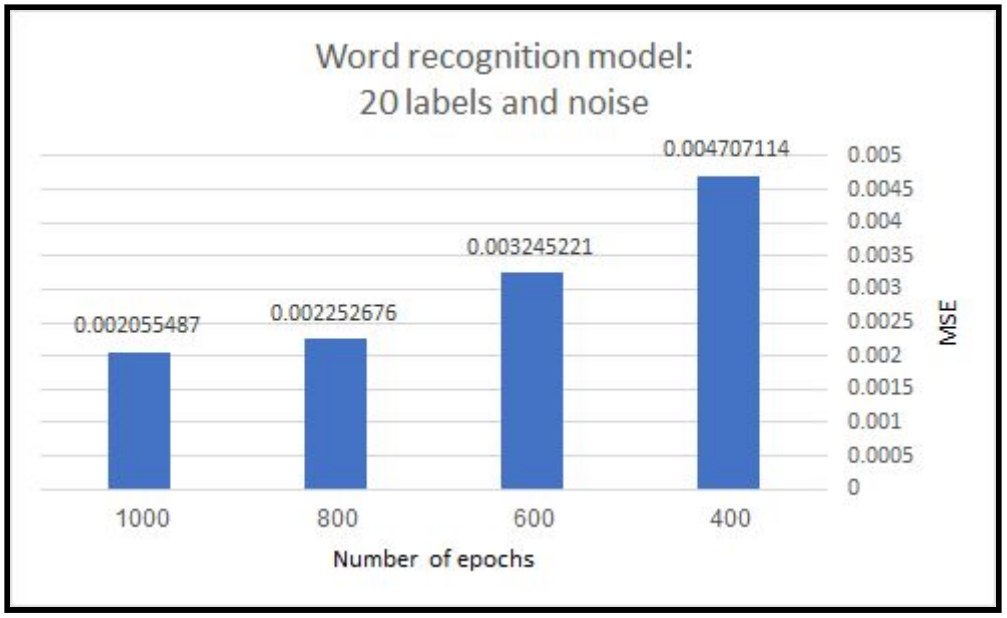


Figure 16
MSE in word recognition model with 21 labels.

Figure 16 is a comparison between 400,600,800 and 1000 epochs with MSE.
We can see in figure 16 a decrease in MSE as we increase the number of epochs but the decrease between 800 and 1000 epochs is negligible.

40 users and noise:

Number of epochs	Precision	Recall	F1-score	Accuracy	MSE
400	0.90	0.90	0.90	89.51964974403381	0.0049379714764654636
600	0.93	0.93	0.93	92.90392994880676	0.0042725731618702 41
800	0.92	0.92	0.92	92.35807657	0.0039006180595606565

				241821	
1000	0.94	0.94	0.94	93.2314395904541	0.0034409204963594675

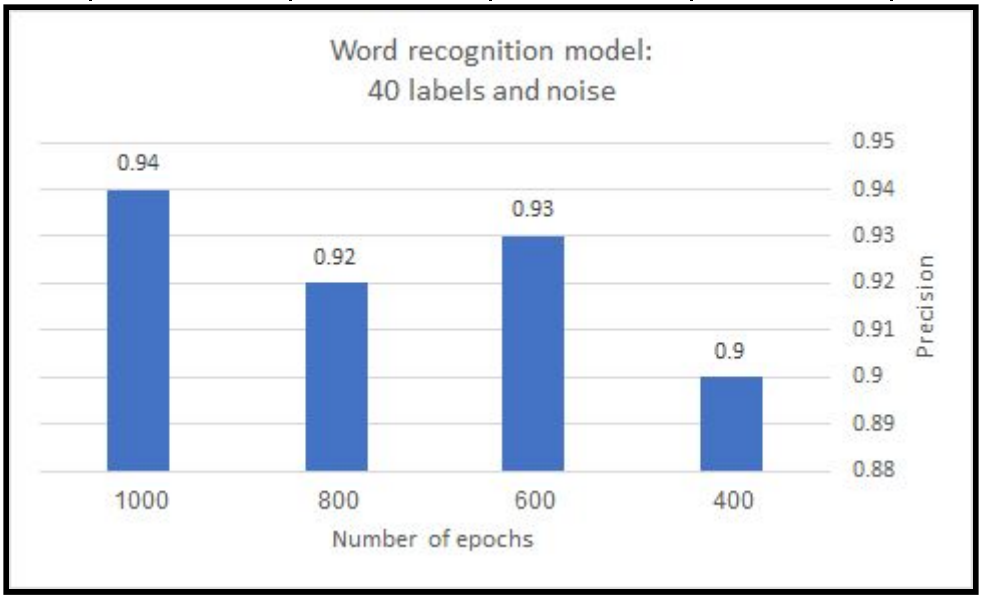


Figure 17
Precision in word recognition model with 41 labels.

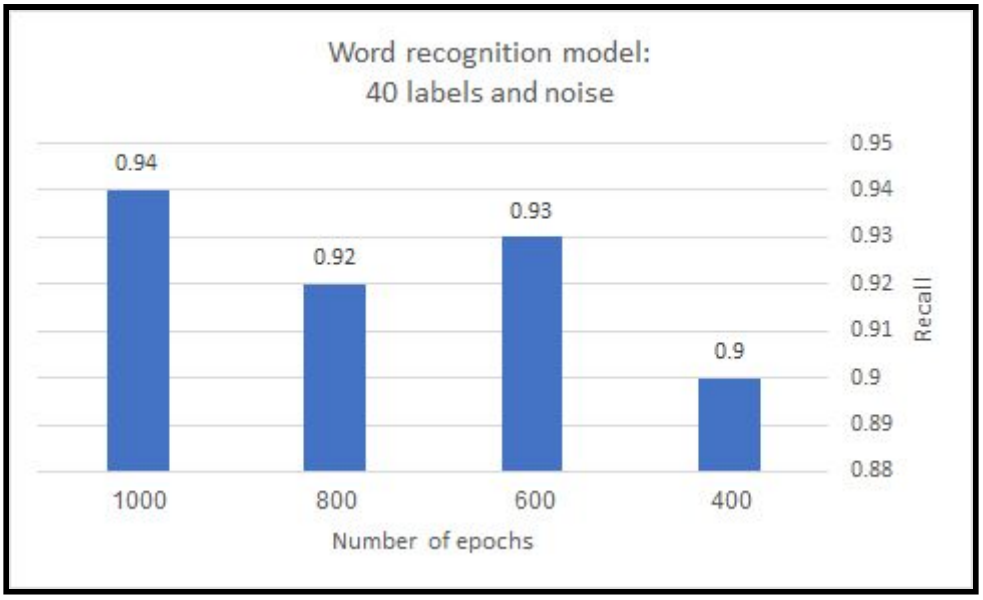


Figure 18
Recall in word recognition model with 41 labels.

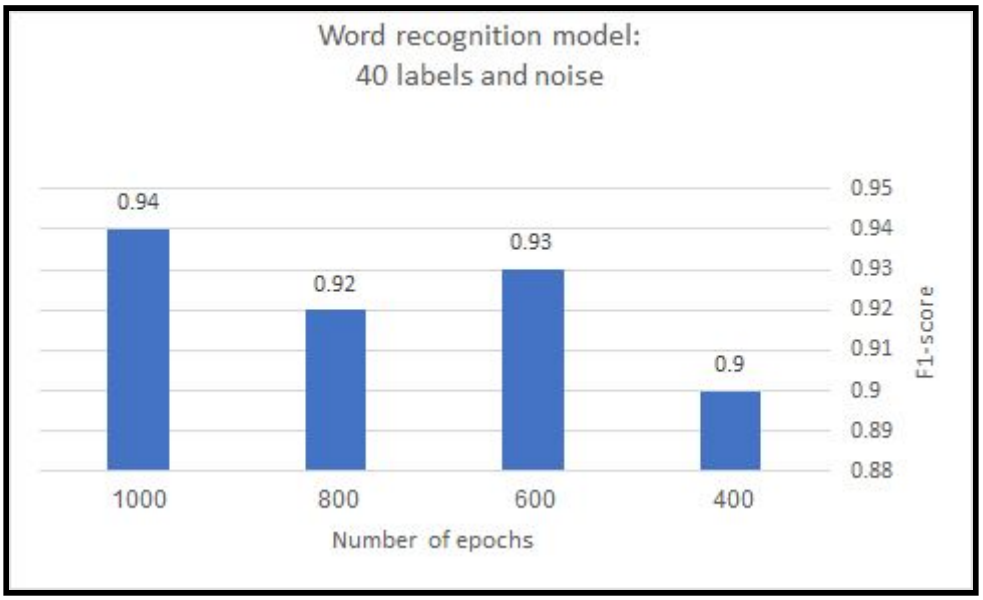


Figure 19
F1-score in word recognition model with 41 labels.

Figures 17,18 and 19 are a comparison between 400,600,800 and 1000 epochs with precision,recall and F1-score.
From the figures we can see that the precision,recall and F1-score are identical.
Unlike the model with 21 labels there isn't a constant increase, instead we can see a slight rise and fall between the epochs.

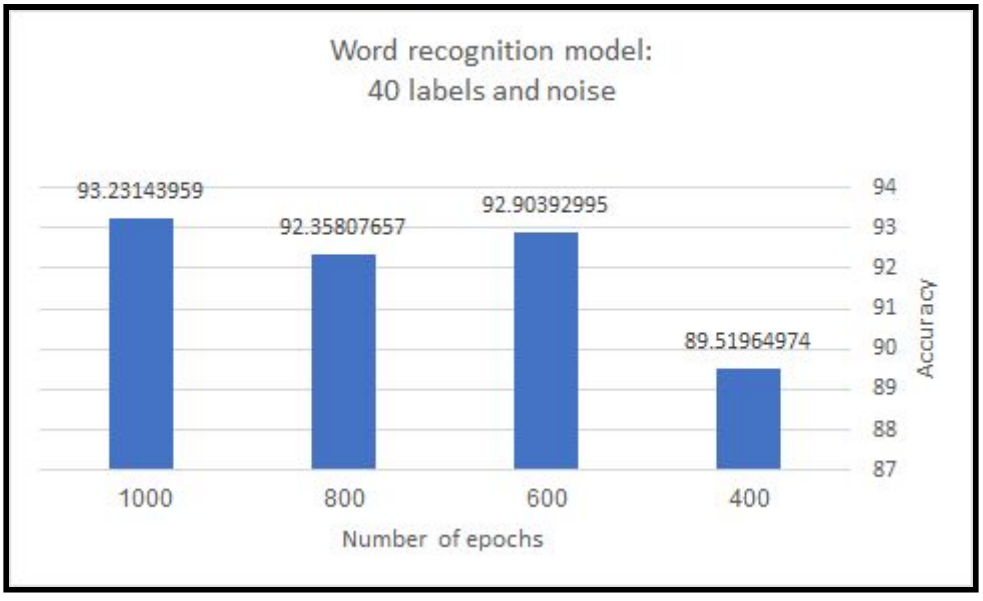


Figure 20
Accuracy in word recognition model with 41 labels.

Figure 20 is a comparison between 400,600,800 and 1000 epochs with accuracy.

The figure shows an increase up until the 600th epoch but from the 600th epoch to the 1000th epoch the change is negligible.

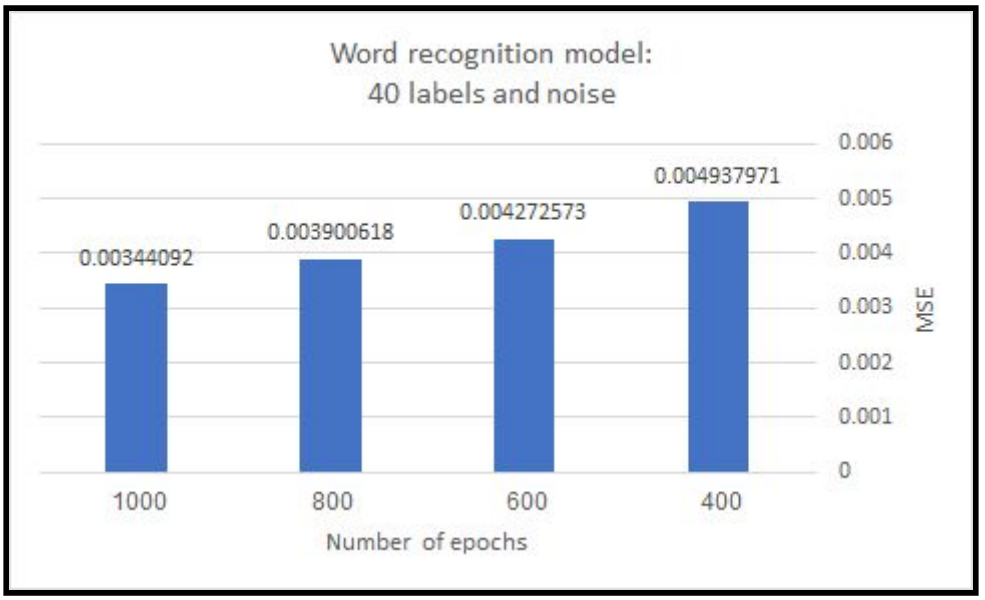


Figure 21
MSE in word recognition model with 41 labels.

Figure 21 is a comparison between 400,600,800 and 1000 epochs with mse.
We can see in figure 21 a decrease in mse as we increase the number of epochs but the decrease between 800 and 1000 epochs is negligible.

Conclusion:

After many experiments we realized that as the number of labels increases the percentage of the evaluation methods decreases.
In our case the model with 21 labels the results were high but we experienced overfitting, yet the model with 41 labels was not overfitted and still had high results.

Speaker recognition:

5 identical voice samples:

Number of epochs	Precision	Recall	F1-score	Accuracy	MSE
30	0.600000	0.600000	0.600000	60.00000238418579	0.12237970530986786
50	0.600000	0.600000	0.600000	60.00000238418579	0.11284387111663818
70	0.600000	0.600000	0.600000	60.00000238418579	0.14927792549133 3

5 different voice samples:

Number of epochs	Precision	Recall	F1-score	Accuracy	MSE
30	0.700000	0.700000	0.700000	69.9999988079071	0.11582915484905243
50	0.700000	0.700000	0.700000	69.9999988079071	0.11535743623971939
70	0.700000	0.700000	0.700000	69.9999988079071	0.09580546617507935

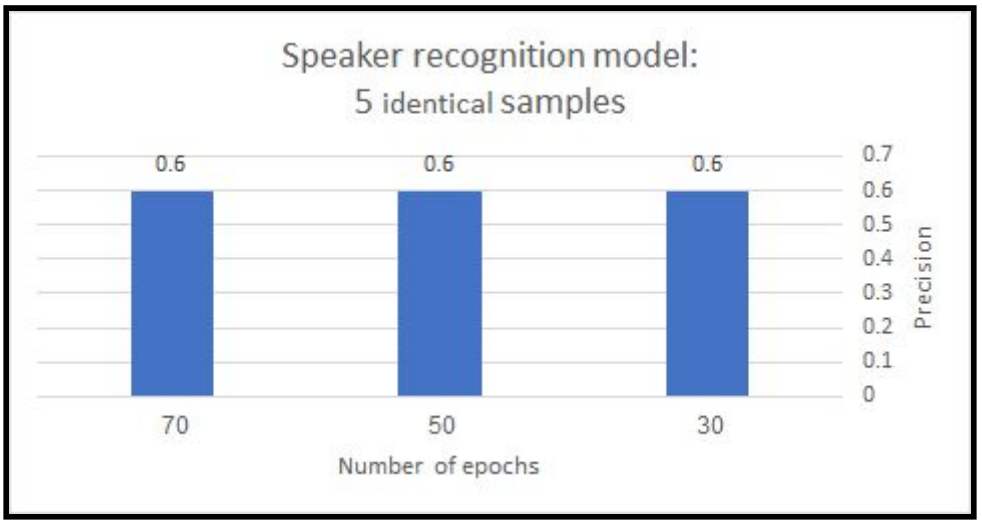


Figure 22
Precision in speaker recognition model with 5 same samples.

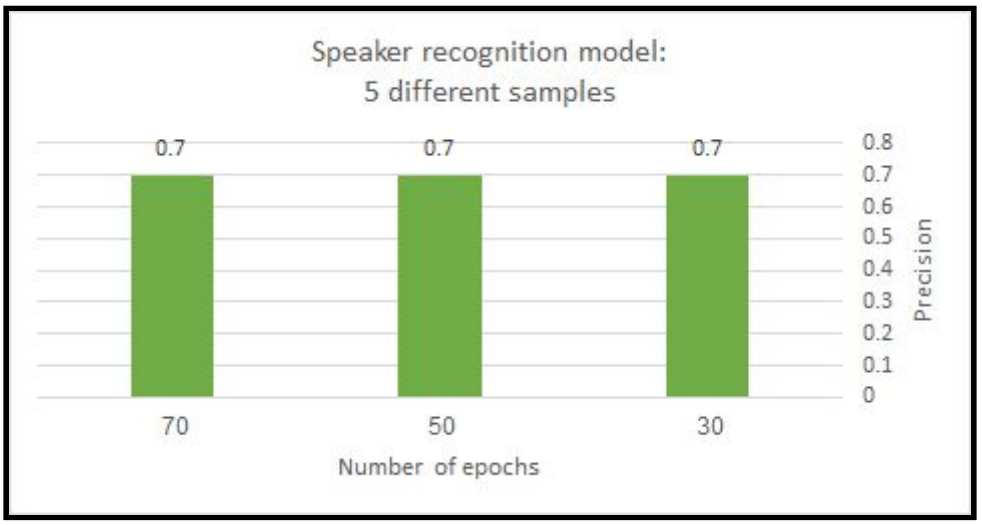


Figure 23
Precision in speaker recognition model with 5 different samples.

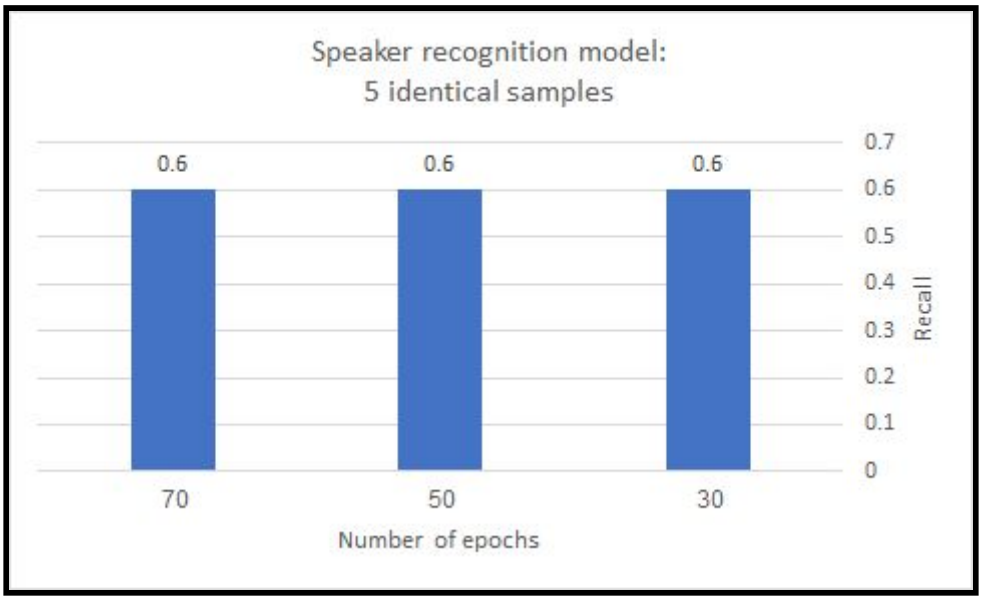


Figure 24
Recall in speaker recognition model with 5 same samples.

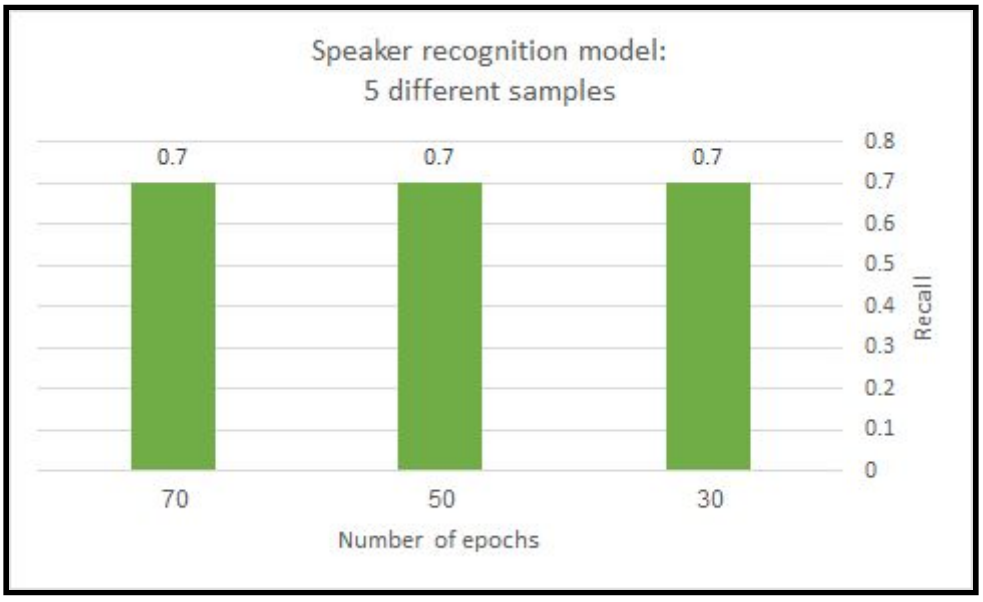


Figure 25
Recall in speaker recognition model with 5 different samples.

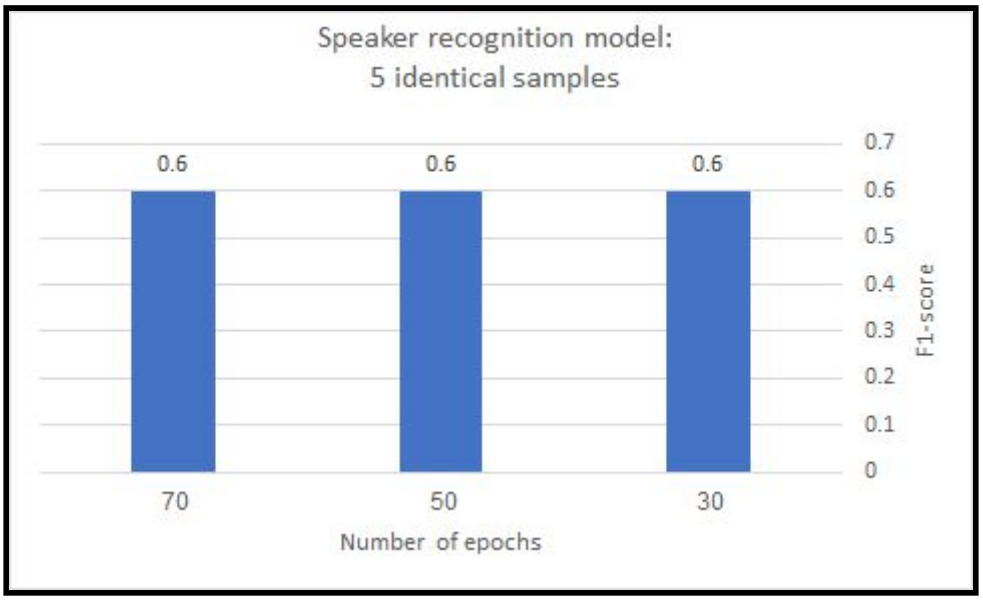


Figure 26
F1-score in speaker recognition model with 5 same samples.

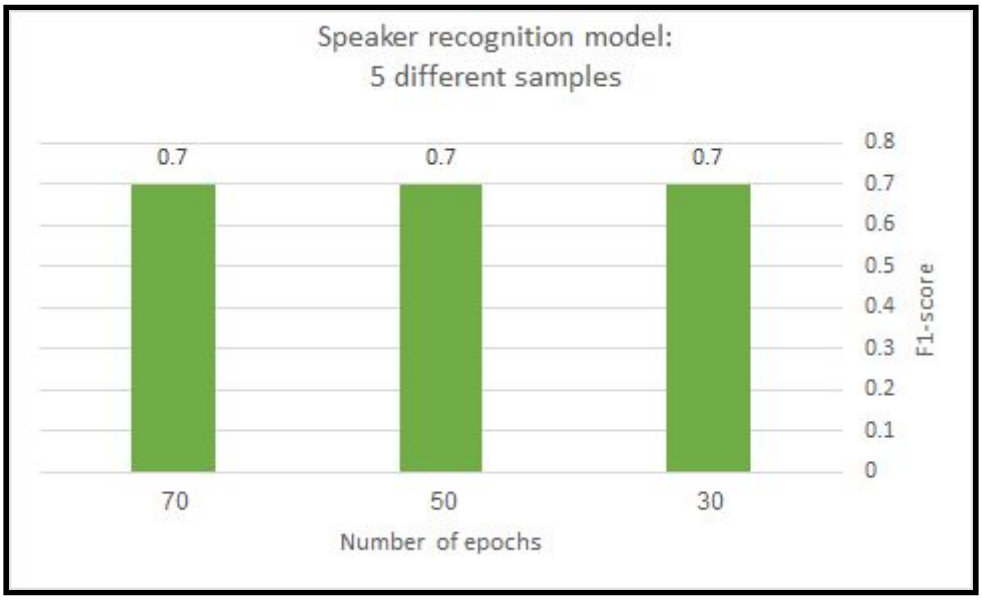


Figure 27
F1-score in speaker recognition model with 5 different samples.

Figures 22,24 and 26 are a comparison of precision,recall and F1-score between 30,50 and 70 epochs for 5 identical samples.
Figures 23,25 and 27 are a comparison of precision,recall and F1-score between 30,50 and 70 epochs for 5 different samples.
From the figures we can see that the precision,recall and F1-score are identical.
In both models there's no change between the epochs however the model with the different samples has slightly higher results.

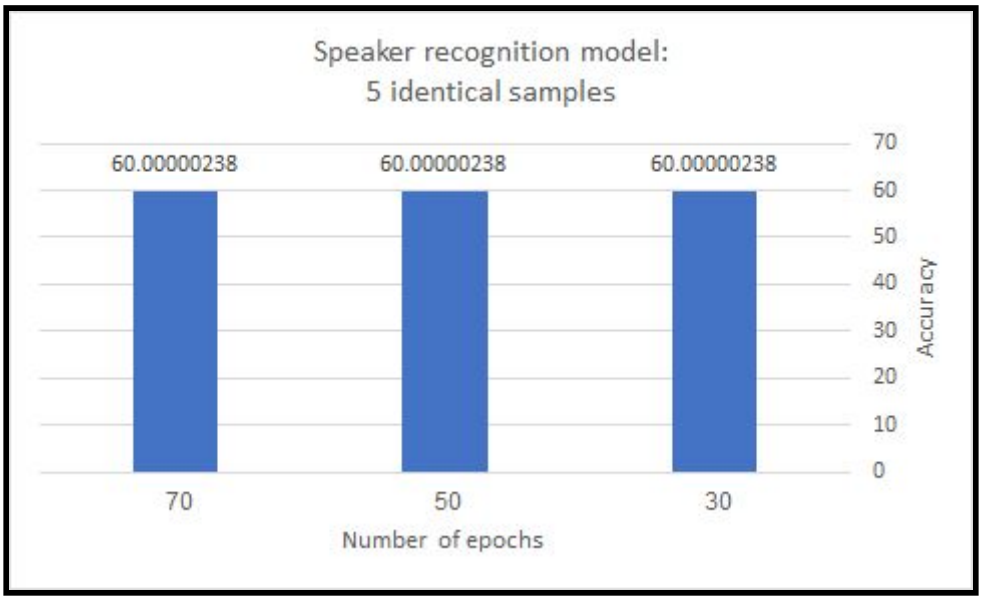


Figure 28
Accuracy in speaker recognition model with 5 same samples.

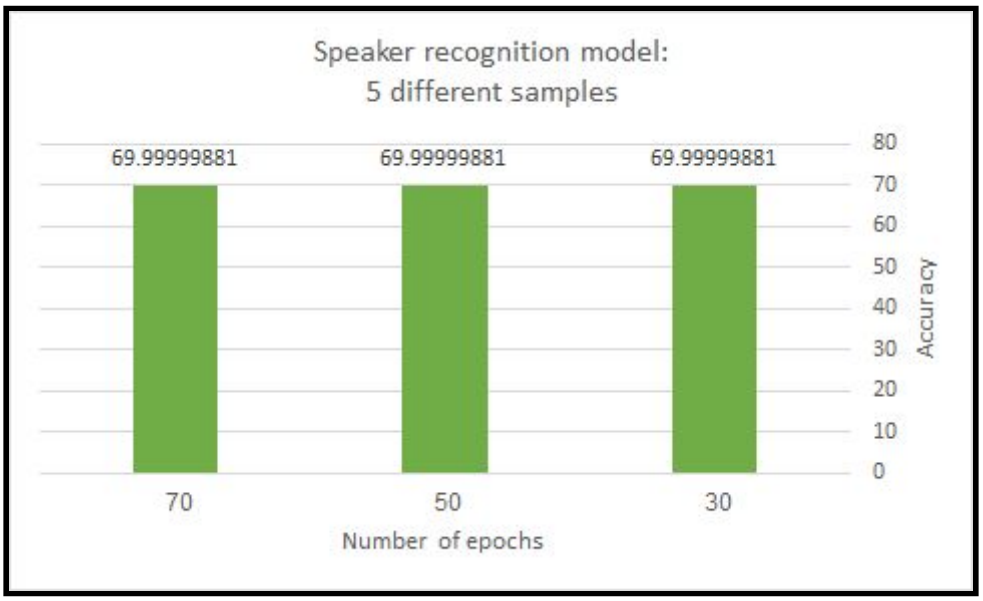


Figure 29
Accuracy in speaker recognition model with 5 different samples.

Figures 28 and 29 are a comparison of accuracy between 30,50 and 70 epochs for 5 identical samples and 5 different samples.
In both models there's no change between the epochs however the model with the different samples has slightly higher results.

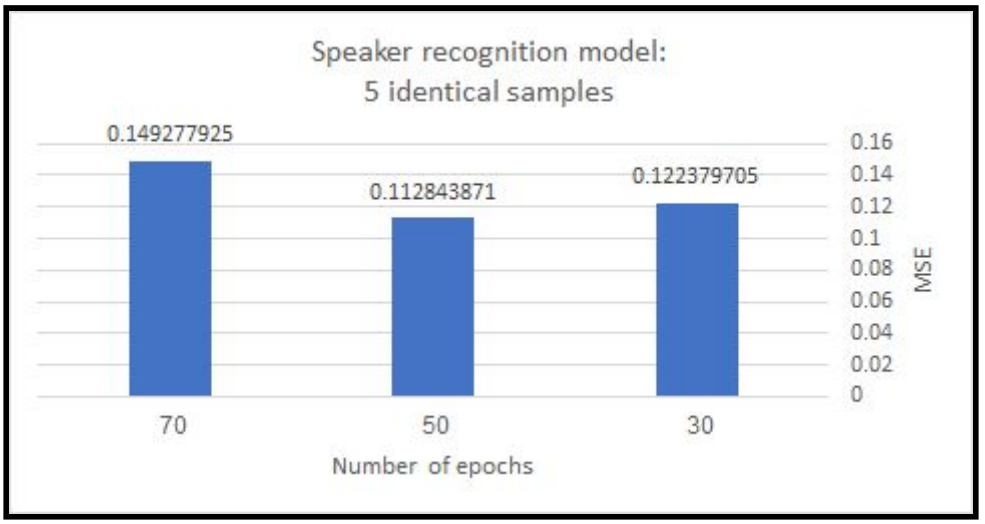


Figure 30
MSE in speaker recognition model with 5 same samples.

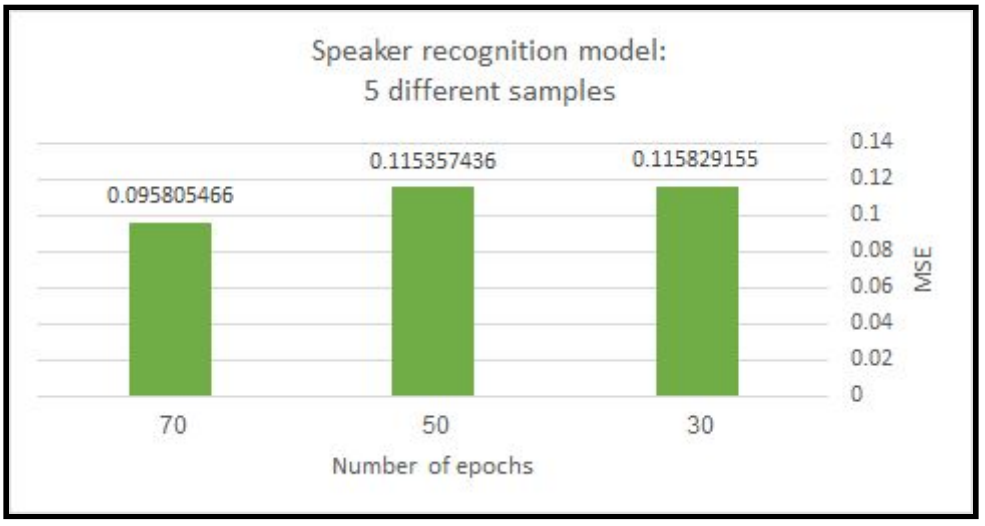


Figure 31
MSE in speaker recognition model with 5 different samples.

Figures 30 and 31 are a comparison of mse between 30,50 and 70 epochs for 5 identical samples and 5 different samples.
In both models there's no change between the epochs however the model with the different samples has slightly higher results.
In figure 31 we can see that there's no difference in mse in 30 epochs and 50 epochs yet at 70 epochs there is a slight decrease, yet in figure 30 we can see a decrease in 50 epochs and then a rise in 70 epochs.
The mse at the model with the different samples is slightly lower than the model with the identical samples.

10 identical voice samples:

Number of epochs	Precision	Recall	F1-score	Accuracy	MSE
30	0.866667	0.866667	0.866667	86.66666746139526	0.04696150124073029
50	0.785714	0.785714	0.785714	78.57142686843872	0.06206311658024788
70	0.733333	0.733333	0.733333	73.33333492279053	0.06770673394203186

10 different voice samples:

Number of epochs	Precision	Recall	F1-score	Accuracy	MSE
30	0.733333	0.733333	0.733333	73.33333492279053	0.094951070 8451271
50	0.733333	0.733333	0.733333	73.33333492279053	0.092320889 23454285
70	0.733333	0.733333	0.733333	73.33333492279053	0.010236737 877130508

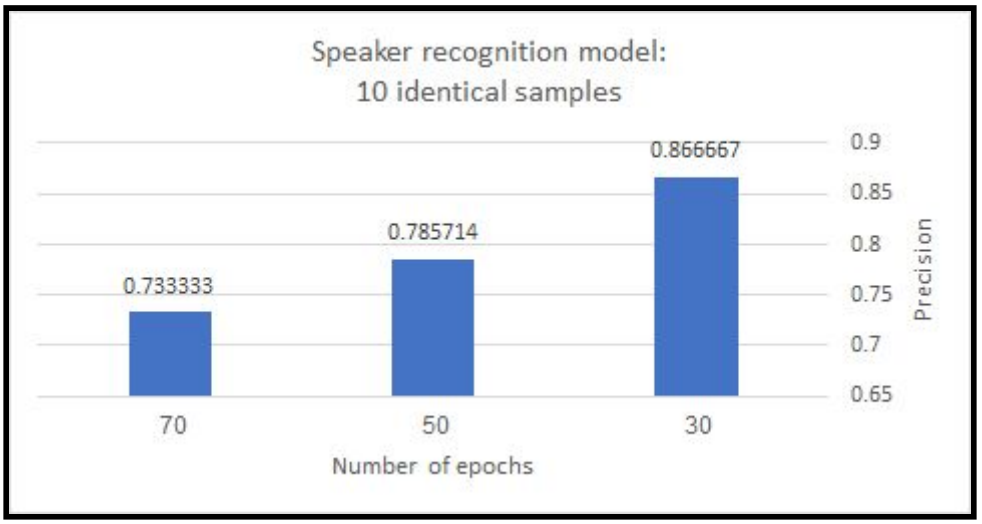


Figure 32
Precision in speaker recognition model with 10 identical samples.

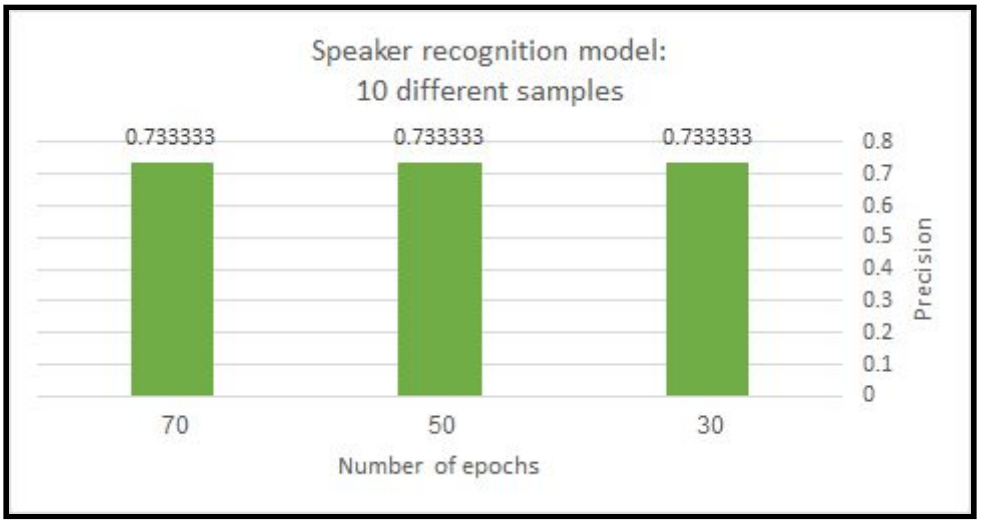


Figure 33
Precision in speaker recognition model with 10 different samples.

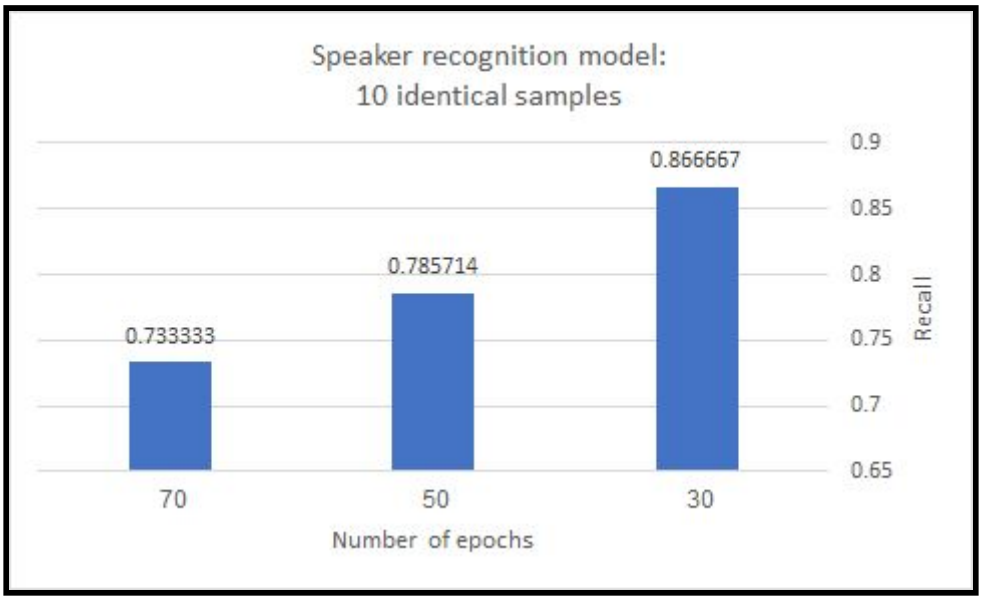


Figure 34
Recall in speaker recognition model with 10 identical samples.

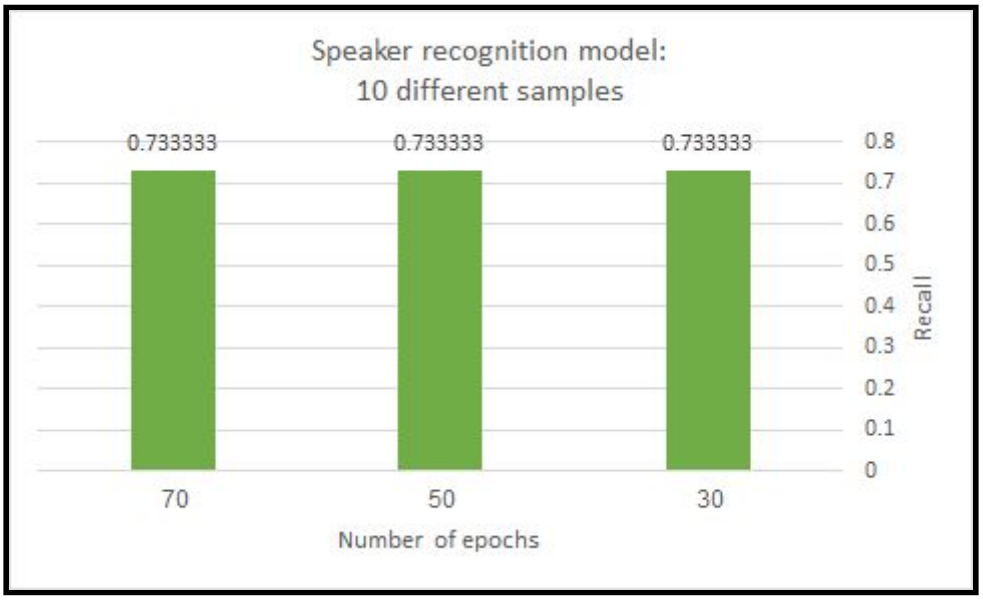


Figure 35
Recall in speaker recognition model with 10 different samples.

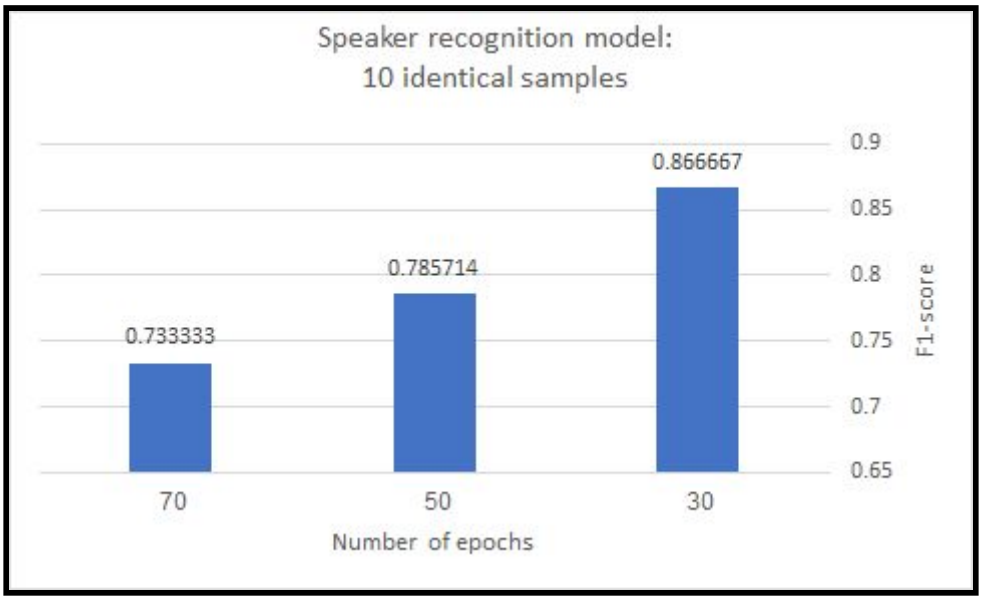


Figure 36
F1-score in speaker recognition model with 10 identical samples.

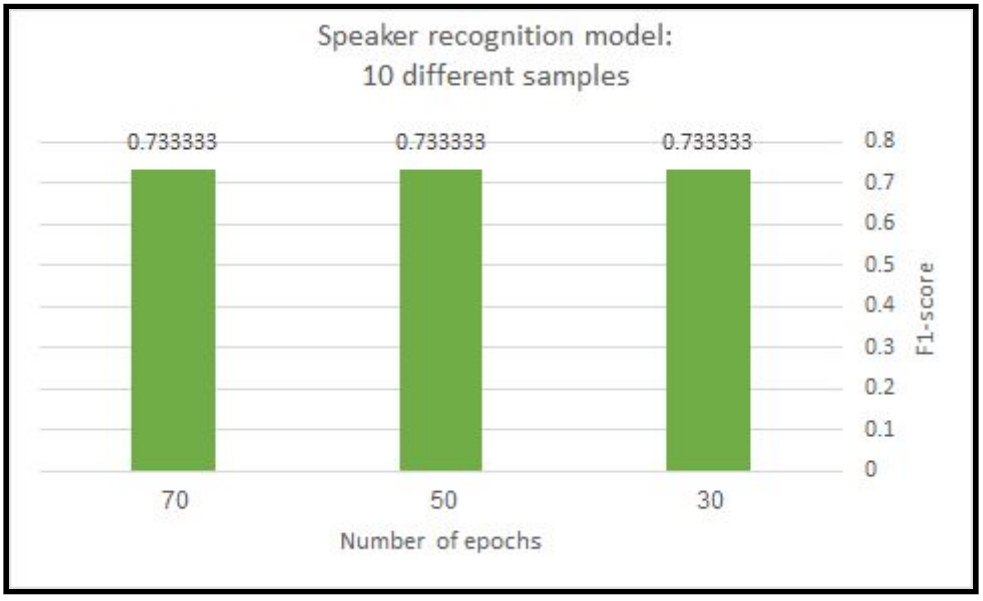


Figure 37
F1-score in speaker recognition model with 10 different samples.

Figures 32,34 and 36 are a comparison of precision,recall and F1-score between 30,50 and 70 epochs for 10 identical samples.
Figures 33,35 and 37 are a comparison of precision,recall and F1-score between 30,50 and 70 epochs for 10 different samples.
From the model with the identical samples we can see a decrease in the results as we raise the epoch number however the model with the different samples stays constant through all the epochs.

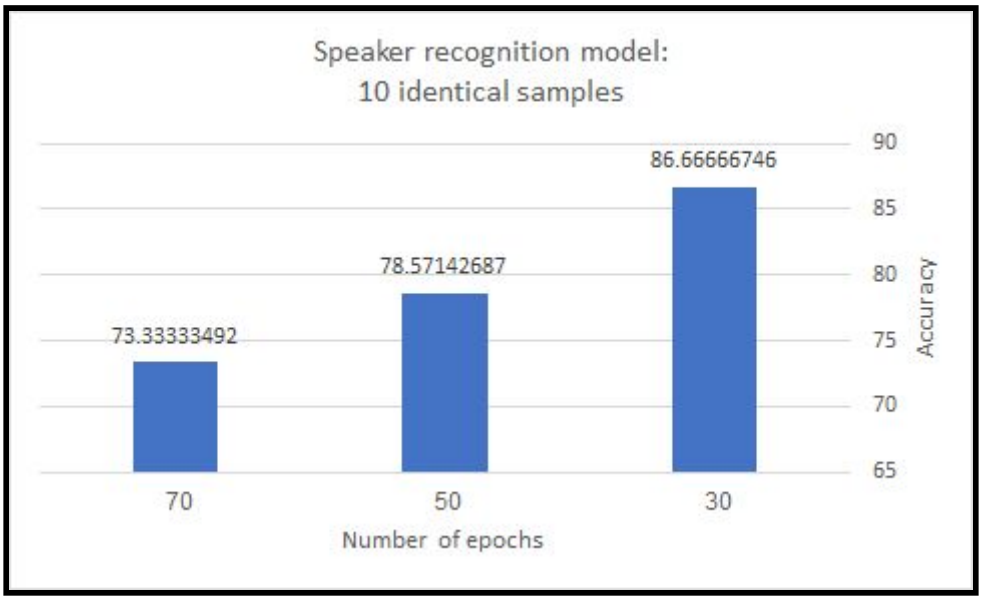


Figure 38
Accuracy in speaker recognition model with 10 identical samples.

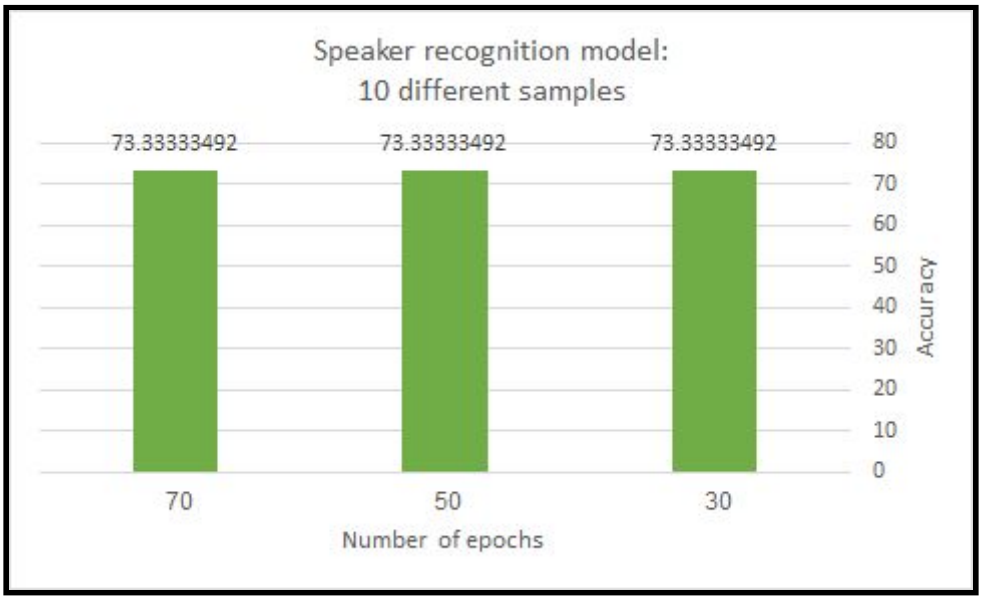


Figure 39
Accuracy in speaker recognition model with 10 different samples.

Figures 38 and 39 are a comparison of accuracy between 30,50 and 70 epochs for 10 identical samples and 10 different samples.
In the identical model theres a decrease in accuracy as we raise the epoch number yet in the different samples model the accuracy is constant.

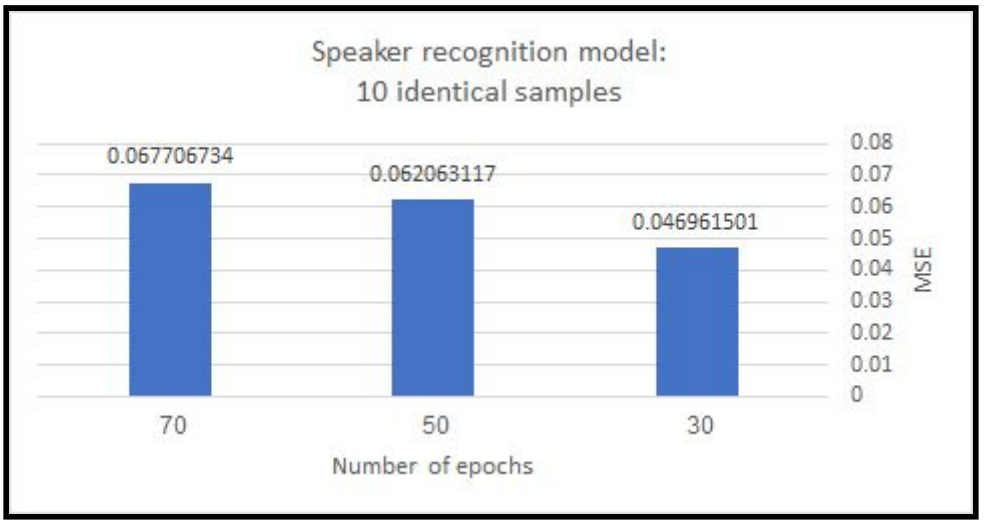


Figure 40
MSE in speaker recognition model with 10 identical samples.

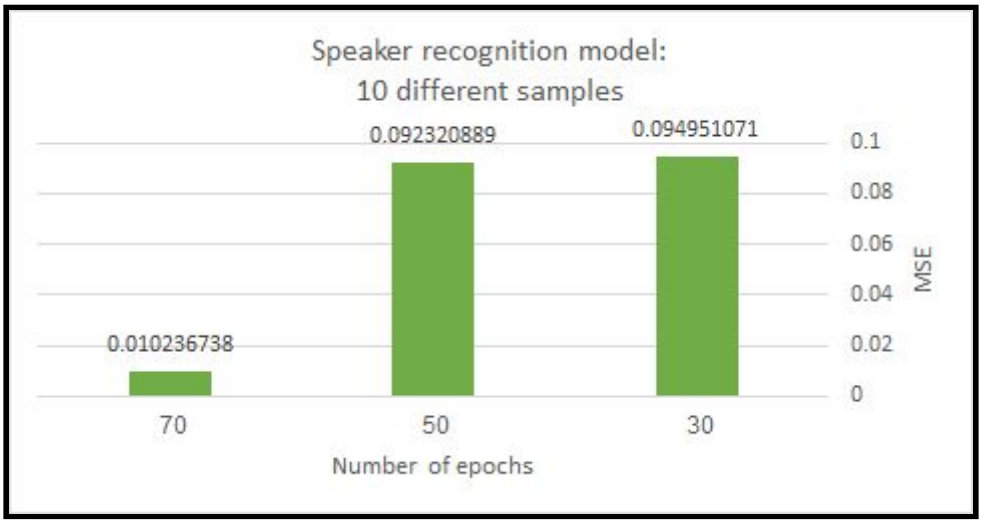


Figure 41
MSE in speaker recognition model with 10 different samples.

Figures 40 and 41 are a comparison of mse between 30,50 and 70 epochs for 10 identical samples and 10 different samples.
In the model with the identical samples there's an increase in mse yet in the model with the different samples the 30 epochs and the 50 epochs are equal but there's a significant descent in the 70th epoch.

15 identical voice samples:

Number of epochs	Precision	Recall	F1-score	Accuracy	MSE
30	0.800000	0.800000	0.800000	80.00000119 20929	0.049049276 85856819
50	0.880000	0.880000	0.880000	87.99999952 316284	0.045068997 88975716
70	0.840000	0.840000	0.840000	83.99999737 739563	0.045599460 60180664

15 different voice samples:

Number of epochs	Precision	Recall	F1-score	Accuracy	MSE
30	0.800000	0.800000	0.800000	80.00000119 20929	0.063153699 0404129
50	0.720000	0.720000	0.720000	75.99999904 632568	0.081388235 09216309
70	0.720000	0.720000	0.720000	72.00000286 102295	0.072000002 86102295

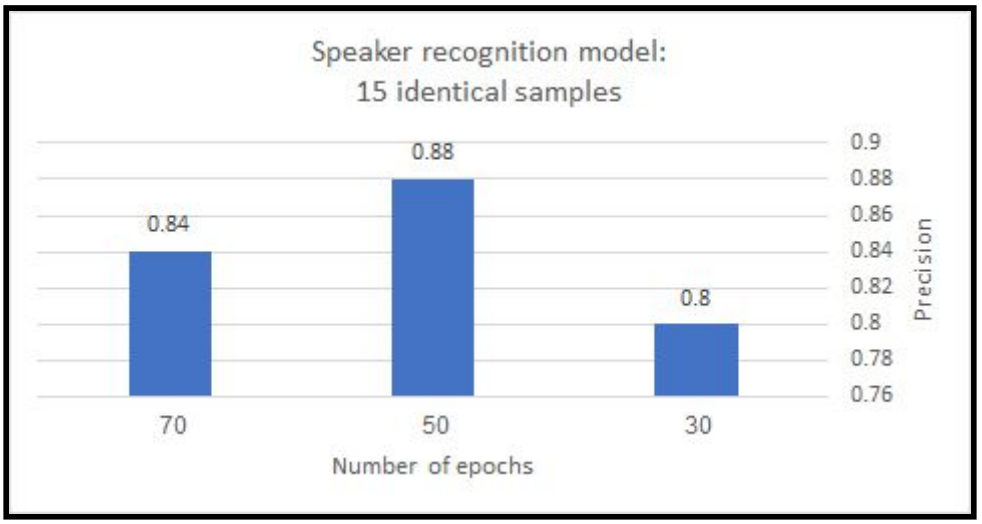


Figure 42
Precision in speaker recognition model with 15 identical samples.

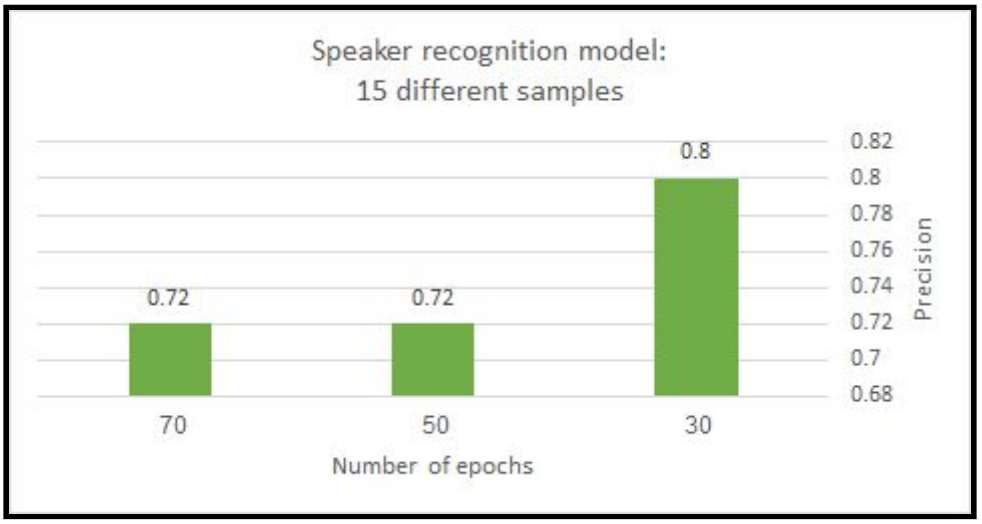


Figure 43
Precision in speaker recognition model with 15 different samples.

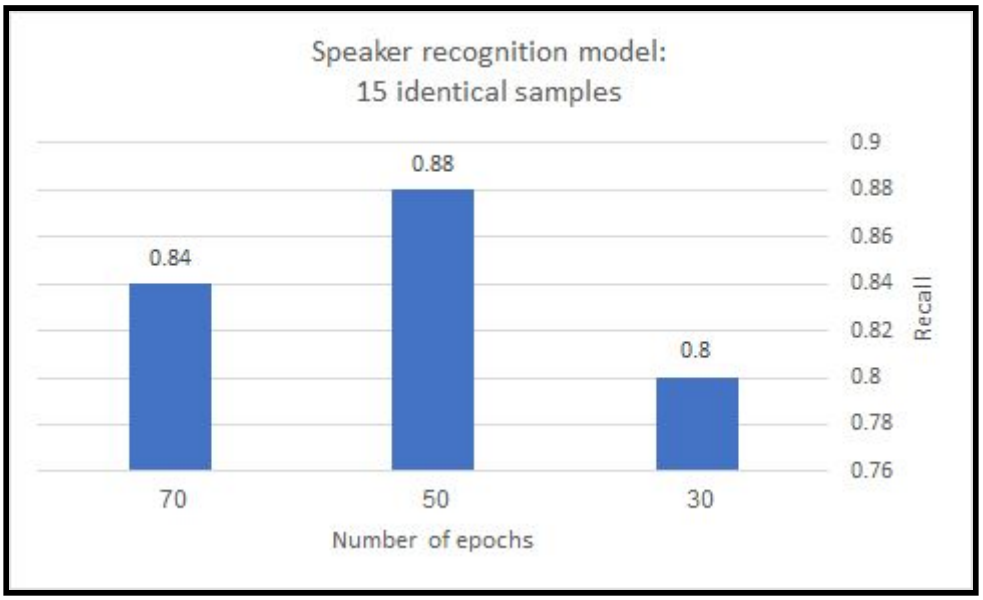


Figure 44
Recall in speaker recognition model with 15 identical samples.

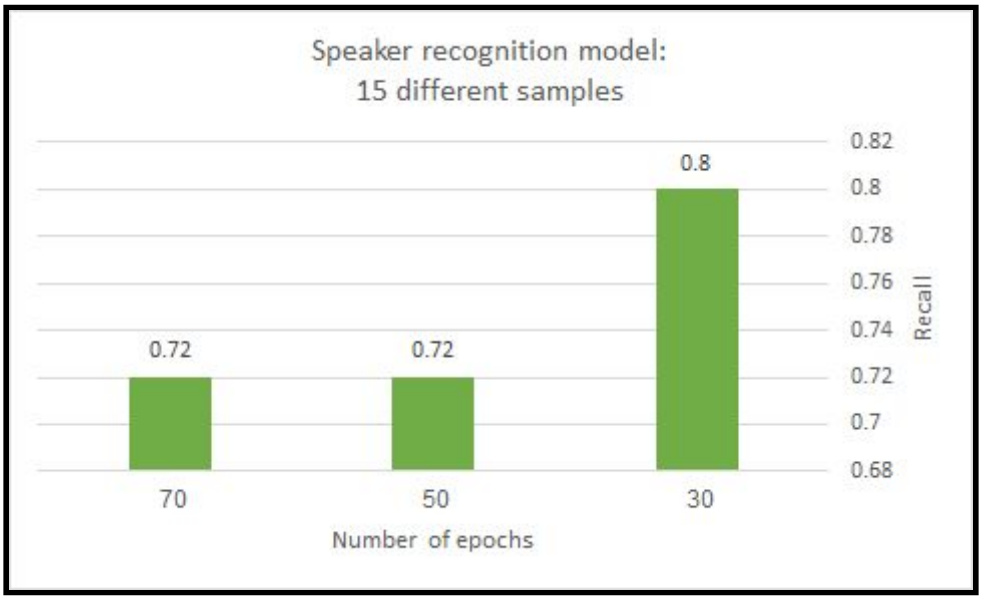


Figure 45
Recall in speaker recognition model with 15 different samples.

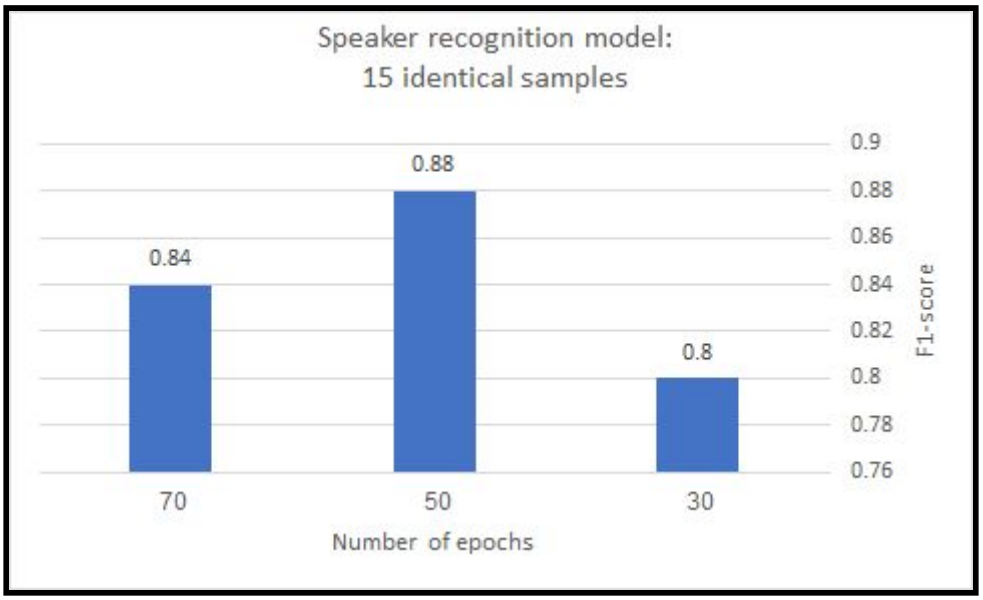


Figure 46
F1-score in speaker recognition model with 15 identical samples.

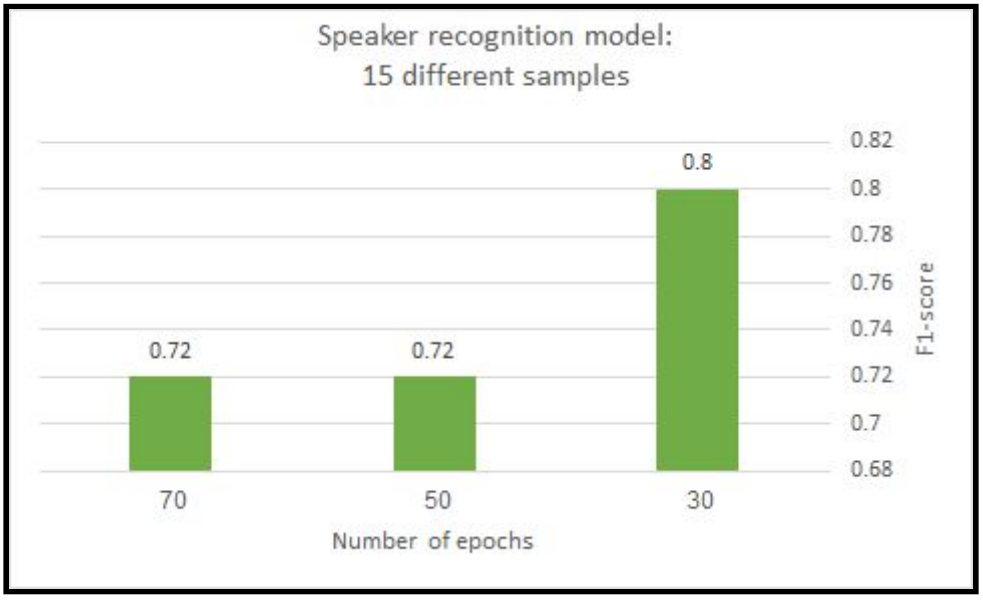


Figure 47
F1-score in speaker recognition model with 15 different samples.

Figures 42,44 and 46 are a comparison of precision,recall and F1-score between 30,50 and 70 epochs for 15 identical samples.
Figures 43,45 and 47 are a comparison of precision,recall and F1-score between 30,50 and 70 epochs for 15 different samples.
From the model with the identical samples we can see an increase between 30 epochs and 50 epochs but in 70 epochs there's a slight descent however the model with the different samples has decrease between 30 epochs and 50 epochs but no change as we raise to 70 epochs.

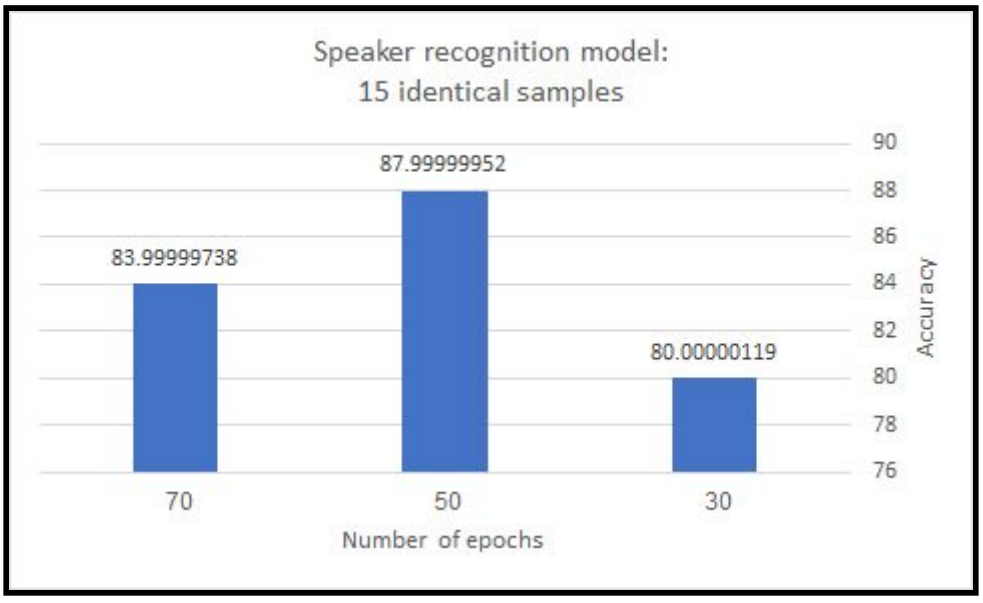


Figure 48
Accuracy in speaker recognition model with 15 identical samples.

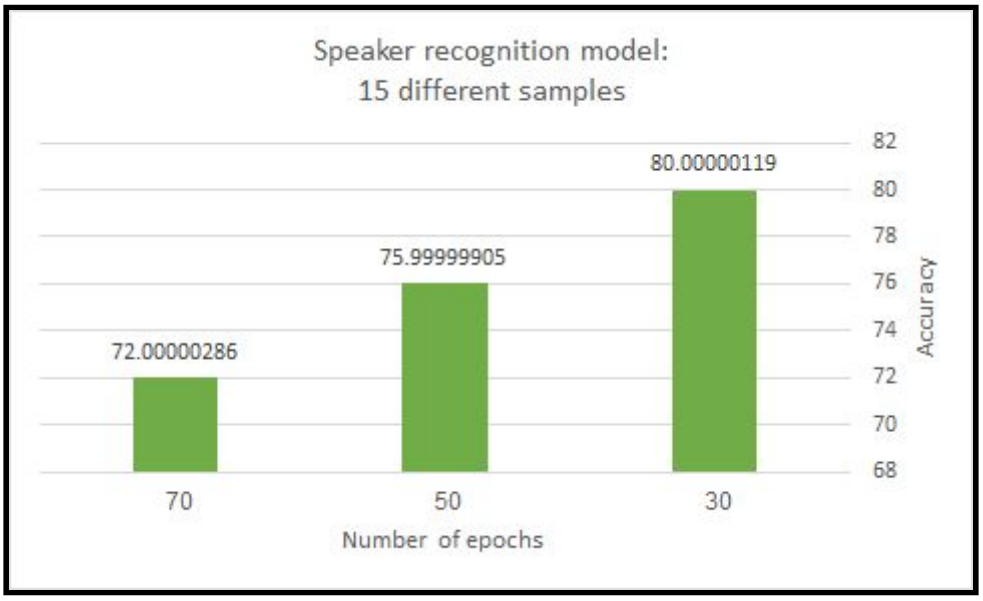


Figure 49
Accuracy in speaker recognition model with 15 different samples.

Figures 48 and 49 are a comparison of accuracy between 30,50 and 70 epochs for 15 identical samples and 15 different samples.
In the identical model theres increase between 30 epochs and 50 epochs but in 70 epochs there's a significant descent.
In the different samples model we can see a decrease in accuracy as we increase the number of epochs.

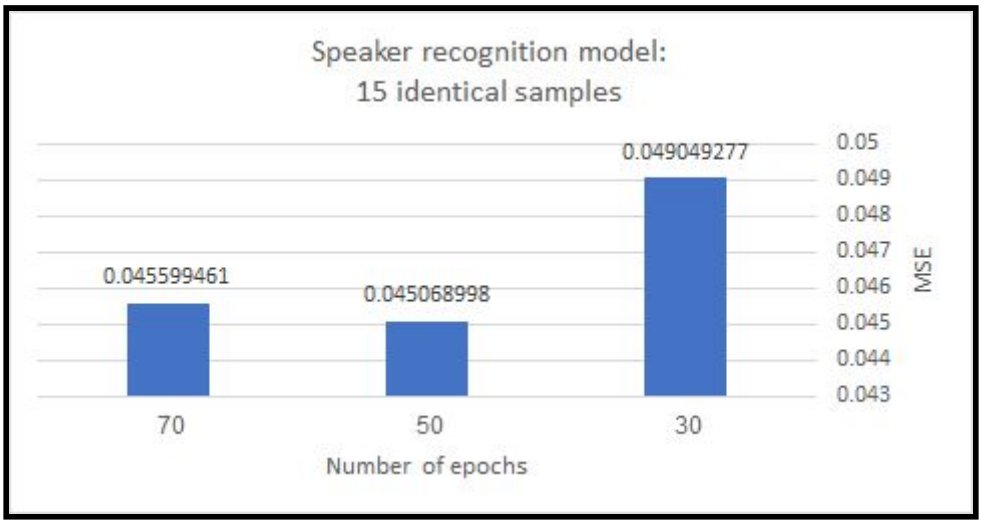


Figure 50
MSE in speaker recognition model with 15 identical samples.

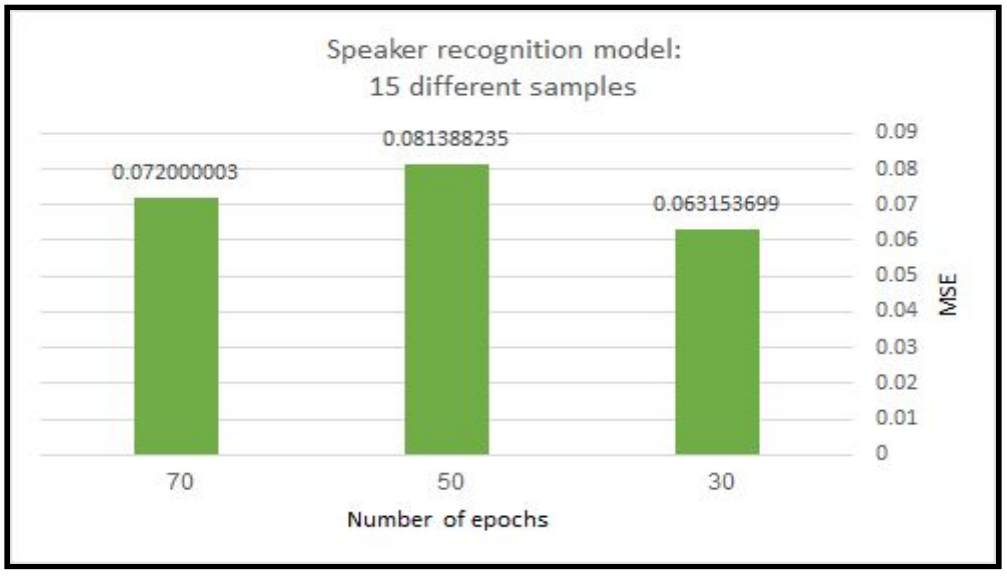


Figure 51
MSE in speaker recognition model with 15 different samples.

Figures 50 and 51 are a comparison of mse between 30,50 and 70 epochs for 15 identical samples and 15 different samples.
In the model with the identical samples there's a decrease between 30 epochs and 50 epochs but in 70 epochs there's a negligible rise.
In the model with the different samples we can see an increase between 30 epochs and 50 epochs but in 70 epochs there's a slight descent.

Conclusion:

After many experiments we can clearly see that models with identical samples perform better than models with different samples.
Models with 5 samples tend to give the same results for identical and different samples' no matter the number of epochs due to small amounts of data.
Models with 10 samples have better results, however models with 15 samples have the best results overall.

10.4. conclusion

	Precision	**Recall**	**F1-score**	**Accuracy**	**MSE**
Word recognition	0.94	0.94	0.94	93.2314395904541	0.0003440920496359467 5
Speaker recognition	0.880000	0.880000	0.880000	87.99999952316284	0.0004506899788975716

We ended up choosing the Word recognition model with 1000 epochs and 41 labels due to overall satisfying results and satisfying predictions.
The speaker recognition we choose the model with 15 identical samples and 50 epoches due to having the best results overall.

11. Testing

11.1. The goal

The goal of testing the software is to check whether it Meets the requirements.

11.2. description of the software

The software is intended to work in the background and if it recognizes that the user is being called it will lower the volume and inform the user who is the caller.

11.3. Planning stages

Functional testing: The goal of these tests is to make sure that the system is fulfilling the requirements from the Initiation & Characterization documents.

UI testing: The goal of these tests is to make sure that the user interface is functioning.

11.4. STP+STD

Functional testing:

STP test number	Test description	The purpose of the test	Step \ process	Required result	passed\ failed
1	test if mfcc is created	test if the mfcc vector is created successfully	read an audio file and send it to the function to create a vector	vector with type numpy	passed
			read an audio file and send it to a different function to create a vector	vector with type numpy	passed

STP test number	Test description	The purpose of the test	Step \ process	Required result	passed\ failed
2	test if the sound is silenced or back	to see if the sound of the system silenced	activate the function that silences the system and see if the sound is silenced	the audio is disabled	passed
			activate the function that silences the system and see if the sound is back	the audio is enabled	passed

UI testing:

STP test number	Test description	The purpose of the test	Step \ process	Required result	Past \ failed
3	test the login page	test the button and functions in the login page	Login is successful with existing user	prompts a message of successful login	passed
			Login is unsuccessful with nonexistent user	Stays on the login page and prompts a message	passed
			Login button working and goes to main page	Goes to main page	passed
			Register button working and goes to register page	Goes to register page	passed

STP test number	Test description	The purpose of the test	Step \ process	Required result	Past \ failed
4	test the register page	test the button and functions in the register page	Register button does add the new account to database and goes to the train page	goes to the train page	passed
			Checks if passwords are matching	Lets you to register and go to the	passed

				train page	
			Close button goes back to login page	goes back to the login page	passed

STP test number	Test description	The purpose of the test	Step \ process	Required result	Past \ failed
5	test the main page	test the functions on the main page	Record button starts the startRecording function	Starts Recording	passed
			Record button animates check when clicked and another click unchecks it	Animated clicked button	passed
			Your name displaying your name from your account X	Displays the name you chose	passed
			can go to options tab and to main tab X	Alternates between tabs	passed
			change name changes the name	Changes the name in the display	passed

			Add people starts the addPeople function	Add new people to the text file	passed
			Add new recording starts recording	Adds new recording to Recordings folder	passed
			record button starts recording	record button records the new recording	passed

12. Bibilograpy

[1].Arora, Preeti, and Shipra Varshney. "Analysis of k-means and k-medoids algorithm for big data." Procedia Computer Science 78 (2016): 507-512.

[2].Beigi, Homayoon. Speaker Recognition: Advancements and Challenges. INTECH Open Access Publisher, 2012.

[3].Boney, Rinu and Alexander Ilin. “Semi-Supervised Few-Shot Learning with Prototypical Networks.” ArXiv abs/1711.10856 (2017): n. pag.

[4].Fort, Stanislav. “Gaussian Prototypical Networks for Few-Shot Learning on Omniglot.” ArXiv abs/1708.02735 (2018): n. pag.

[5].Koch, Gregory R.. “Siamese Neural Networks for One-Shot Image Recognition.” (2015).

[6].Montana, David J. and Lawrence Davis. “Training Feedforward Neural Networks Using Genetic Algorithms.” *IJCAI* (1989).

[7].Pouyanfar, Samira, Saad Sadiq, Yilin Yan, Haiman Tian, Yudong Tao, Maria E. Presa Reyes, Mei-Ling Shyu, Shu-Ching Chen and S. S. Iyengar. “A Survey on Deep Learning: Algorithms, Techniques, and Applications.” ACM Comput. Surv. 51 (2018): 92:1-92:36.

[8].Shanmuganathan, Subana, and Sandhya Samarasinghe, eds. Artificial Neural Networks Modelling. Springer, Cham, n.d.

[9].Smith, Steven W.. “The Scientist and Engineer's Guide to Digital Signal Processing.” (1997).

[10].Snell, Jake, Kevin Swersky and Richard S. Zemel. “Prototypical Networks for Few-shot Learning.” NIPS (2017).

[11].Togneri, Roberto, and Daniel Pullella. “An Overview of Speaker Identification: Accuracy and Robustness Issues.” IEEE Circuits and Systems Magazine 11, no. 2 (June 9, 2011): 23–61. https://doi.org/10.1109/mcas.2011.941079.

[12].Trivedi, nbspPahini A.. “Introduction to Various Algorithms of Speech Recognition: Hidden Markov Model, Dynamic Time Warping and Artificial Neural Networks.” (2014).

[13].Yoon, Byung-Jun. “Hidden Markov Models and Their Applications in Biological Sequence Analysis.” Current Genomics 10, no. 6 (January 2009): 402–15. https://doi.org/10.2174/138920209789177575.

[14].Zhang, Zixing, Ding Liu, Jing Han and Björn W. Schuller. “Learning Audio Sequence Representations for Acoustic Event Classification.” ArXiv abs/1707.08729 (2017): n. pag.

Data resources:

[15].An end-to-end open source machine learning platform.*TensorFlow*, www.tensorflow.org.
[16].High-level neural networks API, written in Python and capable of running on top of TensorFlow, CNTK, or Theano.*Keras: The Python Deep Learning library*, www.keras.io.
[17].A fundamental package for scientific computing with Python.*NumPy*, www.numpy.org.
[18].API design for machine learning software.*scikit-learn*, www.scikit-learn.org/stable.
[19].Python package for music and audio analysis.*LibROSA*, www.librosa.github.io/librosa.
[20].audio input/output stream library.*PyAudio*, www.pypi.org/project/PyAudio.
[21].A simple audio/speech dataset consisting of recordings of spoken digits,*FSDD*. www.github.com/Jakobovski/free-spoken-digit-dataset.
[22].P. Foster, S. Sigtia, S. Krstulovic, J. Barker, M. D. Plumbley. "CHiME-Home: A Dataset for Sound Source Recognition in a Domestic Environment," in Proceedings of the 11th Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA), 2015
[23].Corpus of read speech.*TIMIT*, www.github.com/philipperemy/timit/blob/master/README.md
[24].An open-source toolkit for commercial-grade distributed deep learning.*CNTK*, www.docs.microsoft.com/en-us/cognitive-toolkit.
[25]. A Python framework for fast computation of mathematical expressions.*Theano*, www.deeplearning.net/software/theano/#.
[26].A Python-based ecosystem of open-source software for mathematics.*SciPy*, www.scipy.org/index.html.
[27].Hunter, John D. “Matplotlib: A 2D Graphics Environment.” Computing in Science & Engineering 9, no. 3 (June 18, 2007): 90–95. https://doi.org/10.1109/mcse.2007.55.
[28]A module that provides bindings for the PortAudio library.*python-sounddevice*, www.python-sounddevice.readthedocs.io/en/latest.
[29].An API for recording and/or playing sound. *PortAudio*, www.portaudio.com.
[30].Albanese, Davide, Roberto Visintainer, Stefano Merler, Samantha Riccadonna, Giuseppe Jurman and Cesare Furlanello. “mlpy: Machine Learning Python.” ArXiv abs/1202.6548 (2012): n. pag.
[31].Python library covering a wide range of audio analysis tasks.*pyAudioAnalysis*, www.github.com/tyiannak/pyAudioAnalysis.
[32].Derousseau, Ryan. “What to Do about Your Noisy Office.” BBC Worklife. BBC, April 26, 2017. https://www.bbc.com/worklife/article/20170426-what-to-do-about-your-noisy-office.
[33].A programming language.*Python*, www.python.org.
[34].Amazon patent.*United States Patent*, www.patft.uspto.gov/netahtml/PTO/index.html.

[35].amazon noise cancelling headphones.The guardian, www.theguardian.com/technology/2016/aug/01/amazon-noise-cancelling-headphones-know-your-name.
[36].Gui-ming, Du, Wang Xia, Wang Guang-yan, Zhang Ya-n and Li Dan. “Speech recognition based on convolutional neural networks.” *2016 IEEE International Conference on Signal and Image Processing (ICSIP)* (2016): 708-711.

www.sce.ac.il

קמפוס באר שבע
ביאליק 56, באר שבע 8**4**100

קמפוס אשדוד
ז'בוטינסקי **84**, אשדוד 77245