

Software Engineering Department

ORT Braude College

Capstone Project Phase B – 61998

**MTGAN: Speaker Verification through Multitasking Triplet Generative Adversarial Networks**

22-1-R-28

Deniss Nissim 308328673

deniss4293@gmail.com

Yonatan Ben Avraham 312592009

yoniben222@gmail.com

Supervisors: Prof. Zeev Volkovich

Dr. Renata Avros

**Table of contents**

**1.Introduction3**

**2.Related Works4**

**3.Background 4**

**3.1 Convolutional Neural Network (CNN)4**

**3.1.1 Convolutional Layer4**

**3.1.2 Max-Pooling Layer 5**

**3.1.3 Flatten Layer 6**

**3.1.4 Fully Connected Layer 6**

**3.1.5 Dropout 6**

**3.1.6 Batch Normalization 7**

**3.1.7 Activation Function 8**

**3.1.7.1 Softmax Function 8**

**3.1.7.1.1 Activation 8**

**3.1.7.1.2 Loss Function9**

**3.1.7.2 ReLU Activation Function 10**

**3.1.7.3 Sigmoid Activation Function 10**

**3.2 LSTM 11**

**3.3 Short Time Fourier Transform (STFT)11**

**3.4 Wavelet Transform 12**

**3.4.1 Continuous Wavelet Transform (CWT)12**

**3.4.2 Discrete Wavelet Transform (DWT)13**

**5. Research Process 14**

**5.1 Process Description 14**

**5.2 Experiments 18**

**5.3 Product 19**

**5.3.1 Use Case Diagram 19**

**5.3.2 Package Diagram 21**

**5.3.3 User interface22**

**6. Evaluation/Verification Plan 29**

**7. References 30**

**Abstract:**

**This project is based a model combining various Deep Learning tools intended to perform speaker verification of short unknown utterances of speakers. The model introduces a new loss function called generalized end-to-end (GE2E) loss, which makes the training of speaker verification models very efficient. Also, The model uses a LSTM network. The project novelty lies in applying a Wavelet transform, making it possible to increase the system performance.Key words: Long Short Term Memory (LSTM), Generalized end-to-end (GE2E) loss, Softmax, speaker verification.**

**1. Introduction:**

Nowadays, there is a steady increase in the number of types of daily operations that can be performed virtually due to the constant development of technology. For example, you can already perform transactions related to your personal bank account via the Internet instead of physically accessing the bank branch, making appointments to places, sending files and documents and there are lots of other examples. But because of these developments new risks are born. It is much easier to perform actions on behalf of a particular person and thus impersonate him. Even the use of passwords may fail to provide protection against different attackers. There is therefore a need for a more powerful tool that will protect user privacy and speaker verification may be a particularly effective tool for dealing with this problem. As part of our project, our goal is to verify and improve the solution proposed in [1]. It is going to be done by involving modern skills like the Wavelet Spectrogram which is applied for several types of Wavelet transformations and their fusion.

**2.** **Related Works**

This section discusses the basic principles and especially other works done in this field. [[16](#sixteen)] and [[17](#six)] are state of the art methods in the field of Automatic Speaker Verification (ASV). However, in the last few years, many works found that systems composed of Deep Neural Networks (DNN) achieves better success that the traditional methods, especially in cases where the samples are short. Triplet Loss is a very effective method for classification tasks, consists Deep Neural Networks (DNNs). FaceNet [[6](#six)], which is a model of face recognition, used this method. [[3](#three)] Implements this method for speaker verification. The Triplet method is known as an efficient and effective method and many successful works have been based on it [[5](#five)][[8](#eight)]. Although the effectiveness of Triplet Loss in classification tasks, there are many restrictions such as limited training samples and noise during recording which might damage the task of speaker verification. The major disadvantage of Encoders using Triplet methods is that they have less generality and diversity and thus might fail in test phase. To overcome these issues, a Generative Adversarial Network (GAN) is added to the model. [[2](#two)] and [[10](#ten)] used Encoders behind GANs and achieved great success. Unlike traditional GANs, in addition to the random input vector that fed into the Generator, it receives also embeddings that represent real samples. This type of framework enables achieving more generality and diversity for new samples. The Discriminator verifies that the generated from Encoder/Generator are real. A Classifier receives images both from the Generator and the Encoder (Spectrograms). The output of the Classifier is the ID of the speaker and it is obtained after performing of Softmax Loss in the last layer.

**3.** **Background:**

This section presents the theoretical background of the project.

**3.1 Convolutional Neural Network (CNN):**

CNN is a type of artificial neural network which is used primarily for image analysis.

**3.1.1 Convolutional Layer:**

A convolution layer is a fundamental component of the CNN architecture that performs feature extraction.

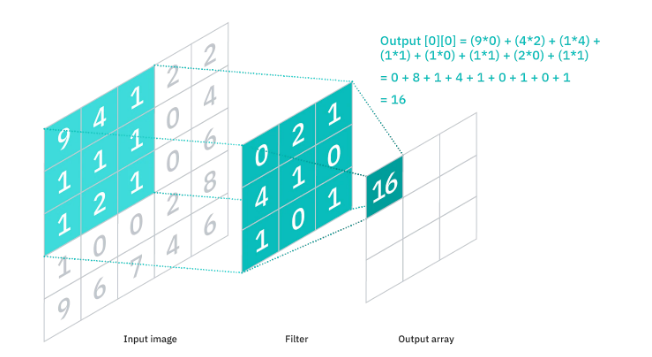
At this point there is input and filter. The filter is doubled in input. Then summarize everything into one result, for illustration you can look at the picture. This process continues and the filter moves one pixel at a time in order until it covers the entire input. The result is called a feature map.

Figure 1

**3.1.2 Max-Pooling Layer:**

The max-pooling layer reduces the output obtained from the convolutional layer by taking the maximum values. That is, max-pooling layers are used to reduce the dimensions of feature maps. The goal is to reduce the sample size of the input, thus reducing the number of computational operations required and speeding up the computation time.

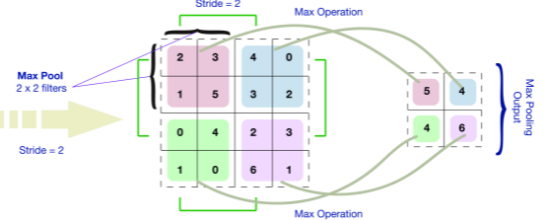


Figure 2

**3.1.3 Flatten Layer:**

Flatten is used to convert the data from a two-dimensional matrix into a single feature vector. After flattening we forward the data to a fully connected layer for final classification.

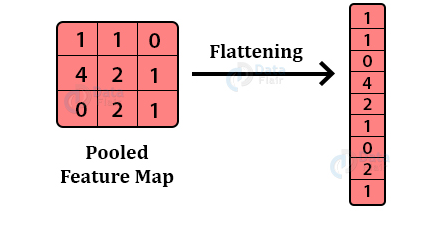


Figure 3

**3.1.4 Fully Connected Layer:**

Fully Connected layers in neural networks are those layers where all the inputs from one layer are connected to every activation unit of the next layer. Typically, the last few layers are full connected layers that compile the data extracted by previous layers to form the final output.

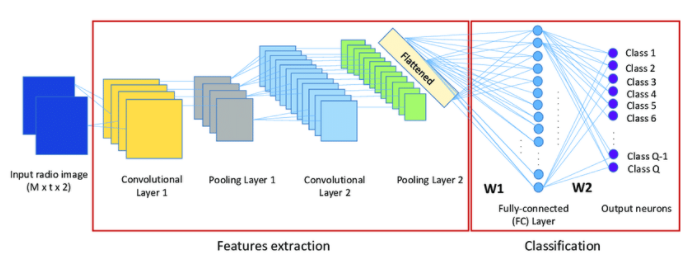
****

Figure 4

**3.1.5 Dropout:**

Dropout is a technique used to prevent a model from overfitting.

The Dropout layer randomly sets input units to 0 with a frequency of rate at

each step during training time. Dropout reduces overfitting but requires extra

computational training costs.

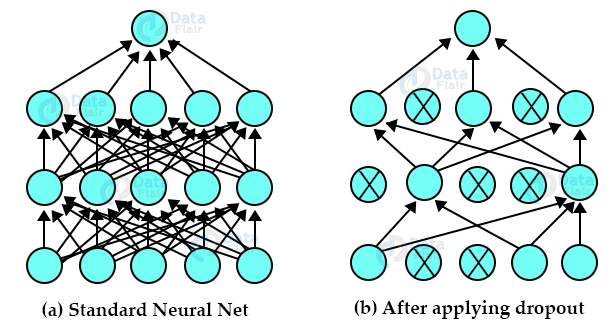


Figure 5

**3.1.6 Batch Normalization:**

Batch Normalization is a very common method that used for training.

Deep Neural Networks and it also used in the model.

It standardizes the inputs to a layer for each mini-batch. This has the effect of stabilizing the learning process and dramatically reducing the number of training epochs required to train deep networks.

The following formulas describe the steps of this method:

1. // mini batch mean
2. // mini batch variance
3. // normalize
4. // scale and shift

Where xi is a single value of the mini-batch, γ and β are parameters to be learned.

**3.1.7** **Activation Function:**

An activation function is a function that is added to an artificial neural network to help the network learn complex data patterns. The activation function is ultimately decides whether to activate the next neuron. The decision is made by weighted calculations with bias. Types of operating functions: softmax, ReLu, and sigmoid and we explain each of them separately.

**3.1.7.1 Softmax Function:**

There are two types of Softmax functions, the one if Softmax Activation and the second is Softmax Loss:

**3.1.7.1.1**  **Activation**:

Softmax function converts values into probabilities. Here is the formulation of the function:

The Softmax activation function is often placed at the output layer of a neural network. The output of the Softmax describes the probability of the neural network that a particular sample belongs to a certain class.

We can see that one advantage of using the Softmax at the output layer is that it improves the interpretability of the neural network.

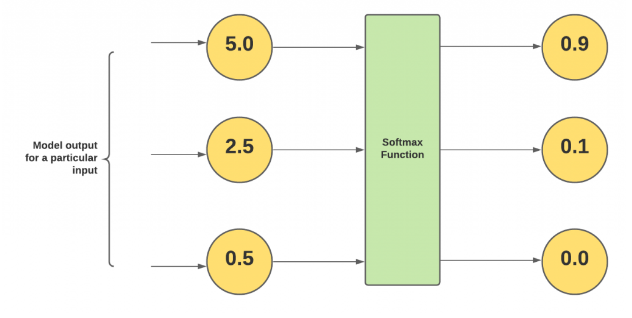


Figure 6

**3.1.7.1.2 Loss Function**

The Softmax Loss function also called a Categorical Cross- Entropy. This function is used in multi-class classification tasks. These are tasks where anexample can only belong to one out of many possible categories, and the model must decide which one.

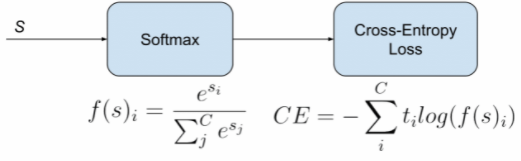


Figure 7

f(s) = CE = -

The picture above shows us that the Cross-Entropy Loss uses the output from the Softmax activation for the calculation of the result. This process describes the working form of the Categorical Cross-Entropy.

**3.1.7.2 ReLU Activation Function:**

The rectified linear activation function or ReLU for short is a partiallylinear function that outputs the input directly if it is positive, otherwise, it outputs zero. The rectified linear activation function overcomes the vanishing gradient problem, allowing models to learn faster and perform better.

Mathematically, it is defined as **y = max(0, x)**.

**3.1.7.3 Sigmoid Activation Function:**

The sigmoid activation function, also called the logistic function, is traditionally a very popular activation function for neural networks. The input to the function is transformed into a value between 0.0 and 1.0. The shape of the function for all possible inputs is an S-shape from zero up through 0.5 to 1.0.

The main reason why we use sigmoid function is because it exists between(0 to 1). Therefore, it is especially used for models where we have to predict the probability as an output. Since probability of anything exists only between the range of 0 and 1, sigmoid is the right choice.

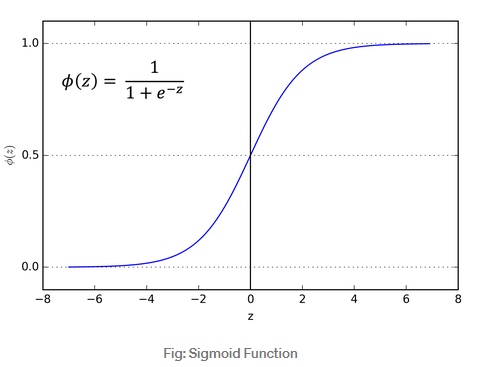


Figure 8

**3.2 LSTM**

lstm is very similar to the RNN network.

Memory blocks contain memory cells and gates. The function of the memory cells is to store the temporary state of a network, with the role of the gates being to control the flow of information.

There are 3 gates, each with a different role:

First, we have the forget gate. This gateway decides whether to discard or save the information. Second, the Input Gate which is responsible for updating the cell state and in addition has the Output Gate whose job it is to decide on the next hidden state.

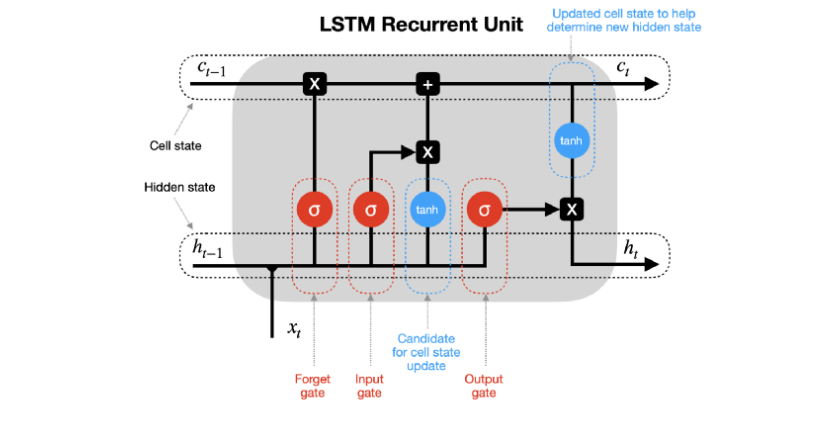


Figure 9

**3.3 Short Time Fourier Transform (STFT):**

In this section, we explain about STFT that will serve as an introduction to wavelets. STFT is a function of time and frequency. In this function, the signal is divided into small segments, where it can be assumed that the parts of the signal are stationary, which means: parts where the frequency does not change over time. In addition, the width of the selected window must be equal to the section where the signal is stationary. Therefore, the size of the window is the same for all frequencies.

The problem is that it is not possible to know the representation of frequency- time accurately, because once you choose a particular size for the time window, that window has the same size for all frequencies. This is called the uncertainty principle.

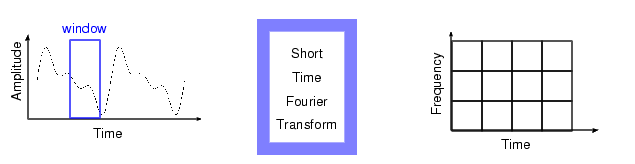


Figure 10

**3.4 Wavelet Transform:**

The Wavelet Transform is a technique of windows of different sizes. When the time interval is long we want more accurate information at a low frequency, while when the time interval is short we want information at a high frequency. Where x is the real signal (also called processed signal), ψ is a mother Wavelet, a is the scale and b is the translation. The scale and size of the window are the same. It can extract information from the transient signals concurrently in the time domains and also frequency domains. Wavelet Transform is superior to Fourier Transform because it gives information about the time of the frequency.

**3.4.1** **Continuous Wavelet Transform** **(CWT):**

The continuous one dimensional Wavelet Transform (CWT) is a

decomposition of x(t) into a set of basis function ψa,b(t) called wavelets

[[12](#twelve)]:

Where: x(t) is the signal to be analyzed, a is the scale, and b is the translation. ψ(t) is the transforming function and it is called the mother wavelet. Filters of different frequencies are used for analyzing the signal.

**3.4.2 Discrete Wavelet Transform (DWT):**

As CWT is a function of two parameters, scale and translation parameters

as a=2j and b=k2j. So DWT requires two sets of functions called Scaling

function and Wavelet function given by:

And:

Where function ψ(t) defined as a scaling function, h[n] is a response of a low pass filter and g[n] is a response of a high pass filter.

The Discrete Wavelet Transform (DWT) uses filter banks to build the time- frequency plane.

A filter bank consists of filters which separate a signal into frequency bands. For a two stage filter it consists of a low pass filter L and a high pass filter H. In Wavelet analysis, the high scale, low frequency components of the signal are called the approximations and the low scale, high frequency components are called the details [[12](#twelve)].

An Approximation is split to Detail and Approximation. Therefore, a sequence of levels is taken. A level represents for us the degree of detail in the analysis. In this example, the number of details is three. D1, D2, D3 are defined as the details and A3 is the approximation. Thus, for each word spoken, only four features are extracted [[18](#eighteen)].

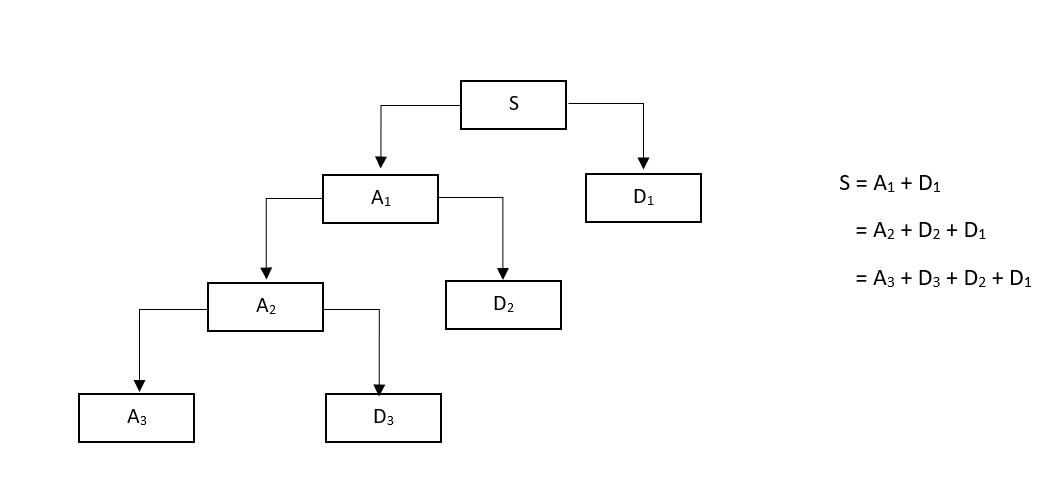


Figure 11: Wavelet Packet Analysis

**5. Research Process:**

The research process consists of several stages. First, we read the paper in [[1](#one)] and tried to understand it in depth. There were some theoretical issues mentioned in the article where we had to deepen the understanding and also learn topics that were completely new to us, only after we did we were able to better understand the algorithm, then we learned about the initial preprocessing and the process of feature extraction. Next, we deepened our understanding regarding various Wavelet transformations [[11](#eleven)][[12](#twelve)][[14](#fourteen)]. Only after we better understood all the theoretical background, algorithm steps and Wavelet transformations, then we started writing the book. The challenges we had were in learning the different topics and finding the relevant sources of information to learn from

**5.1 Process Description:**

The following section explains in detail each section in the training phase of the model.

During the project one of the products we have built is a system that performs speaker verification using a number of tools such as Wavelet transformation and Recurrent Neural Networks. In Phase A of the project we planned to implement the MTGAN system proposed in [1], but after encountering a number of difficulties we decided not to do so. Instead, our research focused on the implementation of the method proposed in [19], which is "Generalized End-To-End Loss For Speaker Verification" with the focus on the implementation of features extraction using the various Wavelet transformations, and comparing the results obtained with this method with the results obtained after using the STFT transformation as proposed in [19].

The next section will focus on presenting this method and a description of each of the stages and components of the training process:

1. The first step in the method is the features extraction. What is actually done is that we take all the utterances that we have in the dataset, and perform on each of them a transformation where its output is a matrix that represents numerical values ​​that represent features in the speaker's voice.

As a part of our research objective, we perform the process of extracting the features using a number of different Wavelet transformations in order to compare the accuracy results of the model with the results after using the Fourier transformation.

1. In the second stage, each batch constructed. Each of these consists of N × M utterances, where n is the number of speakers per batch and m is the number of recordings per speaker. During training, each batch entered into the LSTM network. The LSTM network contains 3 layers with projection. We use 768 hidden nodes with a projection size of 256. Then, we perform L2 Normalization on each output vector from the network.

Where eji represents the embedding vector of the jth speaker’s ith utterance.

1. In the third step, the centroid vector representing each speaker is calculated.

This is a vector that represents the average of all the embeddings of that speaker. The formula for the calculation is described below.

In the process we take all the vectors we got from step 2 and sum up all the vectors that represent a particular speaker and divide by the number of utterances that each speaker has.

1. In the next step, after we calculated the centroids that represent each speaker, we will build a similarity matrix whose number of rows is the number of utterances we have in total (N × M) and the number of columns is the number of centroids we have (equal to the number of speakers).

The values in each row in the matrix represent the measure of similarity between the eij embedding vector to all the centroids and this is calculated according to the following formula:

Where W and b are learned parameters and the cos function calculates the cosine similarity between the input vectors.

As we can see we have here two cases. One, when we want to calculate similarity values in rows in which a comparison is made between a speaker and the centroid belonging to his class, in this case we will use the first formula and the calculation of the centroid will be as follows:

As we can see, all of the embeddings except the embedding that represent the ith utterance of the j speaker are summed and divided by the number of utterances that each speaker has minus one.

In the rest of the cases, we will use the second formula and Equation x to calculate the centroid.

1. Calculation of Softmax Loss function:

1. In the sixth step, the total loss of GE2E is the sum of the losses in the matrix of similarity:

The following figure demonstrates the architecture of the model:

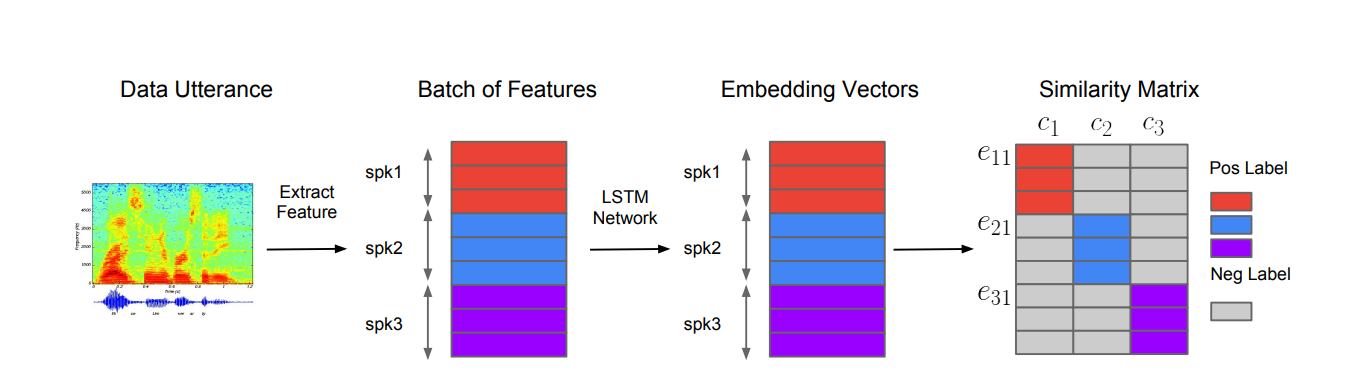


Figure 12

**5.2 Experiments:**

In order to train the model, we used the dataset of TIMIT, which contains utterances of 670 different speakers.

First, we performed the training process using Short Time Fourier Transform in order to obtain features from the utterances. Then, we used 4 different Wavelet Transformation types. The following section shows the results of the loss after training the model using these 5 different transformations. The graph shows the total training loss of 5 different models after 800 epochs each one of them. The tested Transforms are STFT (Short Time Fourier Transform), sym2 (Symlet 2), dmey (Discrete Meyer (FIR Approximation)), db8 (Daubechies8) and coif12 (Coiflet 12).

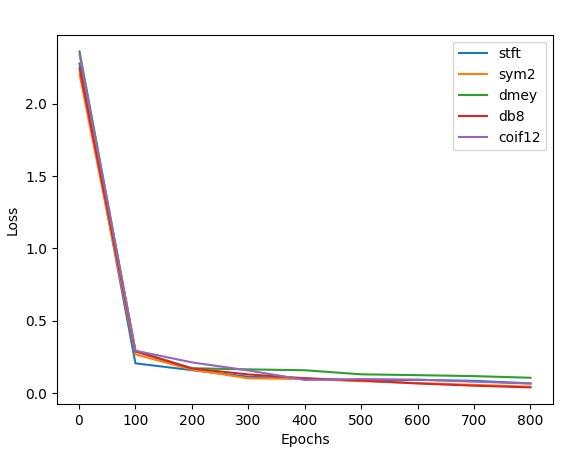


Figure 13

The following table summarizes all the loss values of each transform after the training:

|  |  |  |
| --- | --- | --- |
| Loss | Epochs | Type |
| 0.0673 | 800 | STFT |
| 0.0448 | 800 | Sym2 |
| 0.1063 | 800 | Dmey |
| 0.0396 | 800 | db8 |
| 0.0658 | 800 | Coif12 |

Table 1

As we can see from the graph and the table, db8 acheives the lowest loss value, where 3 wavelet types achieve lower loss values than the STFT.

**5.3 Product:**

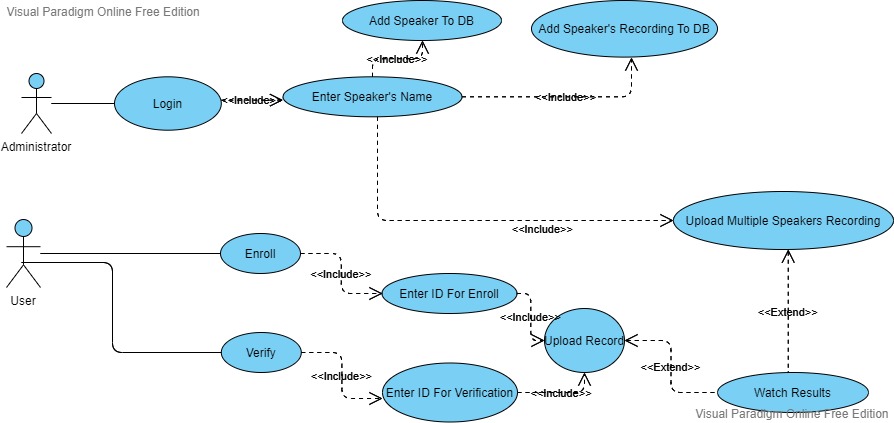
The system has two main uses, one is that given a multi-speakers recordings, recognizes the times in which a particular speaker participates.

The second use is for speaker verification. When a certain person can enter a particular recording of one speaker as a part of the enrollment phase and in the verification phase will enter another recording of one speaker and the system will recognize if it is the same speaker in both recordings. The following section contains various models and Graphical User Interface screens for product description and characterization.

* + 1. **Use Case Diagram:**

The following Use Case diagram is designed to model system usage scenarios designed to provide speaker verification**:**

Figure 14



**5.3.2 Package Diagram:**

The following Class diagram shows in a concise way all the entities in the system, their attributes and their relations.

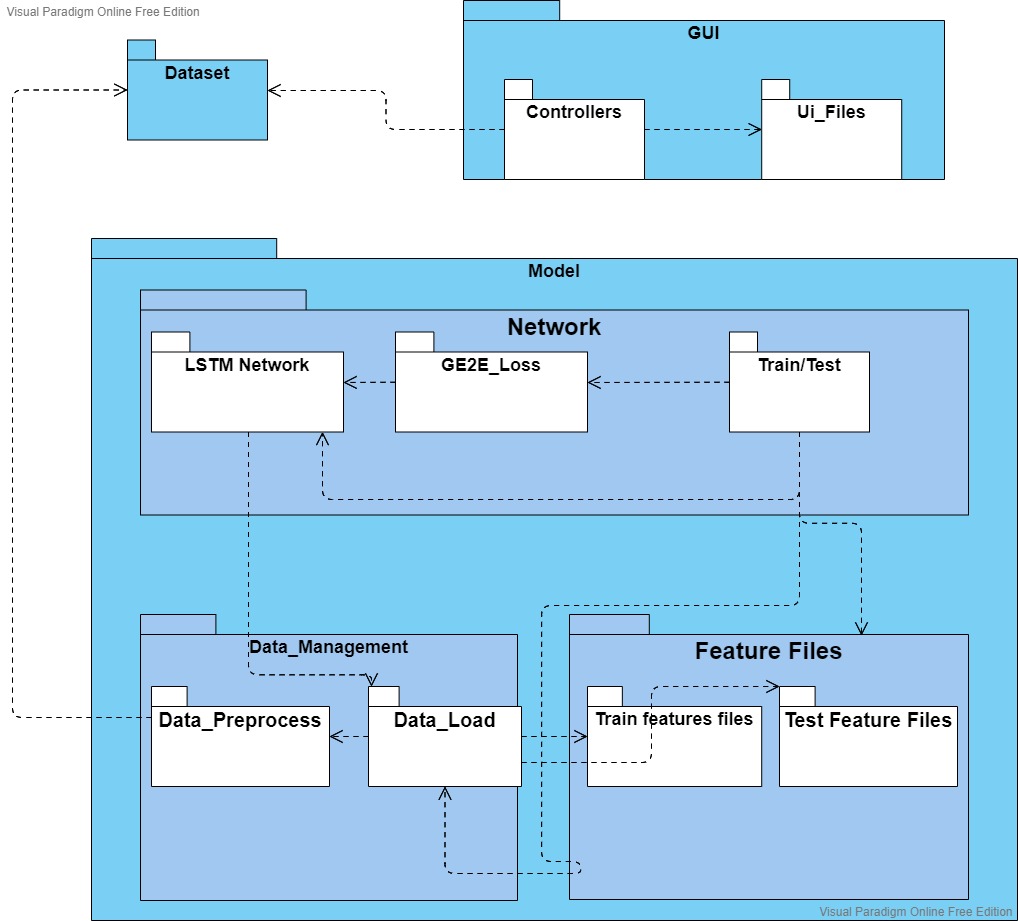


Figure 15

**5.3.3 User interface**

Main Window:

In the main window appears few options:

* Log in as administrator.
* Perform enrollment.
* Perform speaker verification.

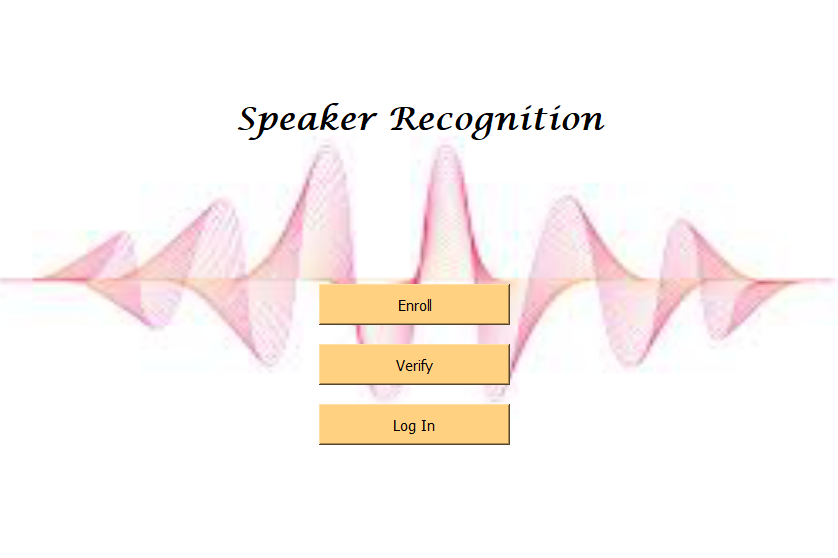


Figure 16

Login:

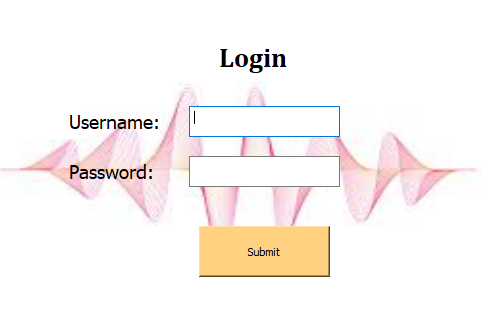
In the Login screen the user has to enter his username and password in order to continue as administrator.

Figure 17

Administrator menu:

After the user enters his username and password and the system identifies him, the "Administrator Menu" will be shown to him.

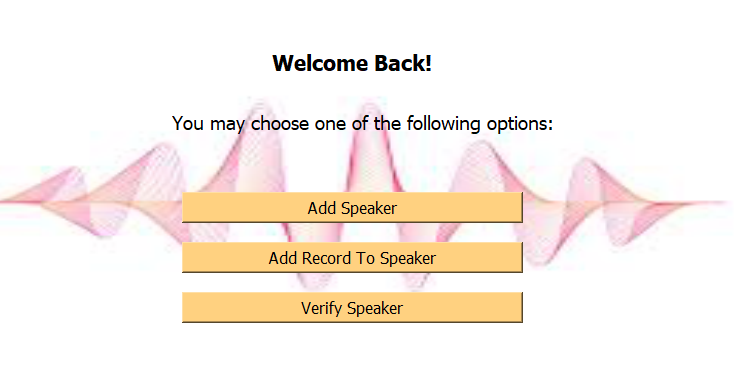


Figure 18

The administrator has three options:

1. Add new speaker to the database.
2. Add new recording to a speaker that saved in the database.
3. Perform verification for a person that is stored in the database.

In the following screen the administrator enters the name of the speaker that he wants to verify:

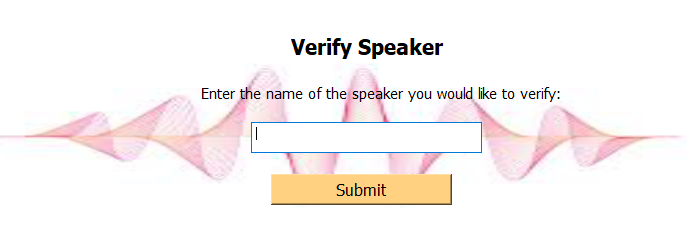


Figure 19

After the user enters a name of speaker that he wants to verify, he uploades a recording of a multi (or single) speakers recording and the system detects the time sections in which the speaker participates:

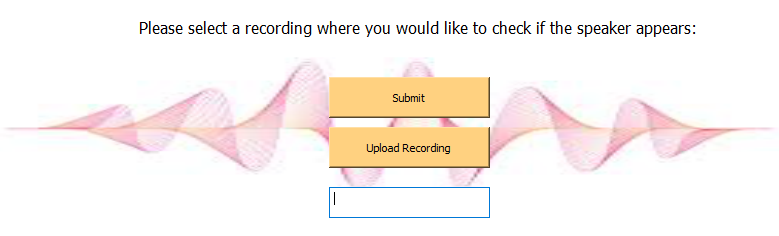


Figure 20

After the administrator uploads the recording to the system, the user waits a while and finally sees a results graph describing the level of similarity of the requested speaker he is trying to identify, to the speaker identified in the same segment.

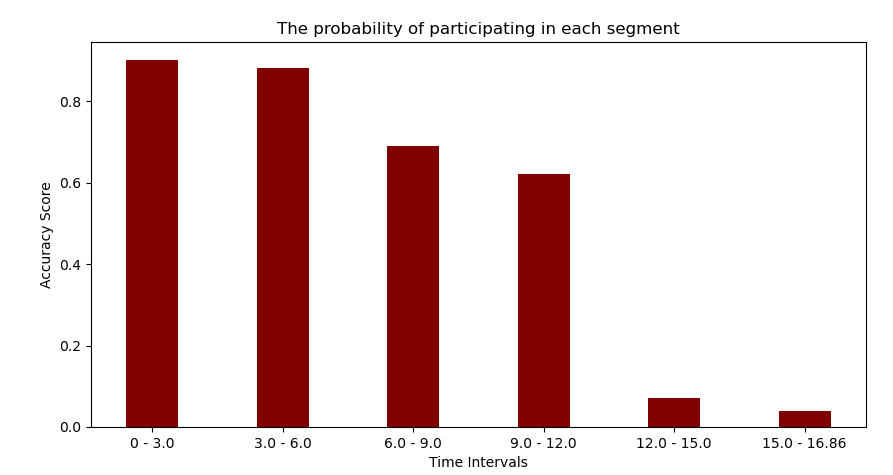


Figure 21

After the user chooses "Enroll" option in menu, he has to enter his id number.

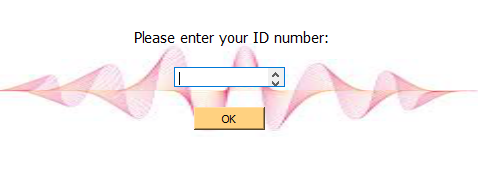


Figure 22

If his number does not appear in the database, he will see the following Enrollment screen:

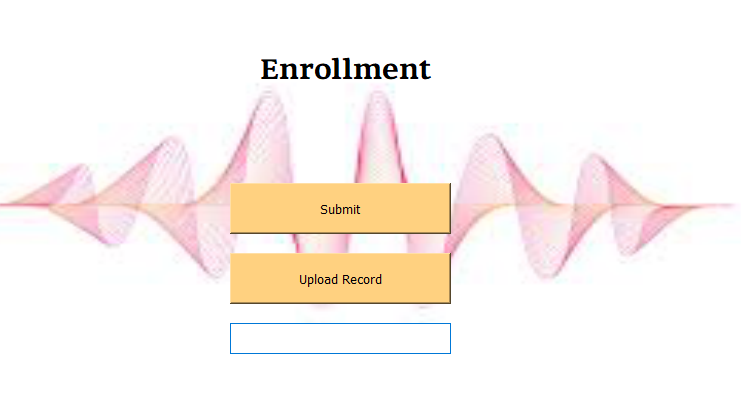


Figure 23

If the enrollment goes successfully, the user gets approval message:

תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטי

Figure 24

After the user chooses "Verify" option in menu, he has to enter his id number.

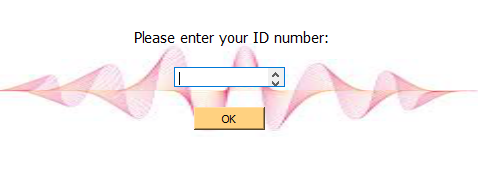


Figure 25

If the user performed enrollment and he did not perform any verification yet, he will see the following Verification screen:

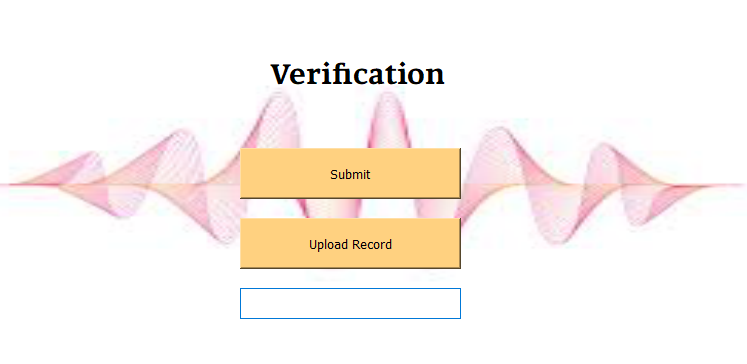
****

Figure 26

If the verification succeeds, the user will see the following message:

תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטי

Figure 27

**6. Evaluation/Verification Plan:**

The following section presents the testing plan that will be implemented in the system.

|  |  |  |
| --- | --- | --- |
| **Expected result** | **Test** | **Test number** |
| "Enrollment" screen will be shown to the user | Click the "Enroll" button  In the Enroll screen | 1 |
| "Verification" screen will be shown to the user | Click the "Verify" button  In the Enroll screen | 2 |
| The system will display a confirmation that the registration was successful | Click the "Submit" button in the Enroll screen | 3 |
| The system will display a message that the verification has failed | Verification failed | 4 |
| "Login" screen will be shown to the user | Click "Login" button | 5 |
| The system will display a confirmation that the verification was successful | Click the "Submit" button in the verification screen | 6 |
| Display error message if username or password is not correct | Entering wrong password or username in Login screen | 7 |
| "Add Speaker" screen will be shown to the user | Click "Add Speaker" from Login | 8 |
| "Add Record To Speaker" screen will be shown to the user | Click "Add Record To Speaker" from Login | 9 |
| "Verify Speaker" screen will be shown to the user | Click "Verify Speaker" from Login | 10 |

Table

All specified tests have passed successfully and the system behaves as expected.

**7. References:**

[1] Ding, Wenhao; He, Liang, “MTGAN: Speaker Verification through Multitasking Triplet Generative Adversarial Networks”, arXiv: 1803.09059, 2018.

[2] L. Tran, X. Yin, and X. Liu, “Representation learning by rotating your faces”, arXiv preprint arXiv:1705.11136, 2017.

[3] E. Variani, X. Lei, E. McDermott, I. L. Moreno, and J. G. Dominguez, “Deep neural networks for small footprint text dependent speaker verification,” in IEEE International C conference on Acoustics, Speech and Signal Processing (ICASSP), Florence, Italy, 2014.

[4] K. Q. Weinberger and L. K. Saul, “Distance metric learning for large margin nearest neighbor classification,” Journal of Machine Learning Research, vol. 10, pp. 207–244, 2009.

[5] D. Cheng, Y. Gong, S. Zhou, J. Wang, and N. Zheng, “End-to-end text-independent speaker verification with triplet loss on short utterances,” in IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, Nevada, USA, 2016.

[6] F. Schroff, D. Kalenichenko, and J. Philbin, “Probabilistic linear discriminant analysis for inferences about identity,” in IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 2015.

[7] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016.

[8]W. Chen, X. Chen, J. Zhang, and K. Huang, “Beyond triplet loss: A deep quadruplet network for person re-identification,” in IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, Hawaii, USA, 2017.

[9]H. Alexander, B. Lucas, and L. Bastian, “In Defense of the Triplet Loss for Person Re-Identification,” arXiv preprint arXiv:1703.07737, 2017.

[10]A. Makhzani, N. J. J. Shlens, I. Goodfellow, and B. Frey, “Adversarial Autoencoders,” arXiv preprint arXiv:1511.05644, 2015

[11] Abdalla, Mahmoud & Abobakr, Haitham & Gaafar, Tamer. (2013). “DWT and MFCCs based Feature Extraction Methods for Isolated Word Recognition”, International Journal of Computer Applications. 69. 21-25. 10.5120/12087-8165.

[12] Mahmoud I. Abdalla and Hanaa S. Ali, “Wavelet Based Mel-Frequency Cepstral Coefficients for Speaker Identification Using Hidden Markov Models“, arXiv preprint arXiv: 1003.5627, 2010.

[13] Haytham M. Fayek , “Speech Processing for Machine Learning: Filter banks, Mel-Frequency Cepstral Coefficients (MFCCs) and What's In-Between“, <https://haythamfayek.com/2016/04/21/speech-processing-for-machine-learning.html>, 2016.

[14] Alim, Sabur & Alang Md Rashid, Nahrul Khair. (2018). “Some Commonly Used Speech Feature Extraction Algorithms. 10.5772/intechopen.80419“.

[15] C. Zhang and K. Koishida1, “End-to-end text-independent speaker verification with triplet loss on short utterances,” in Interspeech, Stockholm, Sweden, 2017.

[16] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, “Front-end factor analysis for speaker verification,” IEEE Transactions on Audio, Speech, and Language Processing, vol. 19, no. 4, pp. 788–798, 2011.

[17] S. J. D. Prince and J. H. Elder, “Probabilistic linear discriminant analysis for inferences about identity,” in International Conference on Computer Vision (ICCV), Rio de Janeiro, Brazil, 2007.

[18] Tiwari, Ritu & Meena, Hemant Kumar & Kala, Rahul. (2008). Speaker Identification using Wavelet Analysis and Artificial Neural Networks. 125-130.

[19] Wan L, Wang Q, Papir A, et al. Generalized end-to-end loss for speaker verification[C]//2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2018: 4879-4883.