Lie Detector

המכללה האקדמית להנדסה

אורט בראודה

ORT BRAUDE COLLEGE

Software Engineering Department

ORT Braude College

Course 61766: Extended Project in Software Engineering

Lie Detection Using Convolutional Neural Network

In Partial Fulfillment of the Requirements for

Final Project in Software Engineering (Course 61771)

Karmiel – February 2019

Elias Nijim 206160624

Vladimir Balagula 323792770

Supervisor:

Prof. Zeev Volkovich

Contents

***Abstract**: There are a lot of studies that are trying to reveal the nature of human lie and connect it to physical and Behavioral Sciences. In recent years, a new method has been found useful for that purpose. This method based on non-invasive physiology sensing such as voice pitch variation analyzed by deep learning model of Convolutional Neural Network. This study presents methods that we used to build voice lie detector. Those methods include MFCC algorithm as preparation input of Convolution Layer. The second step defines the architecture of the neural network and the origin of it. The study also reveals teaching mode for a network with backpropagation algorithm. We explain how convolution works combined with MAXPOOL Layer, and which layers our model contains. Also we provide all the documentation data necessary for the implementation of the program.*

We expect that our Convolutional Neural Network will find different patterns using multiple filters and will predict lie with high accuracy.

***Keywords:** Convolutional Neural Networks, MFCC, Voice Lie Detector, Back Propagation.*

1. INTRODUCTION

Nowadays there are various of studies that are working to explain the nature of lying and trying to attach it to physical and Behavioral Sciences. Those studies use a lot of different techniques but, none of them reach the 100 percent of accuracy.

The best technique available today is the lie detector (Polygraph), which was invented in the early 19th century and has the accuracy of 90 percent in exposing lies [1]. Since the Polygraph measures physiological signals as blood pressure, pulse and sweat a person can be trained to lie without all of these symptoms. [2].

In recent years, a new method has been developed to identify lies. This method based on non-invasive physiology sensing such as voice pitch variation and heart rate. All the data were analyzed using a neural network. The researchers were able to bring about 82% accuracy of identifying high-stress situations and 71% accuracy of lying detection, which is lower than the Polygraph. [3]

In our project, we present a new approach to identify a lie. We used a convolutional neural network (CNN); this method gave the option to analyze the voice deeper.

2. BACKGROUND

2.1. Sound

Sound is a type of energy that travels in waves through a medium as air or water. The waves are generated by vibration source which pushes particles of air together and creates a pressure wave. Sound waves move out from the vibration source to all directions. This type of wave is referred to as a longitudinal wave. Though, there are two types of waves (Longitudinal and transverse), gas (the air) can sustain only longitudinal waves because transverse waves require a shear force to maintain them, in this case light is an example of a transverse wave, while sound is a longitudinal wave. When the sound wave reaches the ear, it transforms from longitudinal wave into an electrical impulse which are sent into the brain

There are two main elements which characterize vibration and the way it sounds - the amplitude and the frequency.

Amplitude is the size of the vibration, in other words, the height of the wave determines how loud the sound will be. Amplitude is important when balancing and controlling the loudness of sounds, a larger amplitude will produce a louder sound, and moderate amplitude will perform quite tones.

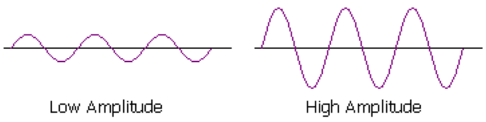

Figure 1 -Low and High Amplitude

Frequency is the speed of the vibration, and this determines the pitch of the sound and determined by the wavelength. Frequency is measured as the number of wave cycles that occur in one second. A cycle is a repeating pattern in given amount of time – a period.

The unit of frequency measurement is Hertz (Hz for short). (The hertz unit is named after Heinrich Hertz, a famous 19th century physicist.). [4]

Pitch of the sound is also determined by the frequency of the wave but from different viewpoints. While frequency measures the cycle rate of the physical waveform, pitch is how high or low it sounds when you hear it. High pitch is generated by high frequency wave and low-pitch generated by low frequency wave.

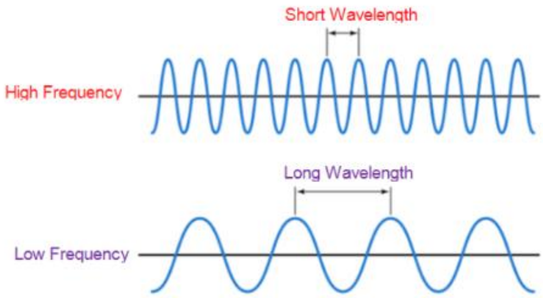

Figure 2 - High/Low Frequency Wave

http://www.techplayon.com/wavelength-frequency-amplitude-phase-defining-waves/

2.2. Analog Signal

To process and represent sound wave into a digital signal in the computer it must be converted into an analog signal by using one of the analogs to digital converter.

Analog signal is a continuous wave that changes over time; it can be used to measure different types of waves such as sound and temperature. One kind of analog signal is "sin wave" which cannot be broken. The most common way to convert an analog signal into a digital signal is via a microphone.

2.3. Digital Signal

Digital signal is a non-continuous signal that was converted into a pattern of bits to represent data as a sequence of discrete values. Digital signal input can be represented in Time-Frequency Domain graph where the Y-axis describes the amplitude and the X-axis describes the time (Figure 3). [5]

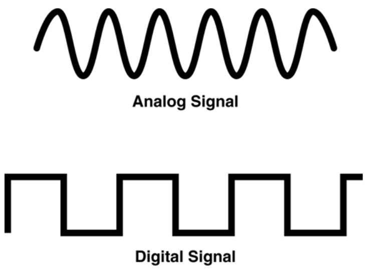

Figure 3 - Analog Signal compares to Digital Signal

https://www.allaboutcircuits.com/technical-articles/an-introduction-to-digital-signal-processing/

To do a deeper analysis of the time domain signal we can use the frequency domain. The frequency domain signal can be analyzed with the use of spectrum analyzer. In addition, time domain signal can be transformed to frequency domain signal by calculating Fast Fourier Transform (FFT).

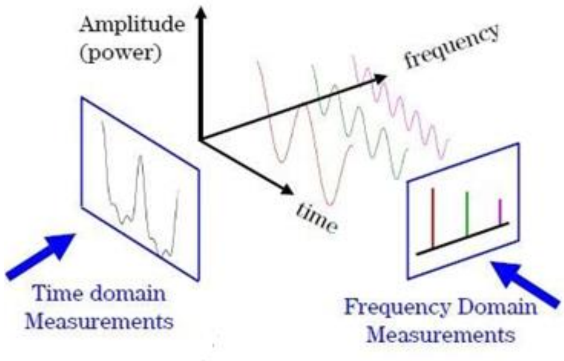

Figure 4 - depicts time domain and frequency domain measurements

http://www.test-and-measurement-world.com/Measurements/Time-domain-measurements-vs-Frequency-domain-measurements.html

2.4. MFCC

Mel Frequency Cepstral Coefficient (MFCC) is commonly used in the speech recognition process and in the analyzing speech signal. The pre-step in the speech recognition system is to identify the components of the signal [6].

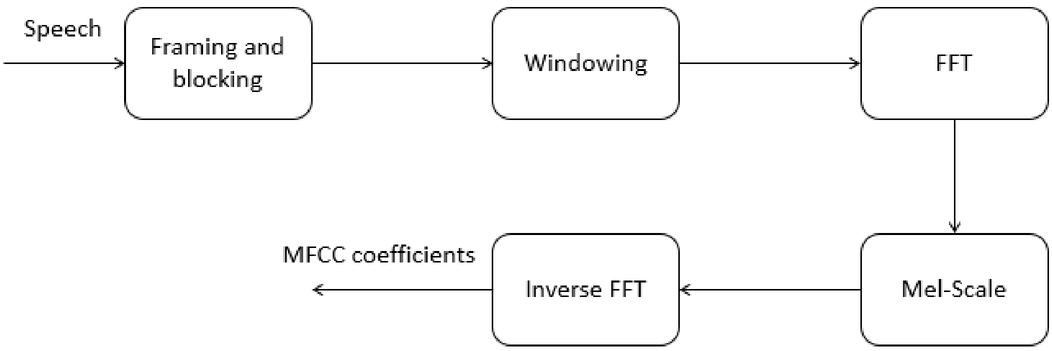

Figure 5 - MFCC Processing

2.4.1. *Framing and blocking*

The first step of MFCC procedure is to divide the signal into continuous smaller frames by dividing the speech signal to “X” signal frames. To reach the best results, we must find the best divider number that contains enough information, many researchers found that the common divider number that divides the speech signal to the adequate number of frames is 256 frames, this number of frames is enough to include all the data which gives the most accurate result, increasing or decreasing the divider number can manipulate with the data amount in the frames which can lead to reducing the results accuracy.

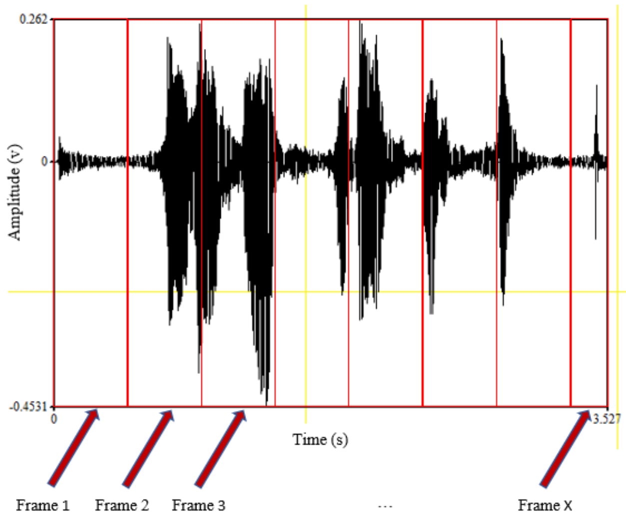

Figure 6 - Framing

2.4.2. *Windowing*

The purpose of windowing is minimizing the disruptions at the beginning and at the end of a certain frame and its function is multiplied as well as the window is one, the multiplication result of the signal in window gives a signal sample, The appropriate window function that brings the best results is Hamming Window (Equation1).

$$\omega(n) = 0.54 - 0.46\cos(\frac{2\pi n}{N-1})$$

Equation 1 - Hamming Window

N = number of blocks in each signal frame.

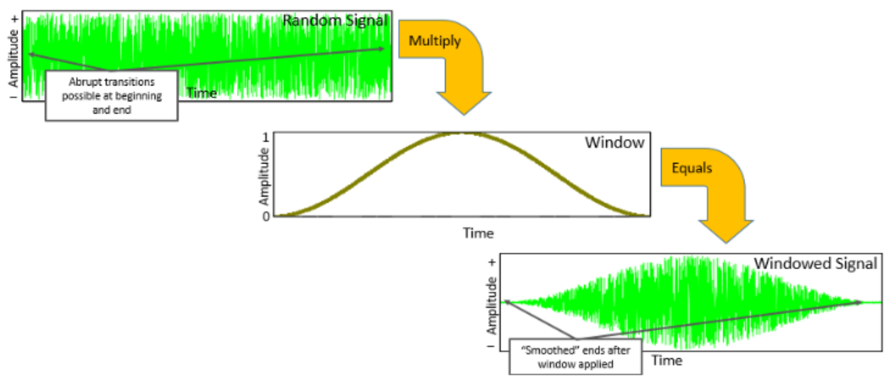

Figure 7 - Hamming window

https://community.plm.automation.siemens.com/t5/Testing-Knowledge-Base/Window-Types-Hanning-Flattop-Uniform-Tukey-and-Exponential/ta-p/445063

2.4.3. Fast Fourier Transform – FFT

Fast Fourier Transform is a rapid computation algorithm to convert time domain signal wave samples to frequency domain spectrum samples (Equation 2).

$$X_k = \sum_{n=0}^{N-1} x_n e^{\frac{-i2\pi kn}{N}}$$

Equation 2 - FFT formula

X_k - Frequency domain sample x_n – Time domain sample N – FFT size K = {0, 1, 2…N-1}

These transformations provide the location of the speech wave signal in time domain graph by breaking the complicated signal into small sin waves and express it as picks on another graph.

2.4.4. Mel scale

The linear scale is not pursued by the human perception of frequency contents of sounds for speech signal. The Mel scale frequency is a linear frequency and a scale of pitches, it is the way of spacing the filter and calculating how much wider it should be, consequently when the frequency gets higher the filters get wider as well. A popular formula to convert hertz into Mel (Equation 3)

$$m = 2595\log_{10}(1 + \frac{f}{700})$$

Equation 3 - Hertz to Mel formula

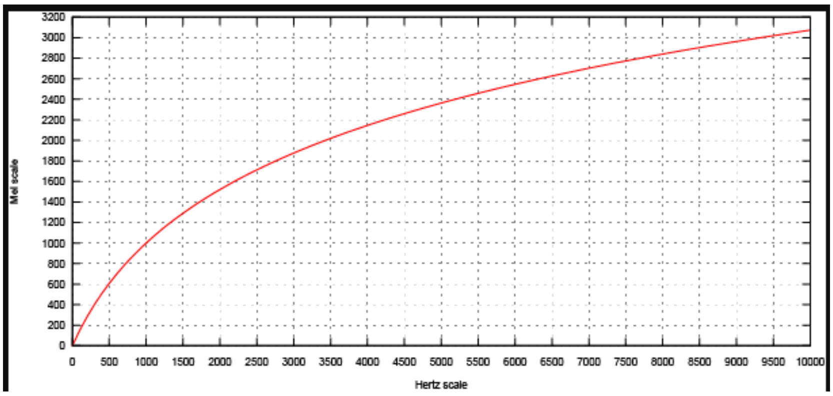

Figure 8 - Mel and Hertz scale

https://cpb-us-w2.wpmucdn.com/u.osu.edu/dist/3/41388/files/2015/02/Mel-Hz_plot-2juu180.png

Mel spectrum is generated from result of the log calculation on the energies at each block. The popular usage of Mel scale is to calculate coefficients

2.4.5. Inverse Fast Fourier Transform – IFFT

Inverse fast Fourier transform is a fast algorithm to perform backward of Fourier transform, which revokes the process of fast Fourier transform. IFFT converts frequency domain wave signal into its original time domain wave signal using IIFT formula (Equation 4)

$$X_n = \frac{1}{N}\sum_{n=0}^{N-1} x_k e^{\frac{-i2\pi kn}{N}}$$

Equation 4 - IFFT formula

X_k - Frequency domain sample x_n – Time domain sample N – FFT size K ={0,1,2...N-1}.

2.5. Neural Network

2.5.1. Biological Neural Network

The neuronal system is a group of nerve cells that divide into a central nervous system (the spine and brain) and peripheral (which is divided into a motor and a sensor). Neurons are nerve cells responsible for passing the information along the human body; each neuron is composed of Dendrites and one Oxon.

The Dendrites are branches on the cell body that initiate information transfer by intake chemical signal from outside the cell and convert them to an electrical signal which passes through the Oxon. At the end of the Oxon, there are neurotransmitter molecules that packed in vesicles, waiting for the electrical process reaching the end of the axon. When it does, there is an increase in the concentration of intracellular calcium, which increases the movement of the full neurotransmitters to the exocytosis (the exit of the chemicals out of the cell). The strength of the signal works according to the principle of "the more neurotransmitters there are, the signal is more powerful". The external neurotransmitters are injected into the synaptic space and are absorbed by specific receptors on the membrane of the next neuron or the target cell to pass on the information. [7]

2.5.2. Artificial Neural Network

Artificial Neural network – is a mathematical model also known as a neural network which based on biological neuron system and is part of artificial intelligence discipline. The main idea of artificial neuron is taken from the biological neuron. The artificial network has same properties as the biological system including modifying signals by weights at the receiving synapse, summing the weighted inputs and transmitting outputs to other neurons.

The main advantage of the artificial network is the ability to improve itself by learning from the mistakes. The learning phase is inserting data with mapped output, which gives the system options of finding relation coefficients between neurons inputs. [7]

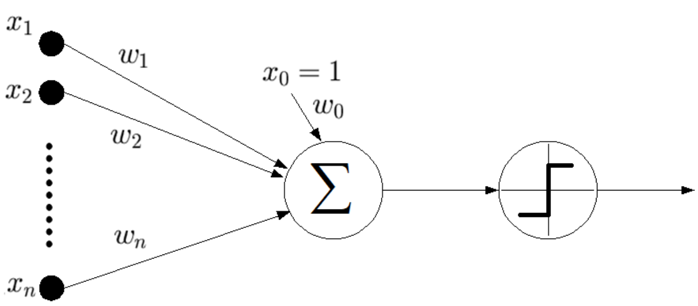

Figure 9 - Diagram of an Artificial Neuron

https://www.funglobalretailtech.com/research/introduction-deep-learning-neural-networks/

$$v_j = \sum_{i=1}^{m} w_{ji}^T y_i$$

Equation 5 - Neuron input

$$y_j = f(v_j)$$

Equation 6 - The neuron output

The w_i on the diagram is a weight for each input for each neuron. All the weights are summarized and converted to a specific output using activation function. The whole network contains a big number of neurons which communicate one with the other by sending and receiving outputs. A large number of neurons can solve complex tasks.

2.5.3. Back-propagation

In order to complete the goal tasks of the neural network which includes processing the input from convolutional layer into an accurate output, the network must adjust weights on each node. To achieve this, in our project we use backpropagation algorithm based on gradient descent. The algorithm calculates the error between produced output and desired output and corrects it to increase the accuracy. [8]

The mathematics stands at the base of this algorithm take in a matter of relative errors (Equation 7) and instantaneous error energy of neuron (Equation 8). Where "d" is the expected result and "y" is neural network result

$$e_i(n) = d_i - y_i$$

Equation 7 - Relative Error

$$E_i(n) = \frac{1}{2} e^2(n)$$

Equation 8 - Error Energy of Neuron

Partial derivation of the above by using chain rules gives (Equation 9)

$$\frac{dE(n)}{dw_{ij}} = \frac{dE(n)}{de_j(n)} \frac{de_j(n)}{dy_j(n)} \frac{dy_j(n)}{dv_j(n)} \frac{dv_j(n)}{dw_{ji}(n)}$$

Equation9 - Pertial Deviation

Positioning derivations of Equation (8) by e(n), derivation of Equation (7) by y(n), derivation of Equation (6) by v(n) and derivation of Equation (5) by w(n) in the Equation (8) will gives Equation (10)

$$\frac{dE(n)}{dw_{ij}} = -e_j(n) f_j' \left(v_j(n)\right) y(n)$$

Equation 10 - Derivation Error Energy by weight

Correction of the weights defined by delta rule (formula 7) where "η" is a learning rate.

$$\Delta w_{ij}(n) = -\eta \frac{dE(n)}{dw_{ij}}$$

Equation 11 - Weight correction by delta rule

The negative value of η indicates a direction for gradient descent to minimize the error. Using the equations (10) and (11) we define that delta rule depends on learning rate, local gradient δ and y_j (Equation 12).

$$\Delta w_{ij} = \eta \delta y_j(n)$$

Equation 12 Weight correction

2.5.4. Dropout and regularization

Dropout method in NN lies on the fact that in the process of learning a layer is selected from which a certain number of neurons are randomly ejected, which are turned off from further calculations. This technique improves the effectiveness of training and the quality of the result. More trained neurons gain more weight on the network.

Another method is regularization it works by applying some additional information to the loss value to prevent retraining model. The penalty value has been calculate using formula(Equation 13) when n is the number of layers $w^{[j]}$ the weight matrix for the layer j^{th}, m the number of inputs and λ is the regularization parameter.

$$x = \left(\sum_{j=1}^{n} \left|\left| w^{[j]} \right|\right|^2 \right) \frac{\lambda}{2m}$$

Equation 13 - Regularization penalty value

2.5.5. Convolution

The main idea of the convolutional layer is to derive meaningful information using pattern filters on the input.

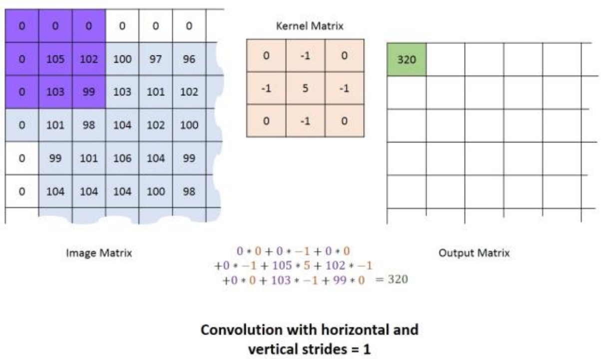

Figure 10 - Filter Scanning

http://machinelearninguru.com/computer_vision/basics/convolution/convolution_layer.html

The input to the Convolution layer is two-dimension matrix. The convolution use different filters to extract various features. The filter is a smaller matrix which moves through the image, multiply each pixel value with the filter on it and summarize it. The result is smaller image with the new weight.

In the polling level (MAXPOOL layer), new matrix is built with maximum values from every filtered cell. The process is repeated multiple times depending on the user's definition. [9]

2.6. Architecture

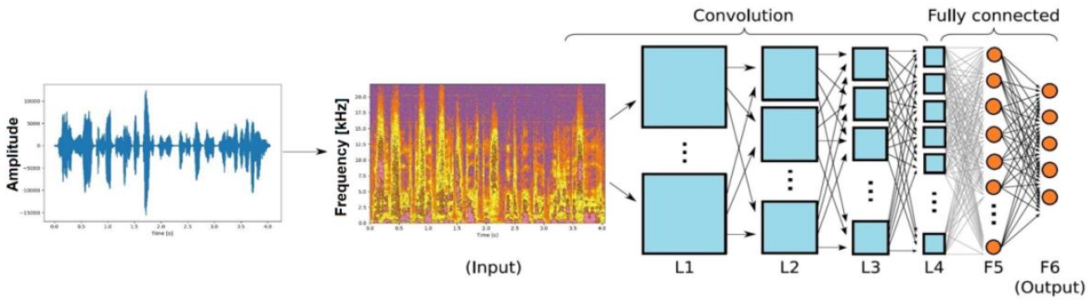

Figure 11 - Convolutional Neural Network

In this project, the user was given an option to record a voice in a WAV format by using a microphone or upload a WAV file. The next step, MFCC obtained from the multi-spectrogram as a table of real values. The MFCC coefficients are the convolution layer input. We choose the first 225 lines with 32-real values in a CSV file. The data for the network was a list of arrays 225x32 with real rates, which analyzed as an image and was normalized with a range from 0 to 255.

We built our CNN model using VGG16 convolution layers which were attached to fully connected layers with RELU as the activation function. One of the benefits of RELU (Equation 13) is a fast calculation and thus has much better performance than other activation functions.

In our model, there were five convolutional blocks with pre-trained weights from "ImageNet." The next layer after the blocks was the Flatten layer which converted the two-dimensional output into one-dimensional input and transferred it to a fully connected layer.

$$f(x) = max(0, x)$$

Equation 14 - ReLu function

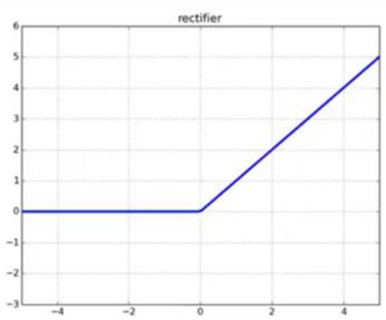

Figure 12- ReLu function graph

https://towardsdatascience.com/activation-functions-in-neural-networks-58115cda9c96

The two following levels consisted of a fully connected layer with 1000 neurons followed by the dropout layer at the end of each of them. The whole system concealed by lie detection classifier layer. The classifier layer used a SOFTMAX activation function (Equation 14) to display the probability of lying.

The multiple parameters were chosen when model was trained for given better results as optimizer Adam with learning rate value 0.00005, batch size 10, epoch number 45 and number of MFCC coefficients is 32.

Our CNN implemented using Python and TensorFlow as the back-end with KERAS library and PyQt as a user interface with style sheet design. KERAS is a library created especially for "deep learning” for research and developing. TensorFlow is an open source library for dataflow programming and computation.

$$\mathrm{out}_j = \frac{e^{x_j^T w_j}}{\sum_{i=1}^{k} e^{x_i^T w_i}}$$

15 - SoftMax function

The final model included an architecture that was a combination of VGG16 and the following dense layers:

1. Convolutional Block #1
    a. Convolutional 2D Layer
    b. Convolutional 2D Layer
    c. MaxPool Layer
2. Convolutional Block #2
    a. Convolutional 2D Layer
    b. Convolutional 2D Layer
    c. MaxPool Layer
3. Convolutional Block #3
    a. Convolutional 2D Layer
    b. Convolutional 2D Layer
    c. MaxPool Layer
4. Convolutional Block #4
    a. Convolutional 2D Layer
    b. Convolutional 2D Layer
    c. Convolutional 2D Layer
    d. MaxPool Layer
5. Convolutional Block #5
    a. Convolutional 2D Layer
    b. Convolutional 2D Layer
    c. Convolutional 2D Layer
    d. MaxPool Layer
6. Flatten Layer
7. Fully connected Layer with rectifier activation function (RELU)
8. Dropout Layer
9. Fully connected Layer with SOFTMAX activation function

3. SOFTWARE ENGINEERING DOCUMENTATION.

3.1. Use Case

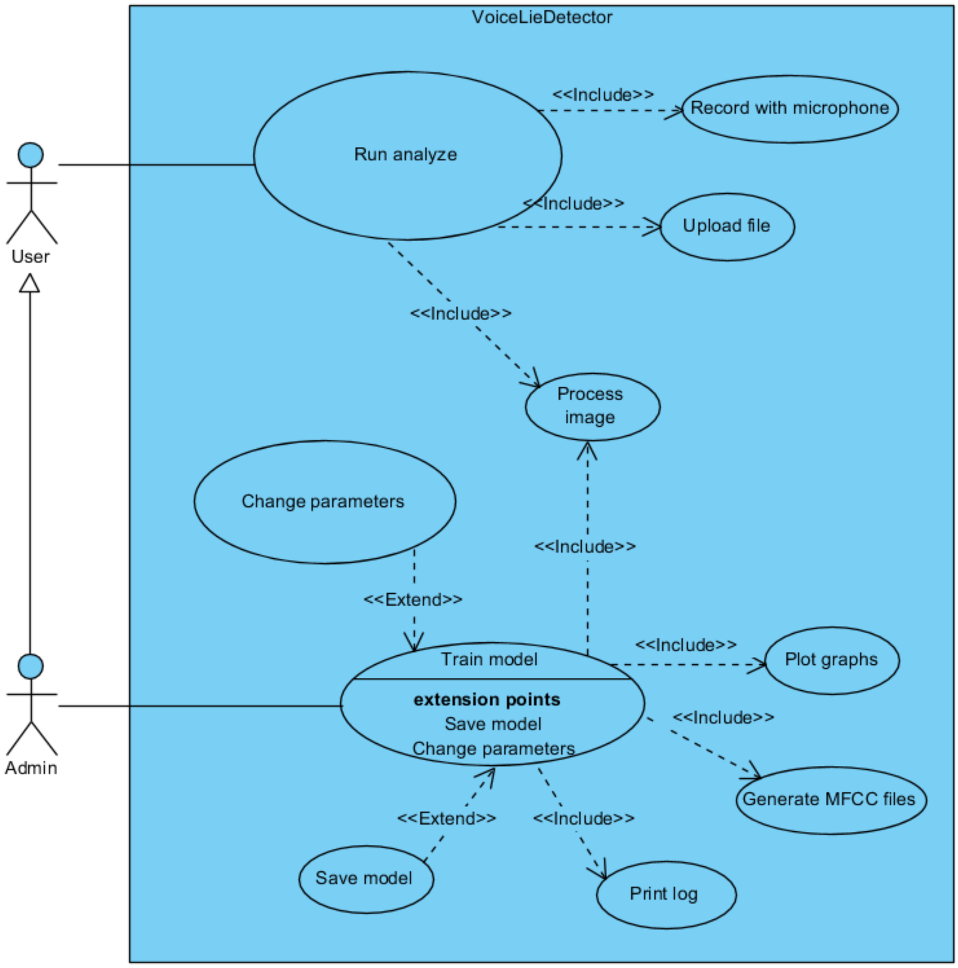

Figure 13- Lie Detector use case diagram

UC1: Run analyze

- Goal – run the model on a recorded voice to get if the person is lying.
- Preconditions – learning model and voice file exist.
- Possible errors – microphone is off.
- Limitations – voice record length.
- Pseudo code:

Actor	System
Click on microphone button	Show processing gif and change image of microphone
Click on stop recording	Microphone image has been changed and application created graphs of the voice with option to choose on which model to run the prediction
Click on predict	Insert voice file to the CNN
	Display an answer of the prediction model

Table 1- Run analyze use case

UC2: Train model with different settings

- Goal – train, set up and validate the CNN.
- Preconditions – dataset exists in \\db\\wav folder, labeling to dataset is correct.
- Possible errors – data set is corrupted.
- Limitation – processing time and CPU.
- Pseudo code:

Actor	**System**
Click on "Admin Console"	Open setting window
Set up batch size, learning rate, epoch number, optimizer, number of features and percentage of data to use in training phase	
Click start "Start"	Start training
	Display monitoring window
	Finish training display complete message and ask for store the model
Save model	Model saved in file system

Table 2 - Train model with different settings

3.2. Class Diagram

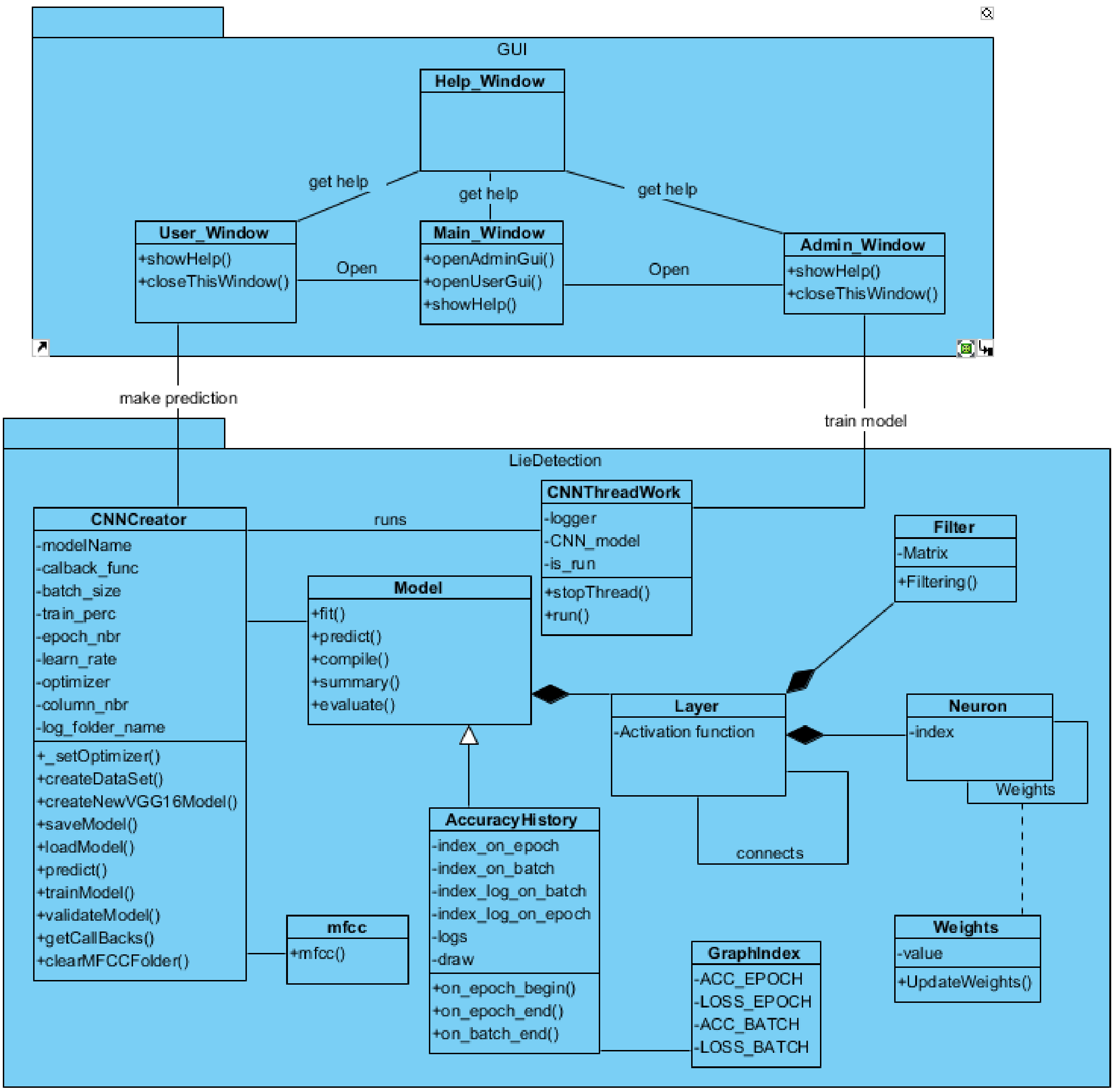

Figure 14 - Class Diagram

3.3. User Interface

3.3.1. Windows

Welcome window: the first displayed window once the program starts, gives the user opportunity to configure the model or to detect lie using existing model.

Figure 15 – Welcome window

Settings: Admin is capable to configure the model by setting new parameters and giving different data sources, in addition, the admin can monitor the model process.

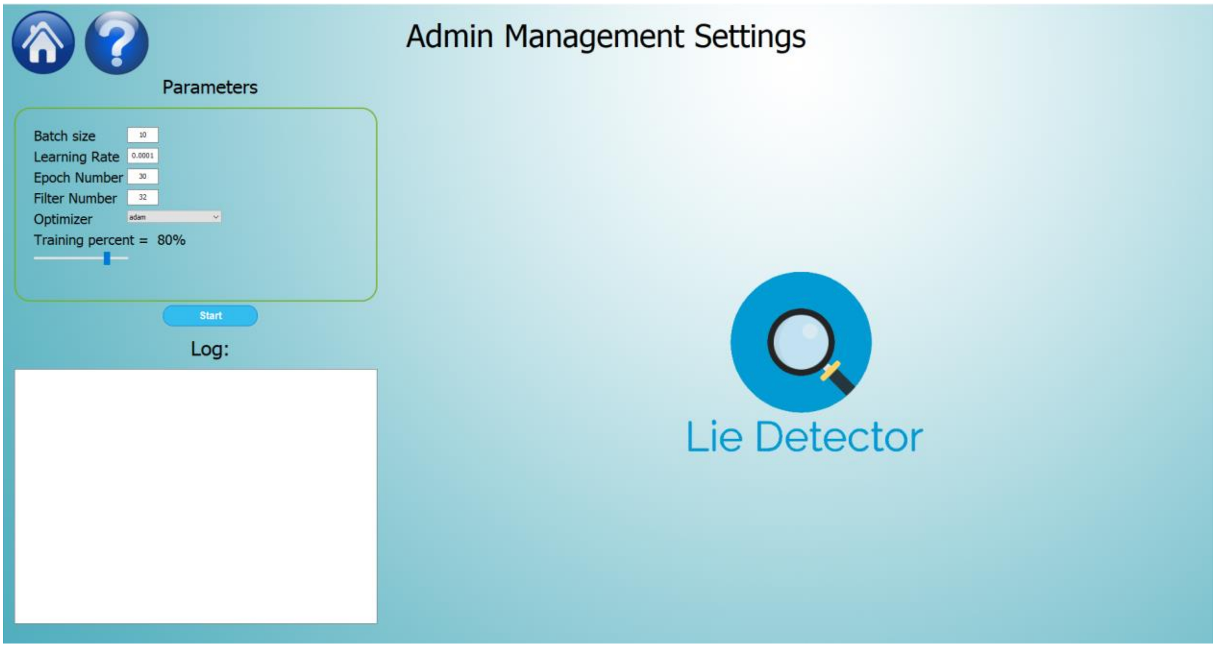

Figure 16 - Settings Window

After finishing learning the system asks for saving the model.

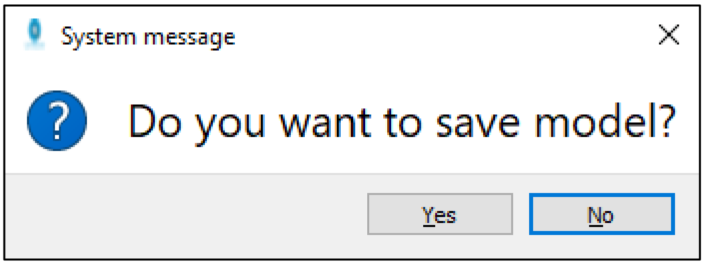

Figure 17 - Save Model Window

Detect Lie: User can choose to upload local .wav or to record voice using microphone

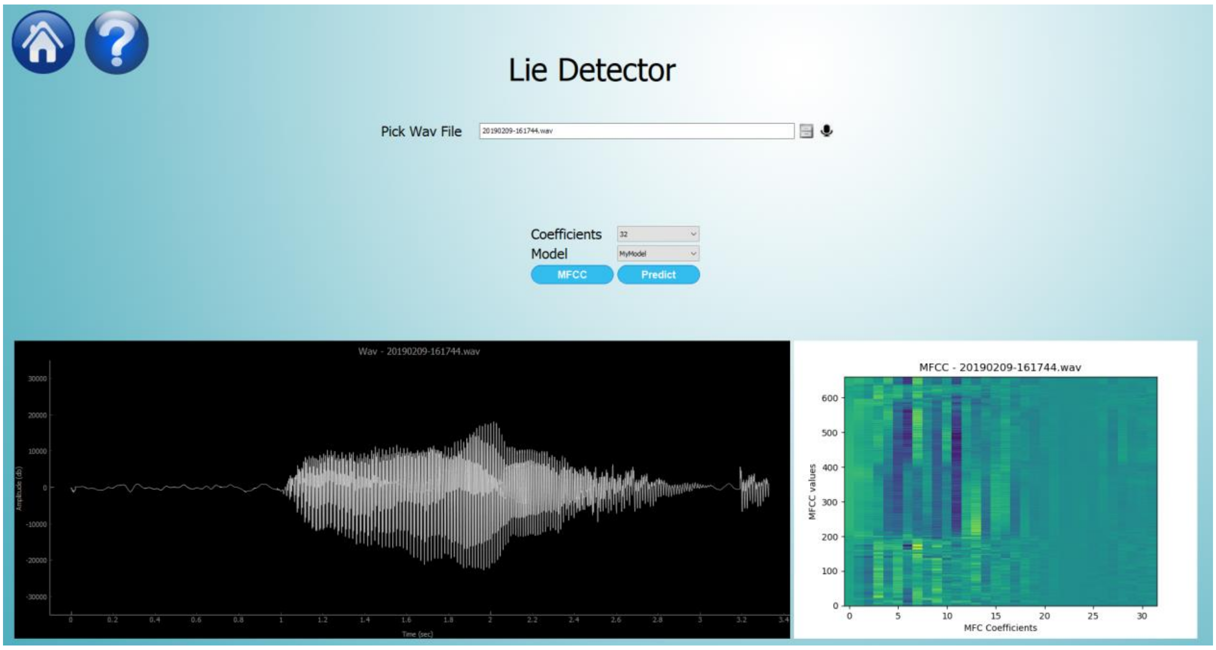

Figure 18 - Prediction Window

Monitor: User can see the results of learning process in real time.

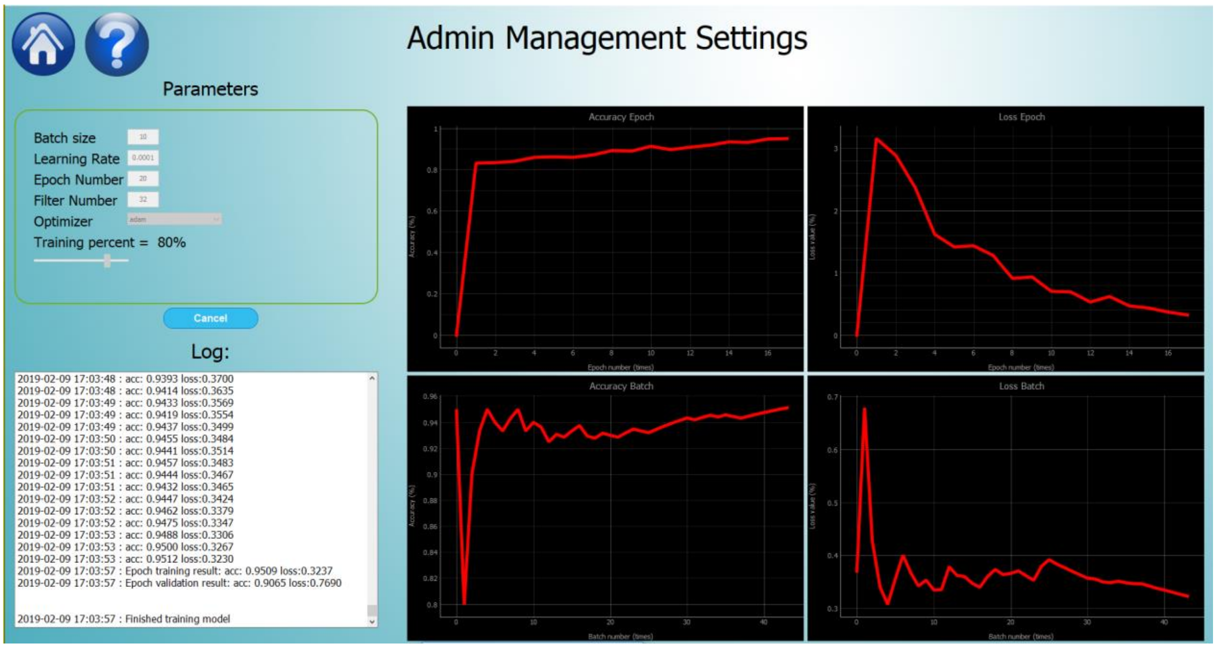

Figure 19 - Monitoring Window

3.3.2. Errors

If batch size is not a positive number:

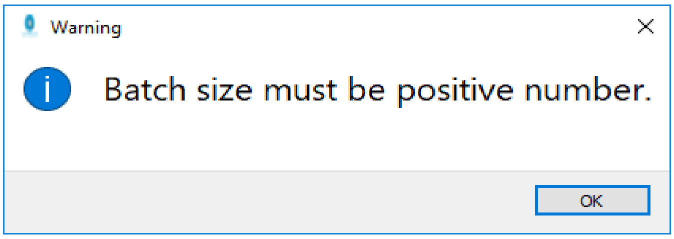

Figure 20 – Wrong batch size error

If the user chooses incorrect value for learning rate:

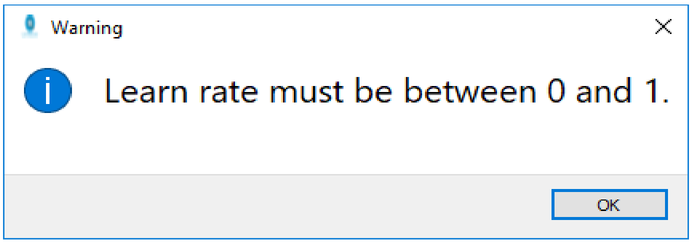

Figure 21 – Learning rate error

If user insert wrong epoch number:

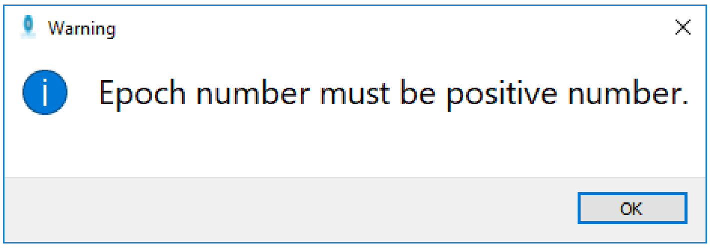

Figure 22 - Wrong epoch number error

If number of MFCC features is incorrect:

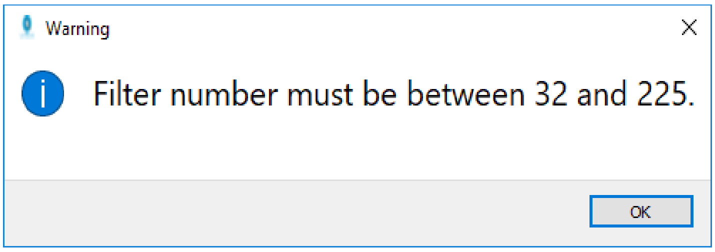

Figure 23 - Invalid MFCC coefficients number

3.4. Testing Plan

Test name	Description	Expected result	Actual Result	Comments
Math MFCC				
MFCC validation	Execute function using the same data in MATLAB and in the project function	The same result with small error	The same result with small error	
Record voice using microphone				
Microphone	Check if microphone connected while clicking on microphone to record	Microphone starting recording if connected	Microphone starting recording if connected	
Start recording	Click on microphone	Image of the microphone has been changed and processing gif appeared	Starting record user	Microphone connected

Test name	Description	Expected result	Actual Result	Comments
Stop recording	Click disabled microphone	Image of the microphone has been changed and processing gif disappeared	Stop recording and save the file to the record folder	
Display voice graph	1) Finish recording/Uploading file	1) Display sound wave plot 2) Display MFCC plot	1) Display sound wave plot 2) Display MFCC plot	
Change MFCC features amount	1) Choose data source as voice record 2) Change number of MFCC coefficients	Displaying new plot with chosen coefficients	Displaying new plot with chosen coefficients	
GUI				
Help	Click on help button	Open help window with explanation	Window opens	
Add file to predict	Choose voice file option	File explorer opens to give an option to choose file	File uploaded and graphs created	
Choosing different number of features	1)change number of MFCC coefficients 2) press detect	Application change the required number of coefficients for the model	Displaying information window with message that coefficient amount has been changed	

Test name	Description	Expected result	Actual Result	Comments
Display monitoring at Learning phase	1)Click on start button at admin window	Opens Monitoring graphs with accuracy prediction and loss value of each epoch and batch	Pass	
Display save option on settings	When finished validating asking for store the model	Pop-up window asking for storing model	Model stored	while the system finished learning phase.

Table 3 – Testing Plan

4. RESULTS AND CONCLUSION

4.1. Results

4.1.1. Experiment 1 German database with single dense layer

	Positive	Negative
Lie	0	14
Telling a truth	0	93

Table 4 –confusion matrix of the validation results in German (VGG16 model with one fully-connected Layer)

$$Accuracy = 86\%$$

$$Recall = 0$$

$$Precision = 0$$

In this experiment, we ran 535 records in German. Sixty-nine of the records classified as liars. Eighty percent of the records were taken to the learning process and the other for the validation. According to table 4, 107 records were scanned through the validation process, and the results were: 0 cases identified as telling a lie (true positive) while 14 of the liars was not identified for their lying and was a false negative. On the other hand, on a group of truth-telling, 0 cases identified as false positive and 93 revealed true positive answer. The accuracy of the experiment was 86% the recall, the number of correctly classified positive examples divided by the total number of actual positive samples in the test set was 0 and is the number of correctly classified positive cases divided by the total number of examples that are classified as positive was 0.

4.1.2. Experiment 2 German database

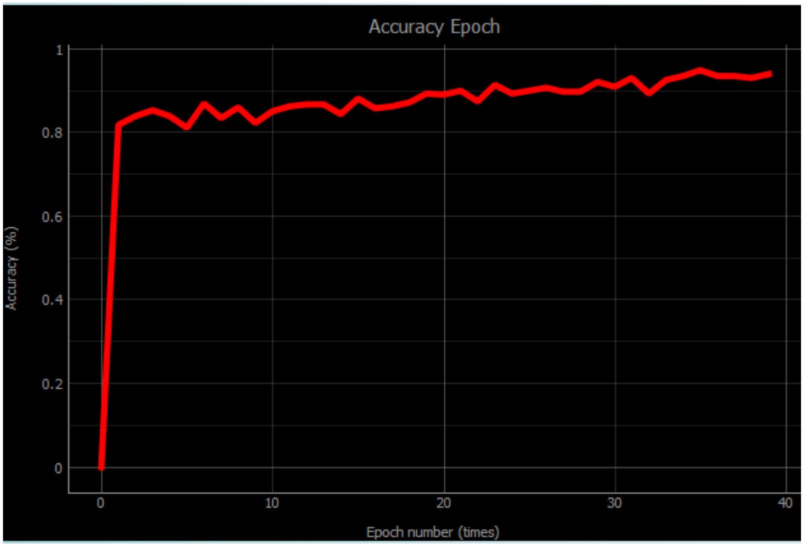

Figure 24 - Accuracy Epoch (German Database)

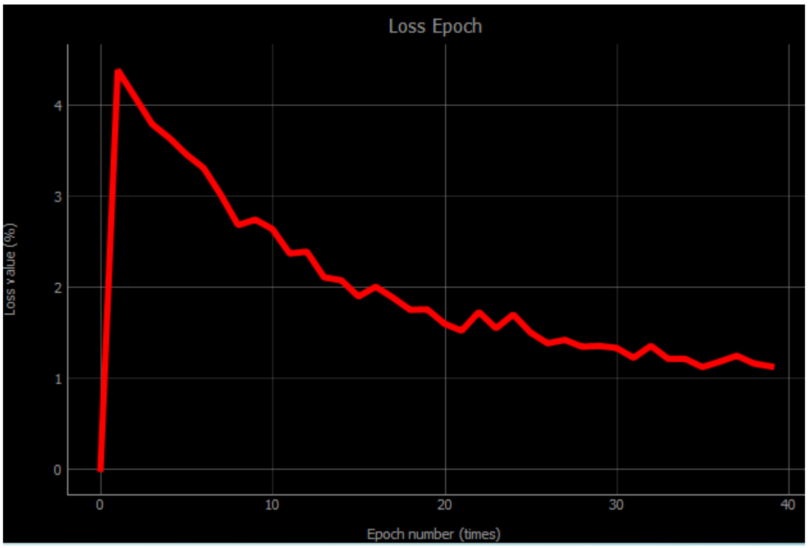

Figure 25 - Loss Epoch (German Database)

	Positive	Negative
Lie	4	10
Telling a truth	1	92

Table 5 -confusion matrix of the validation results in German

$$Accuracy = 90\%$$

$$Recall = 0.28$$

$$Precision = 0.8$$

In this experiment, we ran 535 records in German. Sixty-nine of the records classified as liars. Eighty percent of the records were taken to the learning process and the other for the validation. According to table 5, 107 records were scanned through the validation process, and the results were: 4 cases identified as telling a lie (true positive) while 10 of the liars was not identified for their lying and was a false negative. On the other hand, on a group of truth-telling, only one case identified as false positive and 92 revealed true positive answer. The accuracy of the experiment was 90% the recall, the number of correctly classified positive examples divided by the total number of actual positive samples in the test set was 0.28 and is the number of correctly classified positive cases divided by the total number of examples that are classified as positive was 0.8.

	Positive	Negative
Lie	42	150
Telling a truth	251	997

Table 6 - confusion matrix of the validation results in English on German Model

$$Accuracy = 72\%$$

$$Recall = 0.21$$

$$Precision = 0.14$$

In this experiment, we ran 1440 records in English on the model that was trained in German language. 192 of the records classified as liars. According to table 6 the results were: 42 cases identified as telling a lie (true positive) while 150 of the liars was not identified for their lying and was a false negative. On the other hand, on a group of truth-telling, 251 cases identified as false positive that means that the system recognized them as liars and 997 revealed true positive answer. The accuracy of the experiment was 72% the recall, the number of correctly classified positive examples divided by the total number of actual positive samples in the test set was 0.21 and is the number of correctly classified positive cases divided by the total number of examples that are classified as positive was 0.14.

4.1.3. Experiment 3 English database

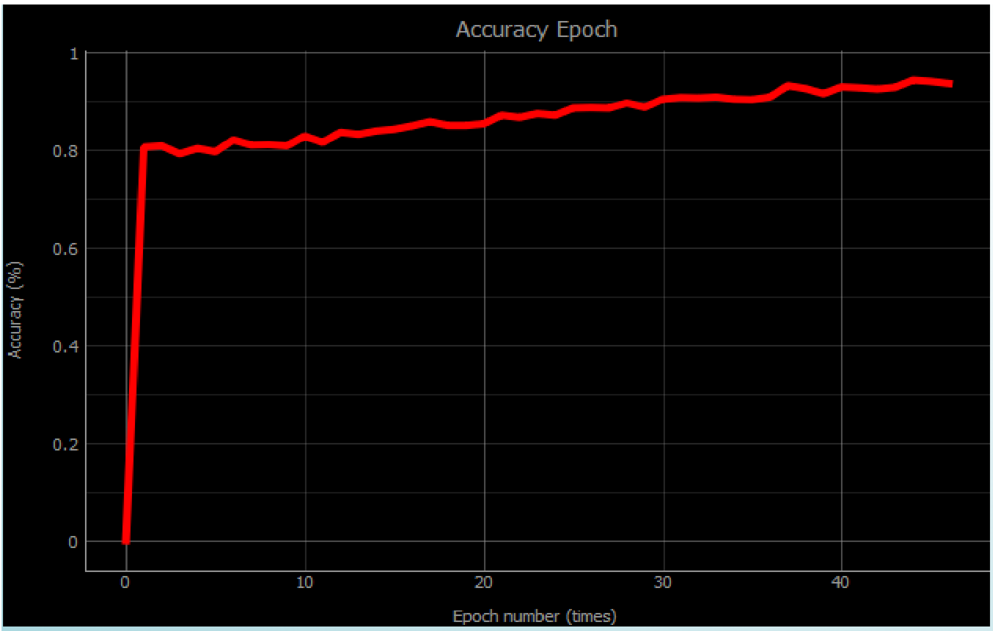

Figure 26 - Accuracy Epoch (English Database)

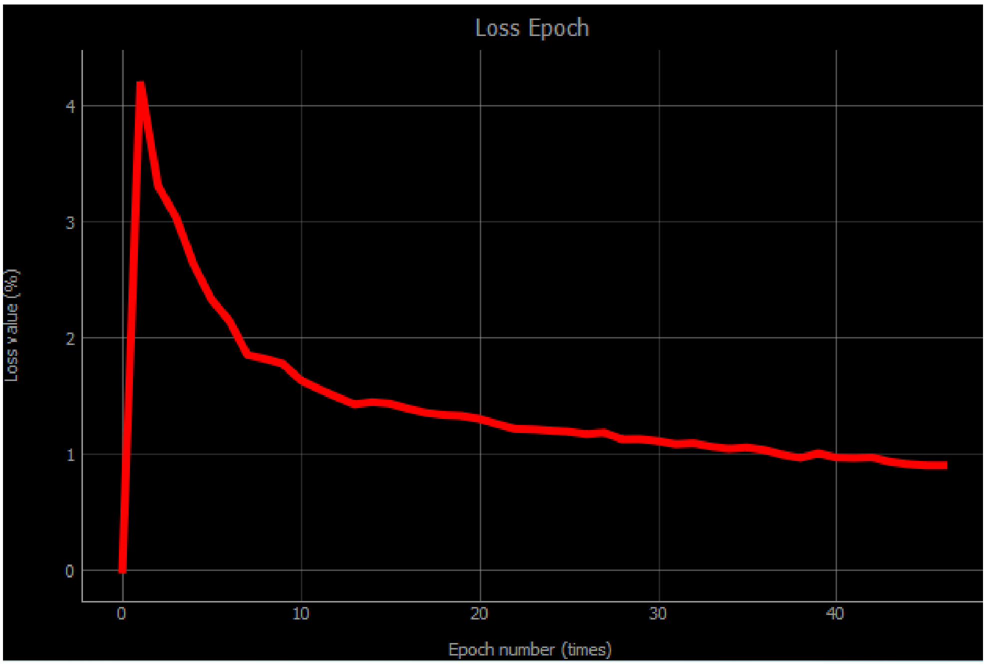

Figure 27 - Loss Epoch (English Database)

	Positive	Negative
Lie	5	38
Telling a truth	10	235

Table 7 - confusion matrix of the validation results in English on English Model

$$Accuracy = 83\%$$

$$Recall = 0.11$$

$$Precision = 0.33$$

In this experiment, we ran 1440 records in English. 43 of the records classified as liars. 1152 of the records were taken to the learning process and the other for the validation. According to table 7, 288 records were scanned through the validation process, and the results were: 5 cases identified as telling a lie (true positive) while 38 of the liars was not identified for their lying and was a false negative. On the other hand, on a group of truth-telling, only 10 case identified as false positive and 235 revealed true positive answer. The accuracy of the experiment was 83% the recall, the number of correctly classified positive examples divided by the total number of actual positive samples in the test set was 0.11 and is the number of correctly classified positive cases divided by the total number of examples that are classified as positive was 0.33.

	Positive	Negative
Lie	35	34
Telling a truth	173	293

Table 8- confusion matrix of the validation results in German on English Model

$$Accuracy = 61\%$$

$$Recall = 0.51$$

$$Precision = 0.16$$

In this experiment, we ran 535 records in German on the model that was trained in English language. 69 of the records classified as liars. According to table 8 the results were: 35 cases identified as telling a lie (true positive) while 34 of the liars was not identified for their lying and was a false negative. On the other hand, on a group of truth-telling, 173 cases identified as false positive that means that the system recognized them as liars and 293 revealed true positive answer. The accuracy of the experiment was 61% the recall, the number of correctly classified positive examples divided by the total number of actual positive samples in the test set was 0.51 and is the number of correctly classified positive cases divided by the total number of examples that are classified as positive was 0.16.

4.2. Conclusions

During this project, three lie detectors were built. The first one was based on the emotion identification using CNN paper [10], the results revealed different accuracy from test to test. The second lie detector we built was based on VGG16 model which usually used in image recognition. The results were good accuracy and loss value but an investigation showed that the model always returned false negative answers. the third model included VGG16 model with double fully connected layers and this model revealed the most successful results. We learned that the learning language has an impact on the output of the model. In order to receive higher accuracy, the input has to fit the learning language.

Also we learned that in order to receive good statistical data, it is very important to use good quality records, with no silence at the beginning.

In the future, there is an option to expand the project by using additional algorithms for processing sound and adding them to the existing input in order to increase the accuracy of the results.

5. BIBLIOGRAPHY

[1] P. Trovillo, "History of Lie Detection," *Journal of Criminal Law and Criminology,* pp. 848-881, 1939.

[2] R. Warner, "What's the Most Accurate Lie Detector? It's Not What You Think," *Huufpost,* 2017.

[3] N. Srivastava and S. Dubey, "LIE DETECTION SYSTEM USING ARTIFICIAL NEURAL NETWORK," *Journal of Global Research in Computer Science,* pp. 9-13, 2014.

[4] B. Hollis, "Physics of Sound," *The Method Behind The Music,* 2004.

[5] D. Krambeck, "An Introduction to Digital Signal Processing," *All About Circuits,* 2015.

[6] S. Gupta, J. Jafar and A. Fatimah, "FEATURE EXTRACTION USING MFCC," *Signal & Image Processing : An International Journal (SIPIJ),* vol. 4, no. 4, pp. 102-104, 2013.

[7] S. Haykin, Neural Network and Learning Machines, 3rd ed., 2008.

[8] L. Fausset, Fundamentals of Neural Network, 1994.

[9] D. Gupta, "Architecture of Convolutional Neural Networks (CNNs) demystified," *Analytics Vidhya,* 2017.

[10] E. Franti, I. Ispas, V. Dragomir, M. Dascalut, E. Zoltan and I. Stoica, "Voice Based Emotion Recognition with Convolutional Neural Networks for Companion Robots," *The CNN designing and training,* pp. 11-12, 2017.