Using Deep Learning and EMG to recognize non-audible speech

Rommel T. Fernandes
Seaver College of Science and Engineering
Loyola Marymount University
Los Angeles, CA, USA
Email: rferna16@lion.lmu.edu

***Abstract*—Many post-stroke victims deal with physiological problems such as speech impediments due to aphasia. With the advancement of Human-Computer Interface (HCI) research, this paper aims at non-audible speech recognition using Electromyography (EMG) and Deep Learning. We will first briefly introduce HCI systems such as Silent Speech Interfaces and review how Deep Learning and Machine Learning can be used for speech recognition. Work in progress...**

***Index Terms*—Deep Learning, Electromyography, Silent Speech Interfaces, Human Computer Interfaces**

I. Introduction

Over the past few years, HCI has been an increasing field of study. HCI can be described as a feedback loop between human and computer. With the increased usage of sensors worn on humans, such as watches, heart rate monitors, and other smart sensors, researchers are trying to extract bio-signal information and classify typical human activities. One of the ways HCI is used is for Silent Speech Interfaces (SSI). SSI aims to use signal-extracting systems like electromyography (EMG) and electroencephalography (EEG) to convert signals of silent or non-audible speech and use a machine to classify the results. This feedback loop involves feature extraction, model training, and activity inference [1].

This paper is motivated by recent work that uses machine learning and electrocardiogram (ECG) to detect irregular heartbeats [2] and electroencephalography (EEG) [3] and EMG [4] to predict body movements.

For the purpose of this research, non-audible speech can be classified as the inability to verbalize words or sentences through the use of sound in an effective way. SSI systems are not new; what is new is the computing resources and type of algorithms used to classify speech in SSI systems. In the past, machine learning algorithms such as decision tree, support vector machine, naïve nayes, and hidden markov models were used as classifiers for speech. Though powerful, they require extensive feature extraction from EMG signals. Typically, with traditional machine learning, only shallow features can be learned from those approaches, leading to undermined performance. Recently, we have witnessed the incremental development of deep learning, which alleviates the issue of feature engineering since the models extract valuable information through several iterations. Therefore, using EMG signals to classify non-verbal speech does not have to a be a laborious task.

The paper is organized as follows. Section II discusses the related work. Section III discusses the experimental setup used to capture, analyze and model the data. Section IV discusses the specific models being used in our proposed solution. Section V will discuss the proposed solutions. Section VI discusses the results from our analysis and compares them to the related work. Finally, we will conclude this paper with a discussion, future work, and conclusion.

II. Related Work

The research conducted using EMG to predict speech for SSI systems has been going on for over two decades. Before machine learning became very popular, using EMG to recognize speech patterns involved heavy feature extraction of the data, along with discrete mathematical modeling. Recently, there have been well documented results that have used a combination of mathematical modeling and deep learning to predict speech using EMG. Some of the research addresses syllable and single word based prediction [5], [6]. Other research has addressed using EMG to predict entire phrases [7], [8].

One of the earliest attempts to use EMG to predict speech was done in [6]. The goal was to predict isolated word recognition, which was performed on a vocabulary consisting of the ten English digits *0-9*. Seven electrodes were positioned on the face to extract bio-signals from the subjects. Hidden Markov Models (HMM) with Gaussian Mixture Models (GMM) were used as classifiers. The study was able to get an average word accuracy of 97.3%.

In [9], new approaches to machine learning models were introduced, such as Restricted Boltzmann Machine algorithms. Their corpus consisted of 25 sessions from 20 speakers comprising of 200 read English-language utterances such as phonemes, consonants, and vowels. In the results, the vowel features achieved accuracy scores of around 40%.

The work performed by [7] continued some of the primary research done by [9]. This research focused on using modern deep learning techniques such as Long Short Term Memory (LSTM) and comparing it with GMMs. Their results showed that LSTM models performed better than that of GMM, with a mean Mel-Cepstral Distortion (MCD) score of 5.46 versus

5.69, where MCD is a measure of distance, and lower numbers represent better results.

In [8], built a proof of concept SSI system that uses a one-dimensional Convolutional Neural Network (CNN) as a classifier. Seven electrodes where placed around the throat and face. In their quantitative results, an average accuracy of 92.01% for all subjects was achieved. Their corpus included individual words, and short phrases.

Finally, work done in [2] uses the latest methods to classify bio-signals by converting them to scaleograms and processing them through CNN models. The research done in this paper is used to predict irregular heartbeats through ECG signals. Similar approaches for analyzing signals with a dynamical frequency spectrum, such as EMG bio-signals, can be adopted using the same method of wavelet transformations and classification.

Our paper will attempt to research and contribute the following:

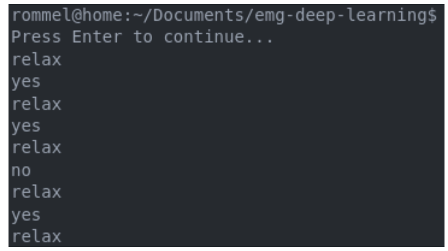
- Reproduce existing work for LSTM models used for speech recognition.
- Use similar techniques of wavelet tranforms and CNN, which is presented in [2] for ECG signals, and apply it to our EMG non-audible speech recognition models.

III. System Description

The number of electrodes and bipolar channels are far greater in [8], [9], [7], [6] than compared to our research, where we are only using three channels. This number was based on [6], which stated that EMG based speech processing requires the very least signals from the cheek area and the throat.

The system consists of two Shimmer3 EMG units, each with a 24 MHz CPU. The EMG units have the capability of recording two channels of data using Ag/AgCl bipolar electrodes with a reference electrode connected to a bone-dense area. The bipolar electrodes are placed strategically based on work done in [5]. The areas where the EMG electrodes are placed are as follows. *Depressor anguli oris* (EMG1), *Zygomaticus major* (EMG2), and *Anterior belly of the digastric* (EMG3). The reasoning of only choosing three EMG channels is to reduce discomfort by the user and abide by the minimum electrode placement documented by [6]. Each bipolar electrode of the muscle group is placed approximately 2 cm apart based on the unit specifications in (Fig.1b). After proper placement of the electrodes on a subject, the EMG units are placed on the subjects upper torso and shoulder, using comfort straps. The EMG units transmit data via Bluetooth to a Linux (Ubuntu) Intel laptop, which captures the EMG recordings and timestamps. The EMG units utilize open source Python to transmit data to the laptop.

Capturing Data: After the subject is connected to the EMG units with the electrodes in place, the process of acquiring EMG data with annotated samples begins. In our experiments we are capturing two types of annotations for a our sample data. Our *first set* of annotations consists of the labels for the words *yes* and *no*. Our *second set* of annotations consists



(a) Annotated labels displayed on screen for subject to read

(b) Connections to speech-focused muscle groups for EMG data.

Fig. 1. Subjects reading rannotated labels on screen while connected to EMG units and electrodes.

of the labels of the numeric digits *0-9*. The annotations are generated at random using a python script that prints out the label for the subject to read (Fig.1a). The label persists on the screen for two seconds; it is then followed by the word *relax*, which persists on the screen for two more seconds. The next label in the annotations is displayed and repeated at random for a total of 50 labels per annotated set. The subject performs this task for the *first set* and *second set*. The associated EMG signals captured with the annotated labels will be used to train the various deep learning models which will be discussed in section V. In total, 10 subjects are recruited to volunteer their data. For each set, the data will be split into 80/20 , which will be used for model training and validation respectively.

Cleaning Data: Once the data is captured from the subjects, post processing of the data can begin. In order to remove the noise from the signals, filters are added after acquiring the data. We assumed that we are capturing muscle movements below 4 Hz. We also want to eliminate the 60 Hz interference from the surroundings. First, we applied a low-pass filter with a cutoff frequency of 4 Hz. The filters are ideal and designed around a window sinc function [10]. After applying a low pass filter, we add a high pass filter with a cutoff frequency of 0.5 Hz, which removes the aliasing and the associated DC offset. The coinciding timestamps of the EMG data with the annotations are mapped together to create an input-output relationship. Fig.2 shows the filtered channels from the EMG with the respective annotated labels. This data will be used

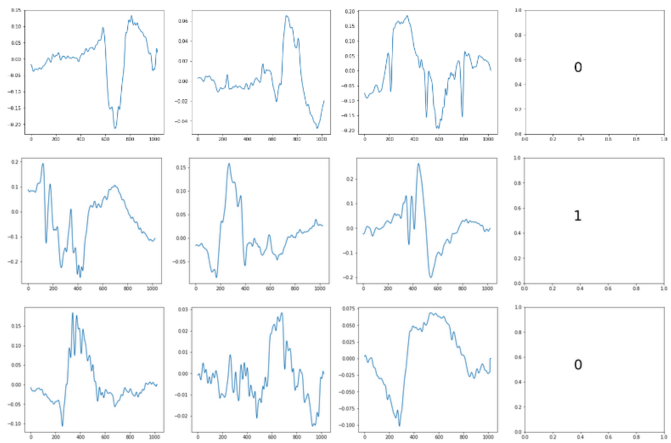


Fig. 2. Signals after cleaning and adding low-pass and high-pass filters. First Column: EMG1, Second Column: EMG2, Third Column: EMG3, Fourth Column: Annotated Labels

for the training and testing of the models.

IV. Experimental Models

Recurrent neural networks (RNN), in particular LSTMs, are an effective tool for sequence processing that learn hidden representations of their sequential input. An LSTM can use its memory cells to remember long-range information and keep track of various attributes of data it is currently processing [11]. An LSTM, is built from an RNN, which works by unrolling data into N different copies of itself. Input of data from previous time steps t_{n-1}, t_{n-2}, t_{n-3} $\ldots$, t_0 can be used when the current time-step t_n is being evaluated. RNNs can learn temporal dependencies in the sequential data, and use it to classify new data. These unique capablities make RNNs and LSTMs ideal for classification of time-series data such as EMG signals. Adding multiple layers to LSTMs can improve results to experiments reltated to speech recognition, as investigated by [12]. In our experiment, we will use the sequential data from each EMG channel to classify the annotated labels in our data sets.

Another deep learning model that we investigated, are *Convolutional Neural Networks*. CNNs are made up of neurons that have learnable weights and biases. Each neuron receives an input and performs a series of matrix operations in order to predict classification scores [13]. CNN architectures make the assumption that the inputs are images, which allow to encode values as matrices and perform dot product operations. In order to convert sequential time-series data into image representations, we investigated *Wavelet Transforms*, specifically Continuous Wavelet Transforms (CWT).

Unlike fourier transform approaches that only show a signal representation in the frequency domain, wavelet transforms show both time and frequency representations. The wavelet transform of a one-dimensinal signal will have two dimensions. This two-dimentional output is the time-scale image representaion of the signal in the form of a scaleogram. This scaleogram gives information about the dynamic behaviour of the system, similar to that of a distinguishable image. Thefore, the wavelet transform represents a suitable method for the classification of EMG signals [14] by using CNN models, which can automatically detect the class each scaleogram belongs to and classify them accordingly.

V. Proposed Solution

Once the data is filtered and transformed it will be ready for modeling. We used the two second window samples of when the subject repeated an annotated label on the screen, and dropped the instances where the word *relax* existed. In our first experiment, we experimented with an LSTM model. The LSTM model consisted of a single LSTM layer followed by a Dense output layer. The LSTM model is comprised of commonly used hyper-parameter values, such as a batch size of 32, with a learning rate of *1e-4*. To measure the loss of the model, we used binary cross-entropy for the binary cases of *yes* and *no*, and categorical cross-entropy for the cases *0-9*. We followed a many to one, sequence input architecture for the LSTM. The many to one architecture allows us to map many input sequences, in our case three, to a single output.

For our second experiment, we used a CNN architecutre as a deep learning model. We converted our singals into wavelet transforms, which generated scalograms that show resolution in the frequency and time-domains [15]. Since there are three EMG channels of data, we created three sclaeograms per label, Fig.3. We then placed the three scaleograms on top of each other and created an image representation. The coefficients from the output of the scaleogram is then utilized as features to train the CNN model. The CNN architecture is a two-layer model with a structure that can be represented as conv1-pool1-conv2-pool2-flat-dense-FC. The convolutional layers have a filter size of 3x3, and a max-pooling kernel size of 2x2, with a stride of 2. The hyper-parameters include a learning rate of *1e-4* and batch size of 8. A smaller batch size is due to the fact of the large tensor size of the training data. All of the models were trained on a Google Cloud Compute Engine with four virtual CPUs, 26 GB memory, and one NVIDIA Tesla K80 Graphics Processing Unit (GPU).

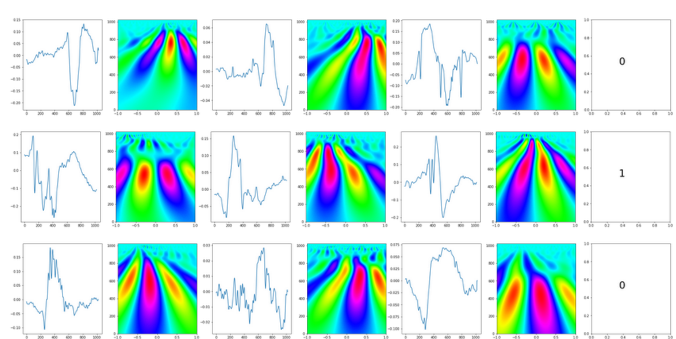


Fig. 3. Continuous wavelet transform for each EMG channel as inputs, which maps to the labeled output.

VI. Experimental Results

To evaluate the performance of the deep learning models, we used precision and recall as identifying factors. In the experiments, recall is the fraction that the model actually predicted correct. Precision is the fraction of the model making a correct positive class classification.

TABLE I
LSTM Model Results

# of class	type	Trained Data		Test Data	
		Prec.	Rec.	Prec.	Rec.
2	no	62.0	62.0	70.0	64.0
	yes	61.0	61.0	62.0	68.0
Average		61.5	61.5	66.0	66.0

TABLE II
CWT-CNN Model Results

# of class	type	Trained Data		Test Data	
		Prec.	Rec.	Prec.	Rec.
2	no	na	na	na	na
	yes	na	na	na	na
Average		na	na	na	na

Table I shows the precision and recall for the LSTM model in our binary classification case. After 10 epochs, the precision and recall scores for the test data are higher when compared to the training set. The continuous wavelet transform with CNN approach was also trained for 10 epochs. Currently for the CWT-CNN model, precision and recall scores are harder to evaluate due to the large size of the data-set.

In comparison with [8], [7], who used similar deep learning models in their research, thier results achieved better accuracy when compared to our work. A major contributing factor to their increase in performance is the abundance of data that was used in the modeling process. In [8], over 30 hours of training data was captured in order to train the model.

VII. Discussion

VIII. Conclusion

References

[1] J. Wang, Y. Chen, S. Hao, X. Peng, and L. Hu, "Deep Learning for Sensor-based Activity Recognition: A Survey," Jul. 2017. [Online]. Available: https://arxiv.org/abs/1707.03502v2

[2] "Classify Time Series Using Wavelet Analysis and Deep Learning - MATLAB & Simulink Example." [Online]. Available: https://www.mathworks.com/help/wavelet/examples/signal-classification-with-wavelet-analysis-and-convolutional-neural-networks.html

[3] A. Eltvik, "Deep Learning for the Classification of EEG Time-Frequency Representations," p. 122.

[4] A. A. Altamirano, "EMG Pattern Prediction for Upper Limb Movements Based on Wavelet and Hilbert-Huang Transform," p. 134.

[5] E. Lopez-Larraz, O. M. Mozos, J. M. Antelis, and J. Minguez, "Syllable-based speech recognition using EMG," in *2010 Annual International Conference of the IEEE Engineering in Medicine and Biology*, Aug. 2010, pp. 4699–4702.

[6] L. Maier-Hein, F. Metze, T. Schultz, and A. Waibel, "Session independent non-audible speech recognition using surface electromyography," in *IEEE Workshop on Automatic Speech Recognition and Understanding, 2005.* San Juan, Puerto Rico: IEEE, 2005, pp. 331–336. [Online]. Available: http://ieeexplore.ieee.org/document/1566521/

[7] M. Janke and L. Diener, "EMG-to-Speech: Direct Generation of Speech From Facial Electromyographic Signals," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 12, pp. 2375–2385, Dec. 2017. [Online]. Available: http://ieeexplore.ieee.org/document/8114359/

[8] A. Kapur, S. Kapur, and P. Maes, "AlterEgo: A Personalized Wearable Silent Speech Interface," in *Proceedings of the 2018 Conference on Human Information Interaction&Retrieval - IUI '18*. Tokyo, Japan: ACM Press, 2018, pp. 43–53. [Online]. Available: http://dl.acm.org/citation.cfm?doid=3172944.3172977

[9] M. Wand and T. Schultz, "Pattern learning with deep neural networks in EMG-based speech recognition," in *2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. Chicago, IL: IEEE, Aug. 2014, pp. 4200–4203. [Online]. Available: http://ieeexplore.ieee.org/document/6944550/

[10] "How to Create a Simple Low-Pass Filter - TomRoelandts.com." [Online]. Available: https://tomroelandts.com/articles/how-to-create-a-simple-low-pass-filter

[11] A. Karpathy, J. Johnson, and L. Fei-Fei, "VISUALIZING AND UNDERSTANDING RECURRENT NETWORKS," p. 11, 2016.

[12] A. Graves, A.-r. Mohamed, and G. Hinton, "Speech Recognition with Deep Recurrent Neural Networks," *arXiv:1303.5778 [cs]*, Mar. 2013, arXiv: 1303.5778. [Online]. Available: http://arxiv.org/abs/1303.5778

[13] "CS231n Convolutional Neural Networks for Visual Recognition." [Online]. Available: http://cs231n.github.io/convolutional-networks/

[14] J. Pauk, "419. Different techniques for EMG signal processing," vol. 10, no. 4, p. 7, 2008.

[15] admin, "A guide for using the Wavelet Transform in Machine Learning," Dec. 2018. [Online]. Available: http://ataspinar.com/2018/12/21/a-guide-for-using-the-wavelet-transform-in-machine-learning/