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 Interaction (HCI) research, this paper aims at non-audible speech recognition using Electromyography (EMG) and Deep Learning. We first introduce HCI systems, such as Silent Speech Interfaces and review how deep learning and machine learning can be used for speech recognition. To recognize non-audible speech, we collected facial surface EMG bio-signals from subjects for binary and multi-class labels. We then used popular deep learning techniques, which include Long Short Term Memory (LSTM)'s, and Convolutional Neural Networks (CNN)'s, along with novel signal processing approaches, such as Continuous Wavelet Transforms. We were able to report 82% precision with binary classifications of non-audible speech. In comparison with previous research, we gained insights on how to improve our results, for binary and multi-class cases, by adding more training data.**

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

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 bayes, 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-audible speech does not have to a be a laborious task.

In comparing other methods to capture non-audible speech EMG is the most effective in terms of its non-invasiveness, cost, and silent-usage. Other methods include: brain computing interfaces such as Electroencephalography (EEG), Near infrared sensors (fNIRS), implants for speech and motor cortex (ECOG), and video camera lip reading. All of these methods have some measure of personal or physical invasiveness, therefore making EMG most ideal for capturing non-audible speech.

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 Mapping 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 and Deep Neural Networks (DNN). Their corpus consisted of 25 sessions from 20 speakers comprising of 200 read English-language utterances such as phonemes, consonants, and vowels. Their results showed that DNN models performed better for phoneme related classifications using EMG inputs.

The work performed by [7] continued some of the primary research done by [9]. This research investigated bidirectional LSTMs, and compared it with other models such as 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.

Work done by [10] used the same data set and corpus as [7]. Primary work focused on evaluating CNN-based for EMG-to-Speech conversations. The researchers used LeNet inspired architecture to convert sEMG to mel-frequency cepstral coefficients (MFCCs) for modeling. Their results showed that the Lenet CNN architecture is able to outperform a plain deep neural network based conversation system.

In [8], work was done to build a proof of concept SSI system that uses a one-dimensional CNN as a classifier. Seven electrodes were placed around the throat and face. Subjects in the research do not open their mouth, make any sound, or provide any muscle articulation in order to in order to train the models. Their quantitative results for the one-dimensional CNN network reported an average accuracy of 92.01% for all subjects. 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.

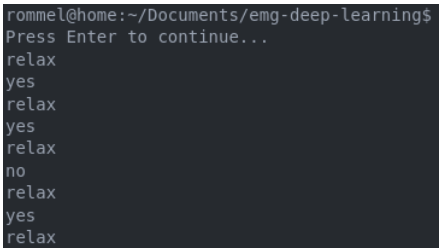
III. System Description

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 labeled annotations for our sample data. Our *first set* of annotations consists of the labels for the words *yes* and *no*. Our *second set* of annotations consists 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 [11]. After applying a low pass filter, we add a high pass filter with a cutoff frequency of 0.5 Hz, which removes 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 for the training and testing of the models.



(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.

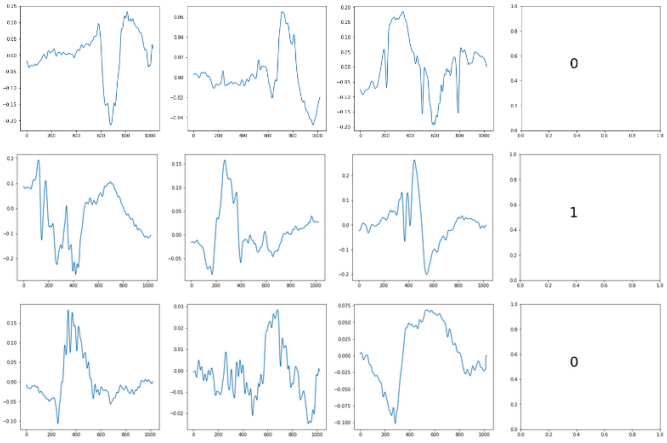


Fig. 2. Signals after cleaning and adding low-pass and high-pass filters. First Column: EMG1, Second Column: EMG2, Third Column: EMG3, Fourth Column: Annotated Labels

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 [12]. 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 related to speech recognition, as investigated by [13]. 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 [14]. Most 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. In [7], [8], [10], they generated mel-frequency cepstral coefficients (MFCC), which creates one-dimensional representation that closely characterizes the envelopes of human speech to use as features in their respective models. In [15], however states that time-frequency representations such as the CWT produced better ac curacies than the MFCC features. The wavelet transform of a one-dimensional signal will generate coefficients as a two-dimension matrix of time-frequency representations, which is also known as a scaleogram. This scaleogram gives information about the dynamic behavior of the system, similar to that of a distinguishable image. Therefore, the wavelet transform represents a suitable method for the classification of EMG signals [16] 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 with 64 filters 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

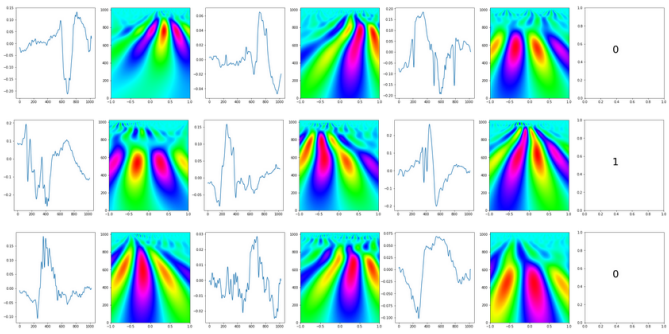


Fig. 3. Continuous wavelet transform for each EMG channel as inputs, which maps to the labeled output.

transforms, which generated scalograms that show resolution in the frequency and time-domains [17]. 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, similar to the LeNet architecture. The hidden convolutional layers have a size of 64 and 32 filters respectively. 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 16. 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).

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	95.0	99.0	93.0	74.0
	yes	99.0	95.0	72.0	93.0
Average		97.0	97.0	84.0	82.0

Table I shows the precision and recall for the LSTM model in our binary classification case. Table II shows the continuous wavelet transform with the CNN approach. The CWT-CNN model performs significantly better for both training and test data sets when compared to the LSTM model. On average for the binary test cases of *yes* and *no*, the CWT-CNN model had precision score of 82%, and the LSTM model had a precision score of 61.5%. For the multi-class test cases of *0-9*, the results were not as favorable with an average precision score below 10% for both LSTM and CWT-CNN models.

VII. Discussion

In comparison with [7], [8], [10], who used similar deep learning models, thier results achieved better accuracy when compared to the work in this paper. A contributing factor to their increase in performance is the amount of data they had available for thier training purposes. In [8], approximately 31 hours of training data was captured in order to train their one-dimensional CNN model. In [7], [10], who trained a variety of different models, utilized on-going corpus of data from previous research projects that focused on EMG-SSI systems. With additional data, we believe that our models for binary and multi-class labels can be on par with other EMG-SSI systems using deep learning that were researched in this paper.

One of the ways our model differentiates and possibly improves is the use of CWT over MFCC. The MFCC requires windowed sampling in small increments. The output of the MFCC are one-dimensional coefficients, while CWT outputs coefficients in two-dimensions, allowing deep learning models such as CNN to learn and extract more features [15]. In previous research, MFCC were primary used for shallow learning speech recognition models such as HMM. As deep learning and CNN models continue to gain popularity, the use of mult-dimensional transformation techniques such as CWT will become more prevalent for speech recognition SSI systems.

The transformation of data, specifically one-dimensional signals to its time-frequency components requires significant processing power. Resource limited systems cannot perform such laborious processes. We trained our models on a cloud computing engine which had the capability of increasing CPU and GPU power. These systems can become costly as the number of dedicated resources are assigned to the task of training a model. One has to justify the efforts of performance versus cost, specifically if the models improvements are only a few percentage. In terms of speech recognition, accuracy is important, the ability recognize speech patterns from a SSI system with high accuracy is a long-term goal.

VIII. Future Work

One of the main ways to improve accuracy in our models is to acquire more training data. Getting subjects to volunteer can at times be challenging, more importantly annotating EMG data with proper labels. Acquiring more data would allow us to also train a variety of words and phrases, so that we can eventually build a robust SSI system. In our model, we

eliminated instances where the subject was told to *rest*. Future models should incorporate the *rest* instances as an additional class, so that the model will learn to identify false positives.

The eventual goal is to build an SSI system that can improve its word recognition capablity over time as more data from similar SSI systems relay information over the network; resembling a Wireless Sensor Network and the Internet of Things [18]. A Bluetooth SSI system would communicate with a base station, process the data through the model, and output the results via an audio or visual interface. As more data is collected, deep learning models would be updated in the cloud, and pushed back to the base stations for processing.

IX. Conclusion

We have shown by using well-known deep learning methods we can effectively create models to recognize non-audible speech using surface EMG. This type of research is critical in designing SSI systems that can be used to interface with people who suffer from speech related problems. The experiments conducted by gathering data from a small sample of subjects showed that, CNN models tend to perform better than LSTM models, in terms of speed, accuracy, and precision in recognizing non-audible speech. Our approach of CWT with a combination of CNN also led to similar results when compared to previous research; however the cost of running such model may outweigh the benefits. In the future, finding ways to capture more robust data will be critical in training precise models that can disseminate and recognize a variety of words or phrases for SSI systems.

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] L. Diener, G. Felsch, M. Angrick, and T. Schultz, "Session-Independent Array-Based EMG-to-Speech Conversion using Convolu- tional Neural Networks," p. 5.

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

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

[13] 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

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

[15] M. Huzaifah, "Comparison of Time-Frequency Representations for Environmental Sound Classification using Convolutional Neural Networks," *arXiv:1706.07156 [cs]*, Jun. 2017, arXiv: 1706.07156. [Online]. Available: http://arxiv.org/abs/1706.07156

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

[17] 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/

[18] S. Ferdoush and X. Li, "Wireless Sensor Network System Design Using Raspberry Pi and Arduino for Environmental Monitoring Applications," *Procedia Computer Science*, vol. 34, pp. 103–110, Jan. 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1877050914009144