RSGC
Royal St. George's College

The Young Researcher

2021 Volume 5 | Issue 1

The Classification of EMG Signals Using Machine Learning for the Construction of a Silent Speech Interface

Varun Chandrashekhar

Recommended Citation
Chandrashekhar, V. (2021). The classification of EMG signals using machine learning for the construction of a silent speech interface. *The Young Researcher,* 5 (1), 266-283. http://www.theyoungresearcher.com/papers/chandrashekhar.pdf

ISSN: 2560-9815 (Print) 2560-9823 (Online) Journal homepage: http://www.theyoungresearcher.com

The Classification of EMG Signals Using Machine Learning for the Construction of a Silent Speech Interface

Varun Chandrashekhar

With 7.5 million people unable to speak due to various physical and mental conditions, patients are forced to use cumbersome/inefficient devices such as eye/cheek trackers. In this study, a speech aid known as a Silent-Speech-Interface (SSI) was created. This device could be used by patients with speech disorders to communicate letters in the English-alphabet voicelessly, merely by articulating words or sentences in the mouth without producing any sounds. The SSI records EMG signals from the speech system which are then classified into speech in real-time using a trained Machine Learning model. It was found that the Support Vector Machine algorithm yielded the highest SSI accuracy of 90.1% The device created measures biomedical signals and translates them into speech accurately using a Machine Learning algorithm. This study's findings could improve the accuracy of future SSIs by identifying the most accurate algorithms for use in an SSI.

Keywords: silent speech interface, EMG, machine learning, convolutional neural network, pattern recognition, speech aid

Introduction

Multiple Sclerosis (MS) is a progressive neurodegenerative disease characterized by lesions in the nervous system that affects nearly 2.3 million people worldwide. As the disease progresses, MS creates communication problems between the brain and body. Two major impairments that come with MS are speech disorders known as dysphonia and dysarthria. These speech disorders are common, affecting about 50% of MS patients (Brown, 2000). Dysphonia affects speech muscles, which can lead to patients being inaudible (Beukelman and Garrett, 1988). Other diseases such as Motor Neuron Diseases (MNDs) cause patients' speech to become unclear, taking away a patients' ability to speak. MND patients are forced to use eye/cheek tracking speech aids which make the user perform specific muscle movements to select letters/words the user wants to communicate. These trivial/cumbersome devices prove to be an extremely slow and fatiguing solution for communication. In this study, these systems which use eye/cheek tracking to develop a Speech Interface will be referred to as Conventional Speech Interfaces (CSI). Although CSI technology allows patients to communicate, it is far from optimal due to the slow rate of communication and high inaccuracy (Kapur et al, 2019). Newer speech aids use technology that doesn't involve traditional eye/cheek tracking.

I. Silent Speech Interfaces and the Electromyograph (EMG) Signal

Silent speech refers to the act of minimally or internally articulating words without producing sounds (Kapur et al, 2019). Producing silent speech is less fatiguing than regular speech or using CSIs. Although silent speech is inaudible, it produces signals that can be recorded and classified into words using Machine Learning. These signals are ElectroMyoGraph (EMG) signals which are created by subtle muscle contractions. During silent speech, speech muscles (cheek, lips, etc.) contract, producing EMG signals in certain patterns. When the same words are spoken, the same muscle contractions occur to produce specific EMG signal patterns. Thus, if the EMG signals can be recorded, it is possible to translate the signal patterns to determine the speech that was silently spoken. This allows for the development of a new type of speech interface known as Silent Speech Interfaces (SSI).

SSIs are a more effective speech aid compared to CSIs. However, an SSI's accuracy is highly dependent on the computer algorithms that are used to translate the EMG signals into speech (Kapur et al, 2019).

The most common method to record EMG signals involves placing electrodes on the skin to detect muscle contractions (Kapur et al. 2019). EMG signals recorded in this manner are also known as surface electromyograph (sEMG) signals as they are recorded from the skin's surface. EMG signals are recorded in this manner due to its non-invasive nature and easy implementation (Kapur et al, 2019).

II. Artificial Intelligence for Construction of Automated Silent Speech Interface

SSIs make use of Machine Learning to identify sEMG/EMG patterns to translate silent speech into language (Denby et al, 2010). Machine Learning (ML) is a subset of Artificial Intelligence (AI) and is defined as "the field of study that gives computers the ability to learn without being explicitly programmed" as said by Dr. Arthur Samuel (1959), who originally coined the term.

ML is a broad field and can be subdivided into 2 main categories: supervised learning and unsupervised learning. The construction of an SSI requires supervised learning, that is, the developer knows what the correct output should be, in which an algorithm is used to approximate the behavior of a function which maps data into one of several classes (Ayodele, 2010). Using supervised learning, it is possible to "teach" and develop an ML algorithm that can translate EMG signals into the letters/words that were silently spoken.

In this study, sEMG signals will be translated into one of the five vowels. Thus ML classification algorithms were used to classify sEMG signals into the letters/words that were silently spoken.

In the case of EMG signal classification, only a few supervised ML algorithms could be used for the construction of an SSI, because only a select set of ML algorithms are capable of analyzing and processing signals (series of numbers). This narrowed the ML models that could be used and set the scope of this study. Access to ML Development Software has allowed several researchers to classify EMG signals. This facilitated the past development of SSIs.

III. Previous Findings

EMGs are typically recorded through an electromyograph. These high-end machines are large/expensive and thus an inconvenient solution to monitor EMG signals for the development of an SSI. A reliable alternative to the electromyograph is the Myoware - a muscle/EMG sensor developed by Advancer Technologies - shown in Figure 1 (Char et al., 2018). A study conducted by Kareem et al. (2017) used the Myoware along with an Arduino microcontroller to record EMG signals. By comparing the sEMG signals recorded by the Myoware sensor to those recorded by the Electromyograph, they found that the Myoware can be used in ML applications due to its high accuracy. Furthermore, it was determined that sEMG signals recorded from the Myoware and the electromyograph have the same patterns (Figure 2). Therefore Kareem et al. identified that classification of sEMG signals is possible using the Myoware due to its high accuracy. This study was important as it justified the use of a Myoware sensor (used in this study) as an alternative to the electromyograph used in other studies.

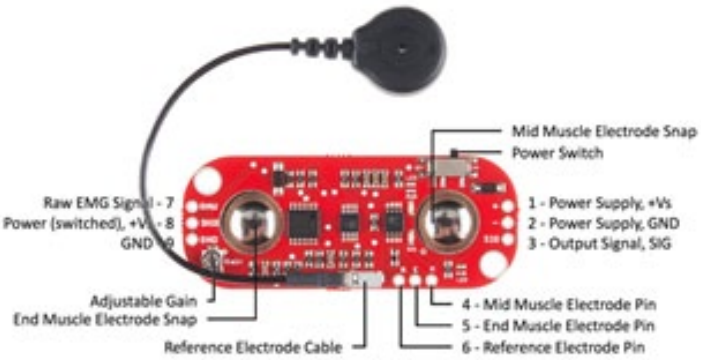


Figure 1: MyoWare Muscle Sensor
This figure shows the Myoware muscle sensor which was used to record EMG signals (Advancer Technologies, n.d.)

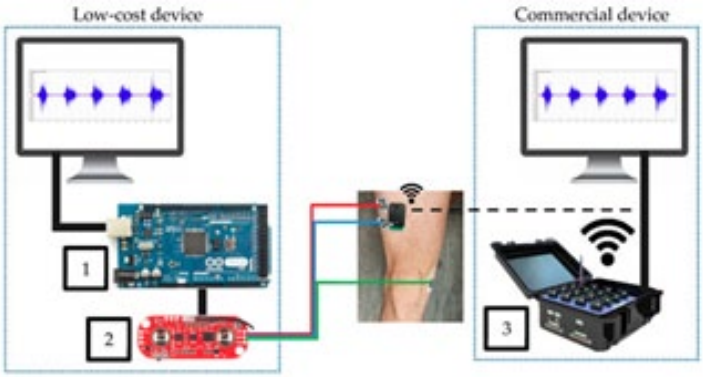
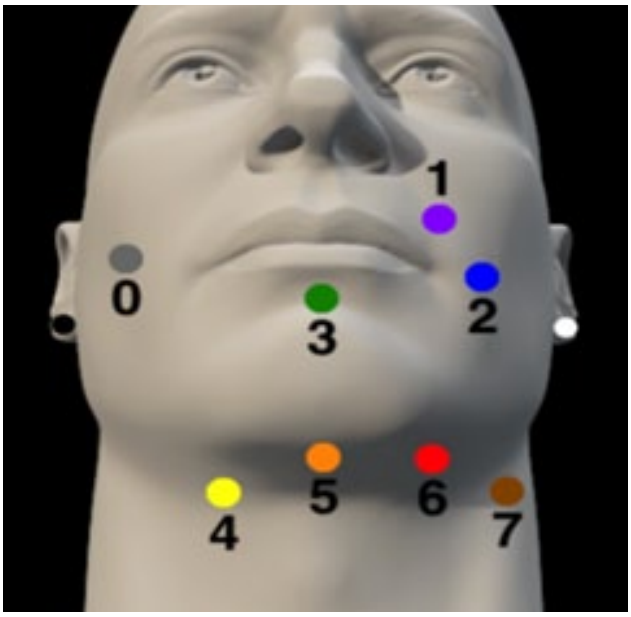


Figure 2: EMG Data Similarity
This figure shows that EMG signals collected from the Myoware and electromyograph are nearly identical (Char et al., 2018)

Non-Invasive Silent Speech Recognition in Multiple Sclerosis with Dysphonia," by Kapur et al. (2019) is one of the most advanced research on the implementation of an SSI. The constructed SSI recorded EMG signals from a multitude of locations from the face/throat as shown in Figure 3. These signals were used to train, validate, and test the Convolutional Neural Network (CNN) - An ML algorithm which identifies image features to classify images (explained in detail below) - that was used to build the SSI. Although the use of the model was never justified, the CNN model yielded high accuracy of 79%. The SSI developed improved the speed/accuracy of communication compared to CSIs (Kapur et al, 2019). This study was crucial in laying the foundation for the methodology in this study as it was the only study that identified steps to create an SSI.

Figure 3: Picture of Electrode Locations
This figure shows electrode placements used in Kapur's study (Kapur et al., 2019)



Another study by Schultz and Wand (2010) developed an SSI using the EMG-PIT corpus, a database of EMG recordings from the speech system. Using a Gaussian Mixture Model (GMM) - An ML classification algorithm that fits numerous gaussian distributions to data points - the developed SSI managed to achieve a low error rate of 10%. Similar to Kapur et al., Schultz and Wand lacks justification for the ML algorithm used. This lack of justification could suggest that other ML algorithms would perform better at classifying sEMG signals in an SSI.

IV. Introduction to the Research Gap

A paper by Karlik (2014) was fundamental to understanding the gap in the field of knowledge. Karlik's study compared several ML algorithms to classify EMG signals for use in arm prostheses. Various ML algorithms - Fuzzy Systems, Probabilistic, and Swarm intelligence - were all used to classify EMG signals. After applying ML to the EMG data, classification accuracies were compared to other studies conducted. Karlik used a CNN model to achieve a 98% classification accuracy, and found that CNN algorithms are

the most accurate type of ML algorithm for classifying EMG data for arm prosthesis.

Despite previous research involving EMG signals and SSIs, no study has identified the most effective ML algorithm to classify sEMG signals for use in an SSI. sEMG signals collected from the speech muscles have different/subtle patterns due to the weak signals produced and the variations in regular speech. Because of this distinction between EMG signals produced from the speech system and the human arm, there still is a lack of understanding of the optimal ML model to use for sEMG translation/classification in an SSI.

Furthermore, other researchers who have developed SSIs, as noted, don't compare various ML models or provide justification as to why a certain ML model was used. This further establishes the gap in research that the optimal ML model for use in an SSI hasn't been identified.

V. Types of Algorithms

Throughout the literature on EMG classification, various ML algorithms have been used (Char et al., 2018). Kapur et al (2018) developed a Convolutional Neural Network to classify EMG signals whereas in Schultz and Wand (2010), a form of Pattern Recognition was used. In the field of electroencephalography (EEG) signal classification, CNNs and pattern recognition algorithms have also been effectively used to classify these biomedical signals (Zia ur Rehman et al., 2018). Due to the common use of these ML Algorithms to classify biomedical signals, these algorithms were explored in this study.

A. Convolutional Neural Network

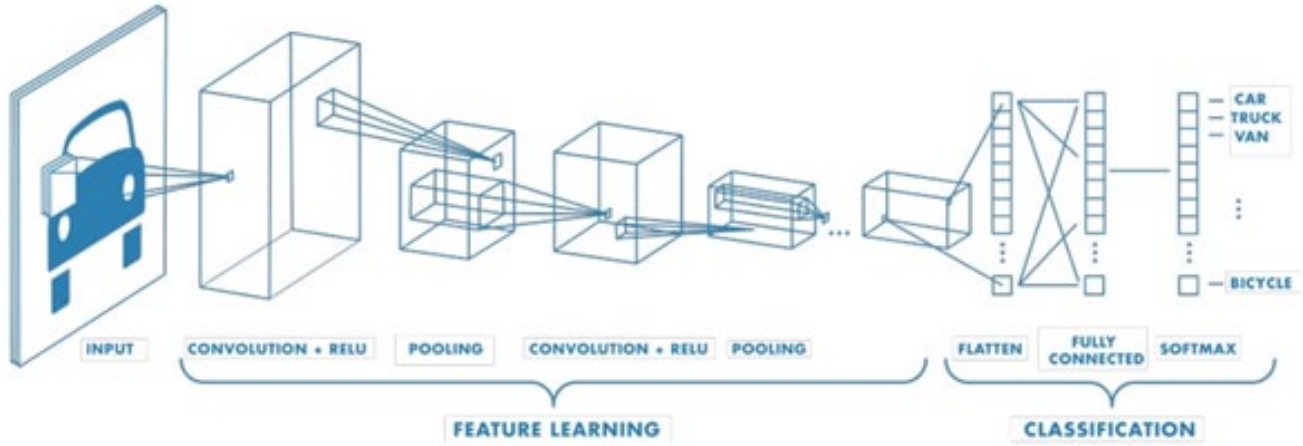
The Convolutional Neural Network (CNN) gained popularity as it was an effective method to recognize objects in images. As this method gained traction in the ML field, it has been optimized by numerous researchers allowing for the development of effective CNNs to classify images (Alaskar, 2018).

As shown in Figure 4, CNNs function by processing an input image with a series of filters known as the feature learning layers (MathWorks, n.d.) which allow the ML model to identify specific "features" of images. Once the algorithm identifies image features, the classification layers match the input images with correct output, in this case being predicted speech (letters/words).

When developing a CNN it is possible to reuse CNN models built by previous researchers and repurpose them for the applications of a new study. This process of repurposing previously developed CNNs is known as transfer learning and involves keeping the same feature learning layers of an existing CNN and replacing just the classification layers for a specific application (Bonaccorso, 2017). Computer scientists have developed numerous CNN algorithms such as

Figure 4: Convolutional Neural Network

This figure shows the different layers of CNN and highlights the Feature Learning and Classification layers (Alaskar, 2018)

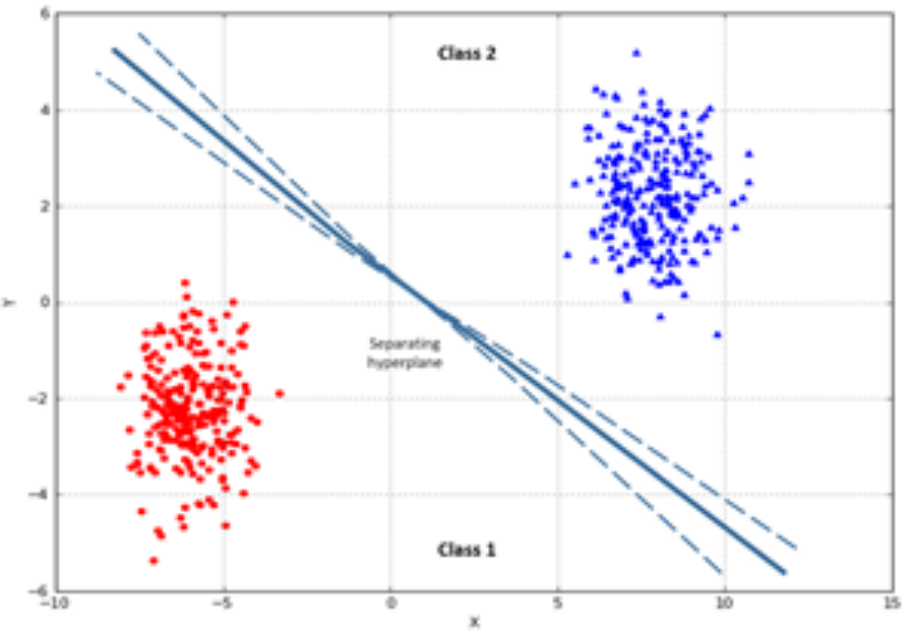


AlexNet (Krizhevsky et al., 2017) and VGGNet-16 (Muhammad et al., 2018) for various applications. However, studies show that GoogleNet (Tang et al., 2017) is the most accurate CNN for image recognition (Mohanty et al., 2016). Thus GoogleNet was chosen to be implemented in this study.

B. Pattern Recognition

Figure 5: Pattern Recognition Algorithms

This figure shows the division of classes (red/blue points) using pattern recognition (Bonaccorso, 2017)



Pattern recognition (PR) is the standardfor biomedical signal classification (Bishop and Nasrabadi, 2007) and was thus deployed in this study which aims to compare PR against the CNN (Research Goals: *Section VI. Research Goals & Research Gap*). PR is known for its simple construction and deployment. PR algorithms attempt to develop a division between multiple classes which are represented in red/blue in Figure 5. PR algorithms can determine which class is associated with an EMG signal by plotting a point based on the given inputs and identifying the point's location relative to the division. There are multiple ways to create a division using PR. In the scope of this study, 7 PR Algorithms were developed/deployed as only 7 PR models could classify EMG signals. These PR Algorithms were used to classify sEMG signals into speech.

VI. Research Goals & Research Gap

SSIs are superior to other vocal aids as shown by Kapur et al (2019) who found that SSIs enable accurate communication. Although this SSI was accurate at EMG to speech translation, no reasoning was provided to justify the use of a CNN. Similarly, Schultz and Wand (2010), also produced an accurate model, but did not justify the use of GM Models for the classification of sEMG signals.

Only Karlik (2014) generated/compared algorithms to determine the most accurate ML algorithm for EMG classification in arm prosthesis. This study identifies the CNN as the most accurate to classify EMG signals for arm prosthesis; however, EMG signals recorded from speech muscles will have different structures/patterns (Eremenko et al., n.d.). Therefore Karlik's study cannot truly identify the most accurate ML algorithm to classify EMG signals into speech. Thus there is no conclusive study to identify the best ML algorithm to classify sEMG signals into speech. This study aimed to identify the most accurate ML algorithm for use in an SSI.

This study attempts to accomplish this by comparing two different types of Machine Learning Classification Algorithms (Convolutional Neural Networks and Pattern Recognition) to identify the most accurate algorithm for use in an SSI. Therefore, this project aims to construct two types of ML algorithms in order to gain insight into the question:

"Which type of Machine Learning Algorithm (Convolutional Neural Networks or Pattern Recognition) is most accurate at classifying surface ElectroMyoGraph (sEMG) signals from the submental triangle (area under the chin) to develop a Silent Speech Interface?" This study would help further improve the ability of SSIs to translate sEMG signals into speech, allowing for more accurate communication.

Engineering Goal: To Develop a Speech Interface using a Muscle Sensor that can both collect and classify sEMG signals from the submental triangle with greater than 80% accuracy.

The engineering goal of achieving an 80% accuracy

was developed as other studies on SSIs also strived to acquire an 80% accuracy. The engineering goal also involved creating an SSI using a low-cost muscle sensor (Myoware) as previous studies only used electromyographs (Karlik, 2014).

Methodology

To achieve the research/engineering goal an SSI that translates sEMG signals into speech using ML needed to be developed. Additionally, to develop the ML algorithms, an EMG dataset to train/test the ML algorithms had to be created. This dataset was created with a total of 1020 EMG recordings - 170 for each of the 5 vowels and another 170 to establish a baseline of not speaking at all. This dataset was developed through the use of a developed Arduino-based EMG recorder. Once the EMG recorder is connected to a laptop running the ML models for EMG classification, the device will function as an SSI which can both record and translate sEMG signals generated from silent speech. To translate EMG recordings to speech, multiple ML algorithms were constructed.

To compare these various methods of ML, a true quantitative experimental method was developed to evaluate the performance of different types of ML algorithms. Additionally, an engineering method was developed to evaluate the SSI created. This true quantitative experimental and engineering method involved using the classification accuracies and F1 scores. Classification accuracy is a measure of how often the model is correct whereas F1 scores provide a more holistic view of the model taking both accuracy and precision into account (Eremenko et al., n.d.). An F1 score is the harmonic mean between the precision and recall of an ML algorithm and is commonly used to compare classification algorithms. These two parameters (classification accuracy and F1 score) were used to answer the research question and determine if the engineering goal was met.

To identify the most accurate model through the true quantitative experimental method, the tested models' F1 scores were compared just as in Karlik (2014). F1 scores range from 0 to 1, and high scores indicate that a model is both accurate and precise (Eremenko et al., n.d.). The model with the highest score was identified as the best performing ML algorithm, answering the research question.

To determine if the created SSI met the engineering goal (80% accuracy), only the accuracy of the ML model with the highest F1 score was considered. This is because the SSI will use the best performing model (model with highest F1 score - identified in experimental method) to translate EMG signals. Thus the accuracy of the SSI is equal to the accuracy of the algorithm used in the SSI.

This method of comparing F1 scores is a common way of evaluating ML algorithms. This study's results are valid as the use of standard algorithm evaluation parameters creates a standard method of comparison between tested algorithms. By using this method of analysis, it is possible to accurately identify which ML algorithm is best suited to translate EMG signals and if the SSI met the engineering goal (80% accuracy).

To evaluate models, the SSI had to be created. Creating an SSI involved the following procedures based on Kapur et al(2019):

1. Arduino-Based EMG Recorder Development - Creating a device to record EMG/sEMG signals
2. Database Preparation - Creating a dataset for Machine Learning using the created EMG recorder
3. PR Methods - Developing PR Algorithms to classify signals
4. CNN Methods - Developing a CNN to classify signals
5. EMG-based Silent Speech Interface Methods - Create an SSI

The above procedures are discussed in further detail below.

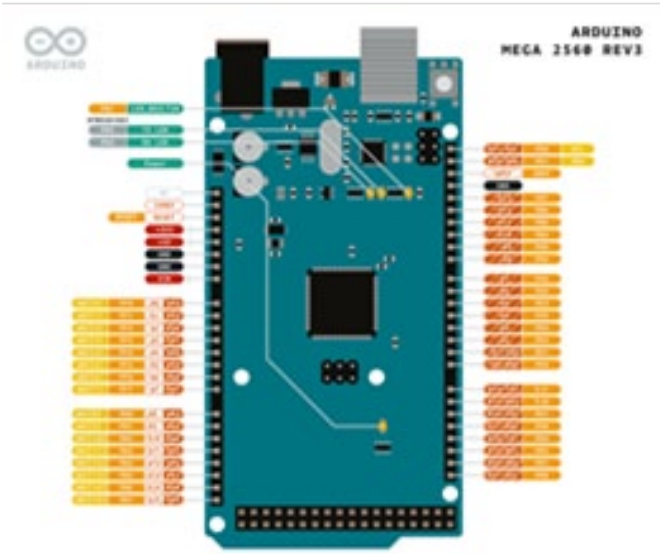
I. Arduino-Based EMG Recorder Development

The first step in the development of the EMG recorder was wiring components together. The EMG recorder would make use of a microcontroller (Arduino) that takes EMG recordings from a sensor and saves data to an SD card. The recorder has buttons to start recording EMG signals.

The Arduino - a small computer that can receive inputs from many sensors (Arduino, n.d.) - served to record EMG signals using the Myoware sensor. The Arduino Mega (Figure 6), was used due to its high

sampling rate (Hartman, n.d.) which is crucial for ML applications as detailed EMG data can be collected (Eremenko et al., n.d.).

Figure 6: Arduino Mega Diagram
This figure shows a diagram of the Arduino Mega that was used to create the EMG recorder (Arduino, n.d.)



For the Arduino to record EMG signals, the Myoware muscle sensor (Figure 1) was used to detect EMG signals. As discussed previously, the MyoWare is the best commercially available muscle sensor and has been used in other studies due to its reliability/accuracy (Hartman, n.d.). Although Kapur et al (2019) doesn't use the Myoware, Kareem et al (2017) justifies the use of this sensor as it produces accurate results.

The Arduino Mega, Myoware, and button were all connected as shown in the schematic (Figure 7) and diagram (Figure 8) below. To ensure functionality, the device was connected this way according to data sheets provided by the Arduino company (n.d.).

Figure 7: Schematic of EMG Recorder
This figure shows the EMG recorder schematic

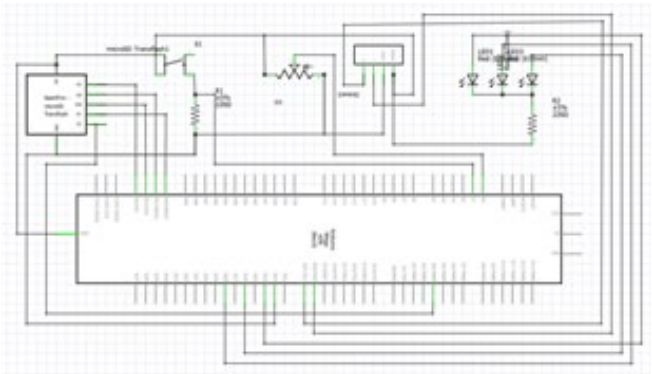
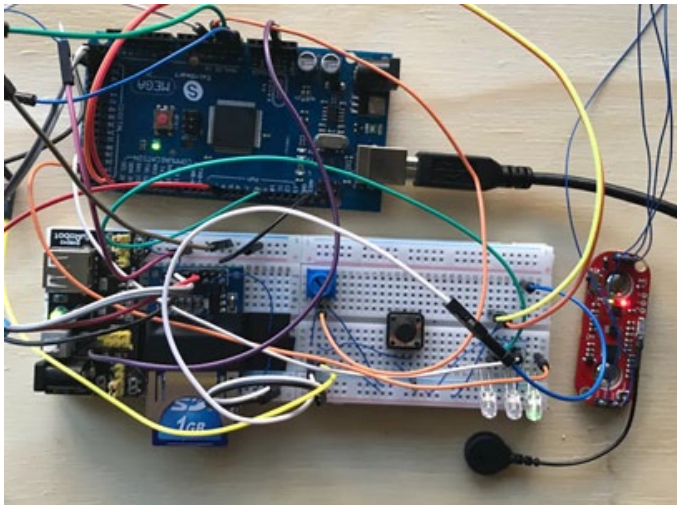


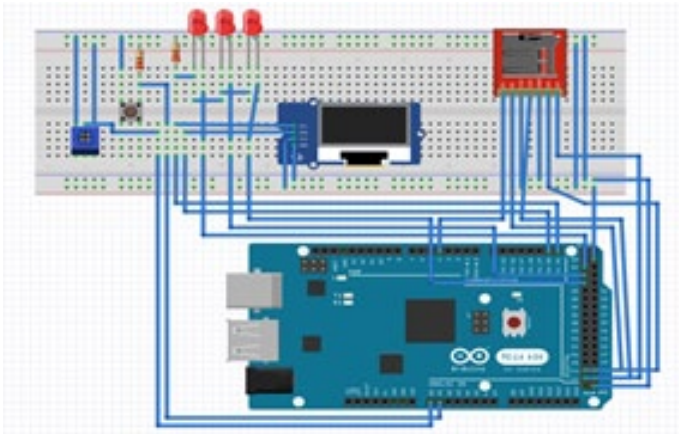
Figure 9: Image of Developed EMG Recorder
This figure shows an image of the created EMG recorder with the red Myoware sensor to the right and the blue Arduino Mega at the top.



An image of the fully constructed EMG recorder is shown in Figure 9. This EMG recorder was programmed using Java to perform various tasks. The device has 3 tasks (discussed in further detail below):

1. Wait for the button to be pressed
2. Take user input on what letter is being silently spoken
3. Record and Save EMG values as quickly as possible

Figure 8: Diagram of EMG recorder
This figure shows the EMG recorder wiring diagram



When the user was silently speaking, the button was pressed causing the device to start recording EMG signals from the silent speech. Tasks 1 and 2 (wait for the button to be pressed & take user input on what letter is being silently spoken) involved taking in user input. The code in Table I shows the commands executed to determine if the button has been pressed, whereas the code in Table II shows the commands executed to take in the input of what letter (A, E, I, O, U) is silently spoken. The code for the button allows the device to start recording EMG data only when the user is ready to speak. This ensures that EMG data is only recorded when silent speech is produced (Kapur et al, 2019). The code to determine what letter is being silently spoken is important as the device needs to associate each EMG recording with a specific letter.

To perform task 3 (record EMG data), the code shown in Table III is executed. This code both records the EMG signals and creates a dataset at the same time. The program does this simultaneously, as it optimizes the program's speed allowing many data points to be collected in a few seconds. The code in Table III is optimized to save 3000 comma-delimited EMG values in approximately 1.5 seconds. The sampling rate of the EMG recorder was 2000 Hz and the feature space was 3000 data points long. The duration of EMG recording after the onset of recording was 1.5 seconds. The features that were inputted for training the ML models were the set 3000 data points collected through the EMG recorder.

After EMG data has been recorded, the data has to be saved. This is done after recording data by using the "Data.close();" function which saves previously recorded EMG data on an SD card (Table IV).

TABLE I
CHECKING STATE OF BUTTON

```
void Button() {
  if (buttonState != !(digitalRead(buttonPin))) { // Button
    has been pressed
   buttonState = !buttonState; // Inverts signal
   displayEMG(); //Update Display
  }
}
```

TABLE II
RECORDING THE LETTER SPOKEN

```
void Potentiometer() {
  potVal = analogRead(potPin); // Reads the "current" state
    of the Potentiometer
  if (abs(oldPotVal - potVal) >= 10) { // If Potentiometer
    Value has changed
   oldPotVal = potVal; // Updating Old Potentiometer Value
   index = map(potVal, 0, 1024, 0, 5); // Mapping Values to
    index value
   Letter = letterList[index]; // Updating Letter Variable

   displayEMG();
  }

}
```

TABLE III
RECORDING EMG DATA

```
void recordEMG() { // Recording EMG Optimized for speed
  for (int i = 0; i <= 3000; i++) {
    Data.print(String(analogRead(modEMGPin)) + ","); // print Raw EMG values
  }
}
```

II. Database Preparation

A database needed to be created to train the ML models. This data was created using the created EMG

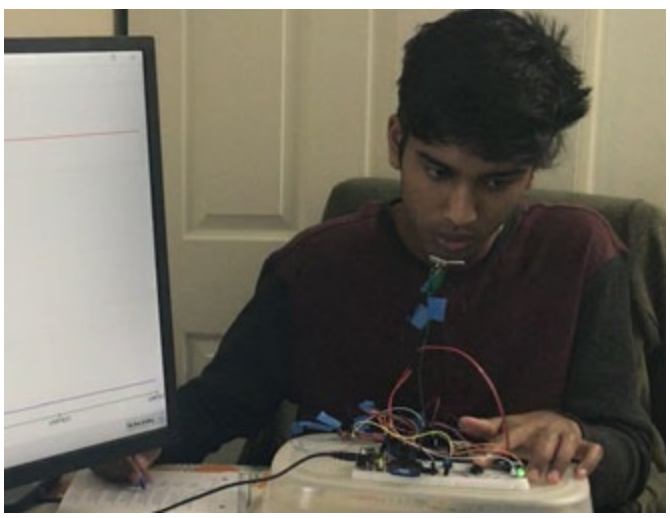
TABLE IV
SAVING DATA ON SD CARD

```
void loop() {
 if (buttonState == 1) { // If button is pressed
   Data = SD.open(Letter, FILE_WRITE); //Open SD card
    for Writing

   recordEMG(); // Void Loop for Recording EMG Signal

   Data.print(";");
   Data.close(); // Closing Data file
 }
```

Figure 10: Electrode Placements and Generating Data



This figure shows electrode placements used when collecting data. Electrodes are placed under the chin (submental triangle), which was also done in Kapur's study

recorder device (Figure 9). Three electrodes were attached to the submental triangle, the area under the chin (Figure 10). A total of 1020 EMG recordings were taken, 170 for each of the 5 vowels and another 170 to establish a baseline of not speaking at all. The letters silently spoken for each EMG recording trial were in a random order - generated by Excel sheets. Electrode placements were justified by Kapur's research where he identifies various areas on the throat to collect EMG signals (Figure 3). The muscle/area targeted in this study is marked with an orange dot that is labeled "5" in Figure 3. Due to time constraints only 1020 signals were collected in this study.

III. Pattern Recognition Methods

Once imported into the MATLAB programming environment, the Classification Learner App - an ML tool used for developing ML algorithms (MathWorks, *Classification Learner App* n.d.) - was utilized to develop the Pattern Recognition Algorithms to classify sEMG signals (Eremenko et al., n.d.). The MATLAB Classification App allowed the easy implementation of different PR Algorithms. Only 7 types of Classification Algorithms - which are commonly used for Pattern Recognition - were capable of translating EMG signals and they were all implemented: Ensemble, K-Nearest Neighbors (K-NN), Decision Tree Classification, Naive Bayes (NB), Linear Discriminant, Quadratic Discriminant and the Support Vector Machine (SVM) Classifier.

The assumptions and ways these divisions are built are different for each of the 7 implemented Pattern

Figure 11: Divisions for Pattern Recognition Algorithms

This figure provides a brief description of the way each division is formed for the 7 implemented pattern recognition algorithms

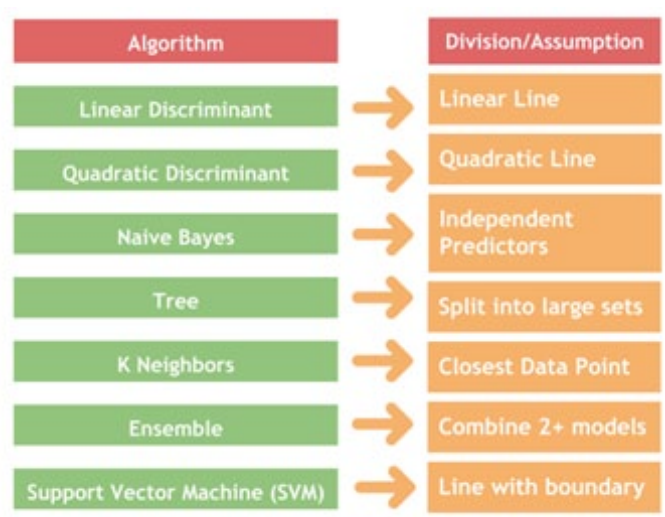


TABLE V
IMPORTING SVM CLASSIFIER

```
classificationSVM = fitcecoc(...
   predictors, ...
   response, ...
   'ClassNames', {'A'; 'E'; 'I'; 'O'; 'U'; 'classification'});
```

TABLE VI
IMPORTING KNN CLASSIFIER

```
classificationKNN = fitcknn(...
   predictors, ...
   response, ...
   'ClassNames', {'A'; 'E'; 'I'; 'O'; 'U'; 'classification'});
```

recognition algorithms. The type of division/assumption made for each algorithm is shown on Figure 11.

The EMG data and the corresponding vowel/letter were imported into the computer. Each algorithm was trained/tested on the same dataset to ensure the validity of the results. Due to time constraints and limited computing power, each algorithm was given only 10 iterations (opportunities) to learn from the data, ensuring that no ML model had an advantage over the other tested algorithms. Additionally train-test splits were used opposed to other traditional methods for training involving K-fold cross validation. The train-test split was used as it was less computationally intensive (80% - train; 20% - test). After training each PR model, each algorithm was tested on the previously developed testing data to determine classification accuracy and F1 scores.

TABLE VII
GOOGLENET IMPORT

```
net = googlenet;
```

IV. Convolutional Neural Network Methods

A method known as transfer learning was applied to create a CNN suited to analyze images of EMG signals. The algorithm was developed and run in MATLAB, using GoogleNet as a basis to create and structure the algorithm (MathWorks, *Googlenet* n.d.). GoogleNet - an open-source CNN - has been used in many research studies due to its high image recognition accuracy which surpasses other prebuilt CNN's used in other studies. The following function in Table VII was used to import the prebuilt CNN into the MATLAB workspace.

The GoogleNet algorithm was repurposed for this study to classify EMG signals into letters. This process of repurposing classification layers (Figure 12) from an existing model is known as transfer learning and is a common process used by many researchers. Transfer learning is beneficial as the CNN model developed using GoogleNet will likely have higher accuracy than other CNN algorithms (MathWorks, *Classification Learner App* n.d.).

Because CNN's require image inputs for classification/training, the signal which was originally a series of numbers had to be converted into an image. In this study, EMG data - in the form of numbers - was converted into a spectrogram, a visual representa-

Figure 12: Transfer Learning Implementation
This figure shows the steps required to use transfer learning for an application (MathWorks, n.d.)

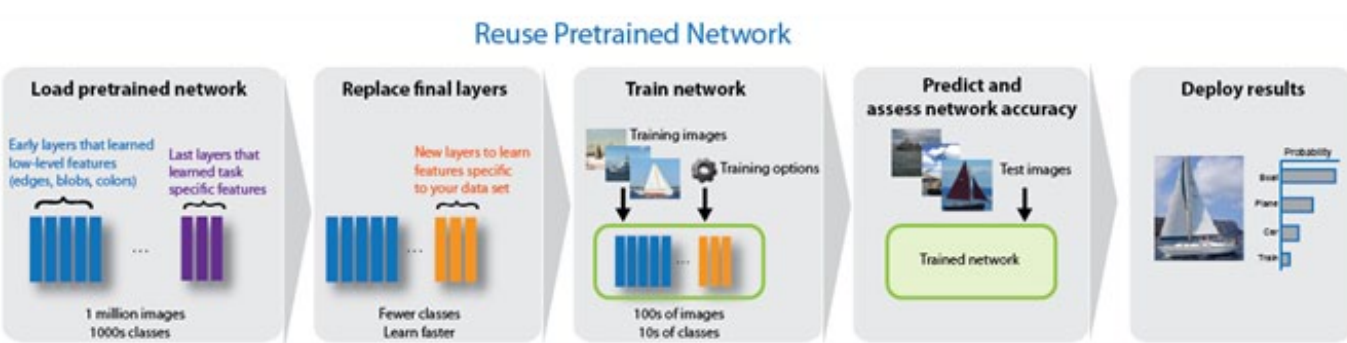


Figure 13: Spectrogram for the Vowel "A"
This figure shows the spectrogram for the vowel "A"
X-axis = time, Y-axis = frequency, green = higher amplitude

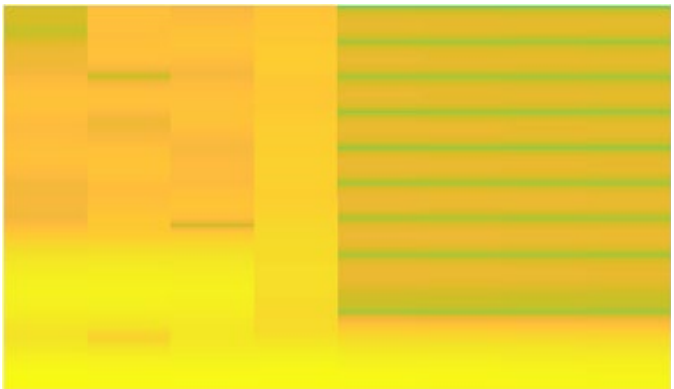


TABLE VIII
CREATING A SPECTROGRAM

```
spectrogram(eval(signalName), [], [], [], 'yaxis');
```

tion of a signal. Spectrograms represent amplitude of a signal over time at various frequencies present in a particular waveform. Spectrograms were used in Kapur's study when developing an SSI. A spectrogram (Figure 13) for each of the 1020 EMG signals in the dataset was built using the function shown in Table VIII. This function came from the signal processing toolbox available in MATLAB - a programming environment - and used a short-time Fourier transform to create the Spectrogram. These spectrograms were used to train/test the CNN.

V. Creation of Silent Speech Interface

An SSI is a speech aid that records silent speech and uses an ML algorithm to translate the recorded EMG signals (Figure 14). Therefore the developed SSI has to be able to record EMG signals and then translate those signals using ML. This was accomplished by combining the previously built EMG recorder and a computer running ML algorithms.

Additionally, because the SSI's use the ML model to translate EMG signals into speech, the translation accuracy of the developed SSI is equal to the accuracy of the best performing ML model. Therefore, as explained before, the engineering goal for this project can be validated using the calculated accuracy of the best-performing ML model.

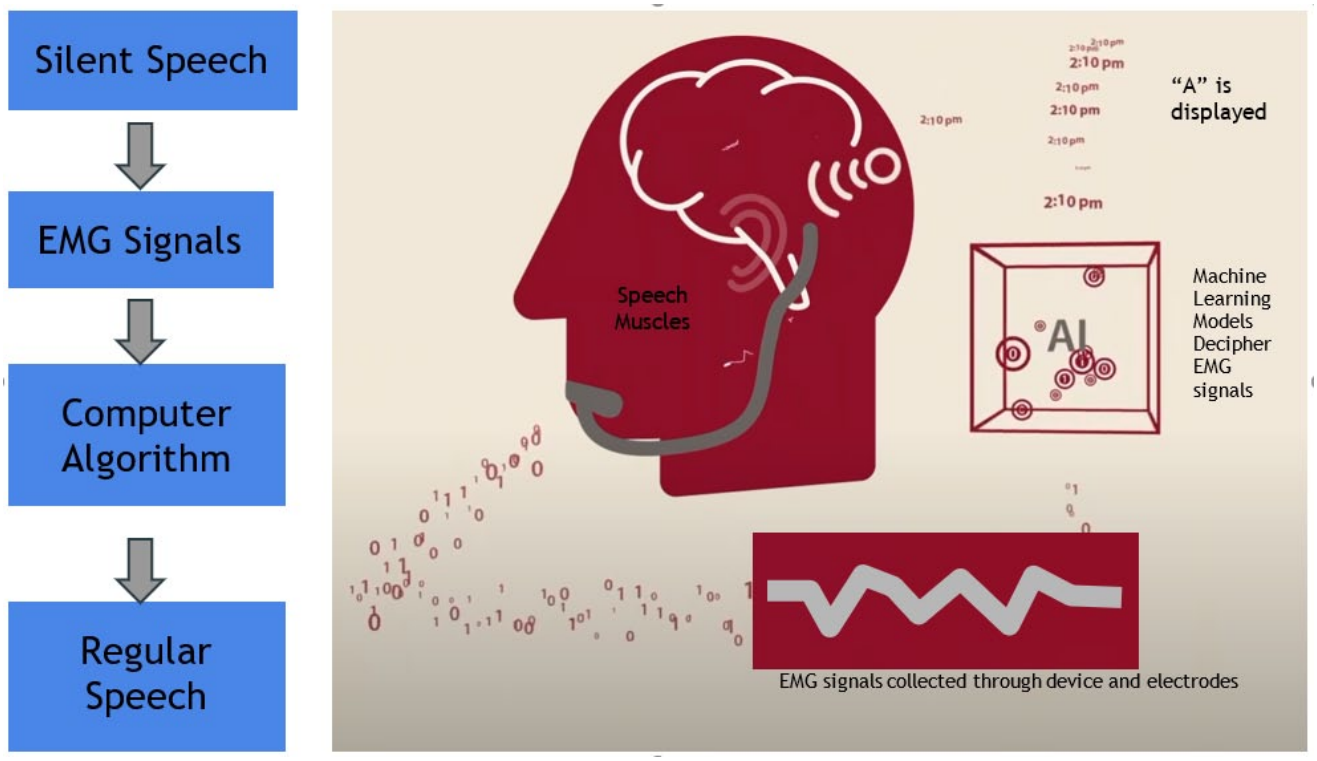
Data Analysis & Results

As discussed in the methodology, the ML algorithms were tested using the same testing set. The classification accuracy and F1 scores were calculated for each tested model. F1 scores are commonly used by data scientists to compare ML models and determine which model is holistically better (Wood, 2019). F1 scores for a model are always calculated by taking the harmonic mean of precision and recall of a model. The precision and recall of a model is always calculated the same way regardless of the number of classes. The code for calculating F1 scores is shown in Table

TABLE IX
CALCULATING F-SCORES

```
% Note: Variables were declared individually for each
  model
% The average of all calculated precision and recall values
  were used. This allow
tp = 50; fp = 50; fn = 5;
precision = tp / (tp + fp); recall = tp / (tp + fn);
F1 = (2 * precision * recall) / (precision + recall);
```

Figure 14: Silent Speech Interface Structure
This figure shows the general structure of an SSI that was used in this study as well as Kapur's study (Kapur et al.,



IX. The classification accuracies and F1 scores for the PR and CNN models are shown in Table X. The classification accuracies for these models were calculated to gain further understanding on what model could provide the most effective/accurate results although it may mean the model did not train completely from

TABLE X
CLASSIFICATION ACCURACY AND F-SCORES

MODEL	CLASSIFICATION ACCURACY	F1 SCORES
CONVOLUTIONAL NEURAL NETWORK (CNN) - GOOGLENET	54.90%	0.60
SUPPORT VECTOR MACHINE (SVM) - GAUSSIAN	80.10%	0.81
ENSEMBLE - BAGGED TREES	74.60%	0.73
K-NEAREST NEIGHBORS (KNN) - WEIGHTED	66.70%	0.70
TREE - MEDIUM	59.80%	0.65
NAIVE BAYES - KERNEL	59.30%	0.65
QUADRATIC DISCRIMINANT	55.50%	0.54
LINEAR DISCRIMINANT	49.50%	0.48

the data. No metrics on error or precision of results were given by the MATLAB development software so they were not explored in the scope of this study.

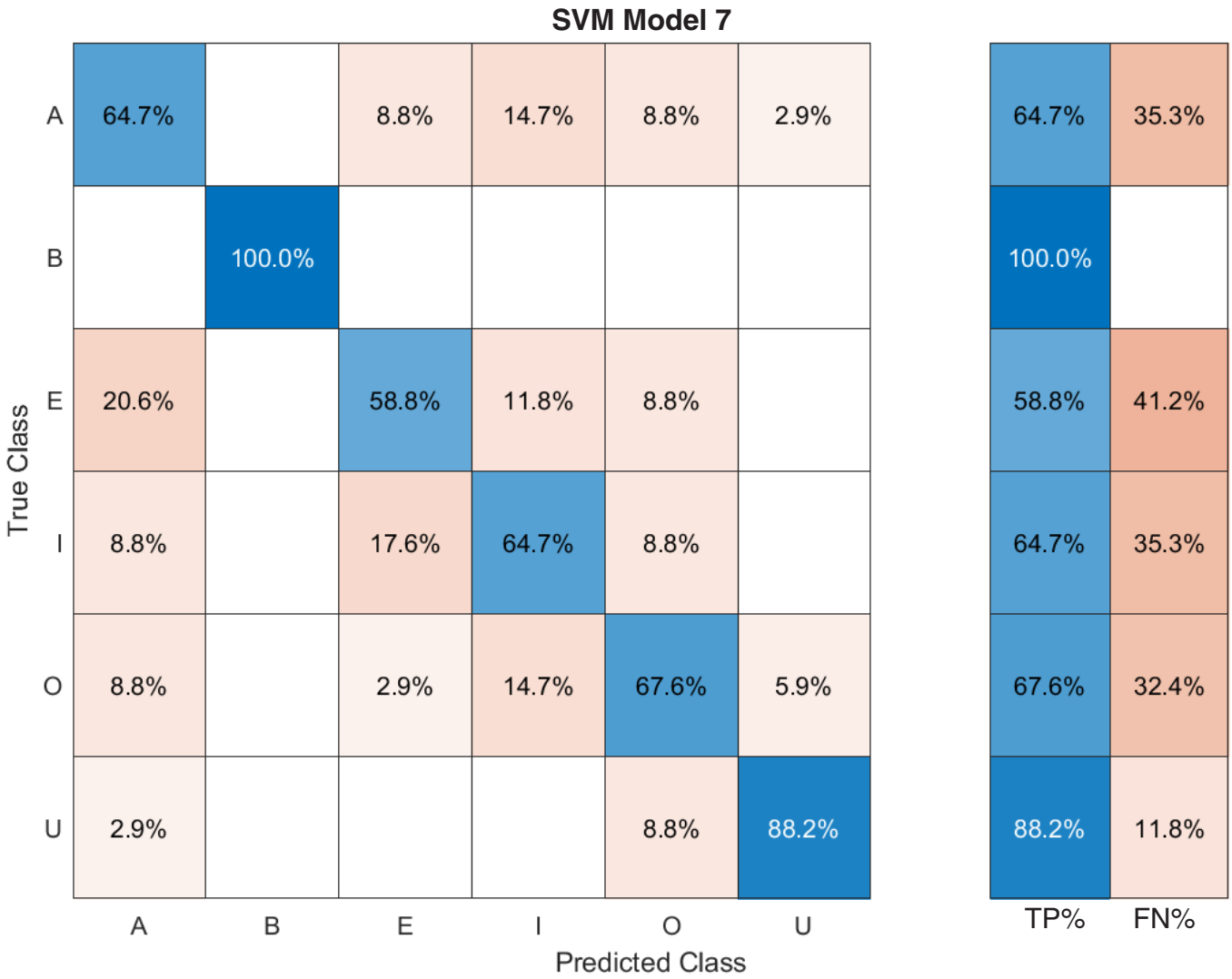
As seen in Table X, the SVM model achieved the highest classification accuracy and F1 score. The accuracy values of the ML models tested ranged from 49.5% to 80.1% while the F1 scores ranged from 0.48 to 0.81. It can also be seen that the F1 scores closely correlated with the classification accuracy for each model and were often only ± 0.02 away from the classification accuracy.

Confusion Matrices are a common way to depict the accuracies of ML models. The Confusion Matrices are shown for the SVM model (Figure 15), which had the highest F1-score/accuracy, and the CNN model (Figure 16) which was the only non-PR algorithm tested. The confusion matrices for both of these algorithms were compared as these algorithms were the most accurate Models for the 2 types of ML models explored in this study (CNN and PR).

In Confusion Matrices (Figures 15 & 16), the rows denote what letter was silently spoken, whereas the column shows which letter was predicted by the trained ML algorithm. Therefore, all the correct predictions lie along the diagonal vector shaded in blue whereas incorrect predictions are shaded in orange.

Through analysis of the Confusion Matrix, it can be seen that both the SVM model (Figure 15) and CNN model (Figure 16) classified the signals for "not speaking" denoted by "B" (Blank) on the axes, with a 100% accuracy. The SVM model's largest error was due to misclassifying the EMG signals for letter "E" as the

Figure 15: Confusion Matrix of Support Vector MachineConfusion Matrix of the SVM model which has an 80.1% accuracy and F1 score of 0.81



letter "A". This accounted for 20.6% of incorrect predictions associated with the letter E. The CNN model's largest error was due to the misclassifying the EMG signals for letter "A" as the letter "I". This accounted for 35.3% of incorrect predictions associated with the letter "A".

Both the SVM and CNN models performed poorly when classifying the letters "A" and "I". This misclassification trend also occurred in the other tested PR models indicating that the EMG signals for "A" and "I" are hard to decipher. The SVM model had accuracies of 64.7% and 56.8% for the letters "A" and "I" respectively whereas the CNN model had accuracies of 20.6% and 32.4% for the same letters. These accuracy values for individual letters were the lowest and brought down the overall F1-score/accuracy.

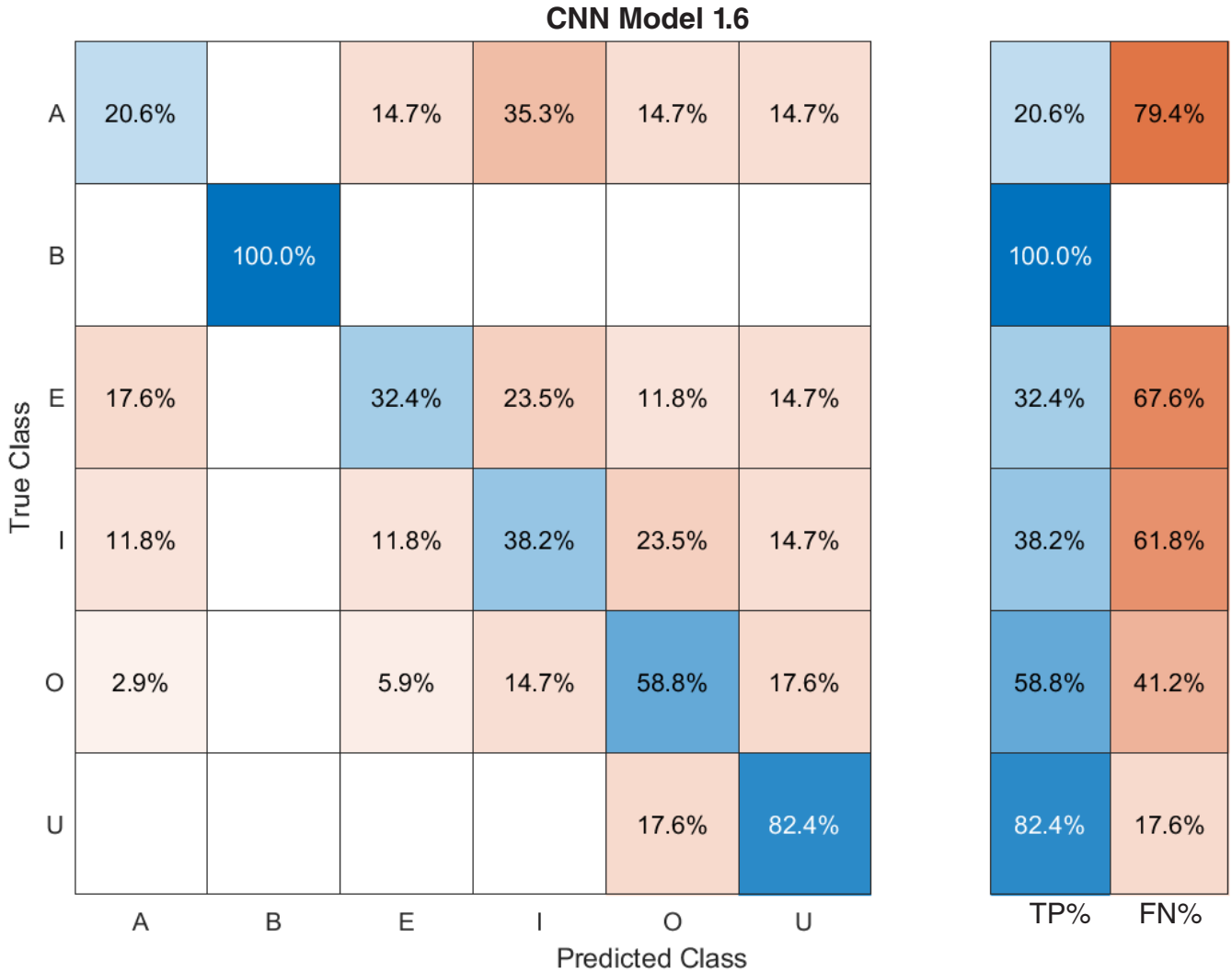
Because these inaccurate predictions were produced by the ML models, the prediction accuracy can improve if the ML models are given more data to train ("learn") from.

Discussion & Conclusion

I. Speech Impediments and Speech Aids

Speech Disorders are common among those with Motor Neuron Diseases (MNDs). MND patients are forced to use CSIs which are cumbersome and inaccurate speech aid systems. These systems make the user perform fatiguing muscle movements to select letters

Figure 16: Confusion Matrix of Convolutional Neural Network (CNN)Confusion Matrix of the CNN which has a 54.9% accuracy and F1 score of 0.60



the user wants to communicate. These inefficient and cumbersome devices prove to be an extremely slow and fatiguing solution for communication.

These issues prompted the development of an SSI which could be used by patients with speech disorders to communicate letters in the English alphabet voicelessly, merely by articulating words or sentences in the mouth without producing any sounds.

SSIs record EMG signals from speech muscles and translate these signals into speech with the use of ML. Existing research on SSIs makes use of high-end equipment (electromyograph) and also fails to justify the ML algorithms used to perform the EMG to speech translation. Furthermore, no study has identified the most accurate ML algorithm to use in an SSI. These gaps in research prompted this study which involved creating an accurate SSI and identifying the most accurate ML algorithm for use in an SSI.

II. Research Goals

Overall, this research study attempts to build an understanding to eliminate the existing gap in the field of knowledge by comparing two different types of ML Algorithms - CNNs and Pattern Recognition - to identify the most accurate ML algorithm for use in an SSI. Therefore, this project aims to construct two types of algorithms in order to determine which ML algorithm is most accurate for use in an SSI. An engineering goal was also developed, aiming to create an SSI with an 80% accuracy using low-cost muscle sensors as opposed to commonly used electromyographs.

III. New Understanding & Conclusions

The classification accuracies and F-scores of the 8 tested models are listed in Table X. It was found that the highest F1-score (0.81) was achieved using the SVM model which is a type of PR algorithm. This means that the SVM Pattern Recognition model is the most accurate ML algorithm to use in an SSI for classifying EMG signals into speech, thus answering the research question. This is the first study to have ever identified the most accurate ML algorithm for classifying EMG signals in an SSI.

The classification accuracy for the SVM model, the best performing model, is 80.1%. This classification accuracy meets the engineering goal of 80% accuracy. This means that the created SSI, which used a Myoware muscle sensor, met the engineering goal and is the first study to have used a muscle sensor to implement an SSI rather than an electromyograph.

IV. Explanation of Findings & Other Research

The SVM model likely outperformed the CNN model as it is a simpler model and can train on data quickly. On the other hand, the CNN, dealing with image inputs, trains slowly requiring more computing power. Because the number of iterations each model could train for was limited, the CNN wasn't able to fully train in the given amount of iterations. This likely resulted in the SVM algorithm yielding a better performance. Additionally the CNN may have performed better if the input images were not distorted. When images were input into the CNN algorithm, a standard process of making slight modifications to the image was applied (shifting the image / zooming into the image). Although these image modifications are used to prevent overlearning, they may have affected the ability of CNN to effectively/accurately classify the precise/particular EMG waveforms.

The PR models tested had a wide range of accuracies. This can be attributed to the fact that some models, such as the Linear Discriminant, divide numerous classes ineffectively with simple methods. Although these algorithms perform well with small inputs, it wasn't useful to classify large EMG signals.

Karlik (2014) found in his study that the CNN is the most accurate algorithm for classifying EMG data for arm prosthesis. This study had different results because the nature of EMG signals from the arm and throat are different resulting in different optimal ML algorithms. Additionally, in this study, time and resource limitations could have impacted the performance of tested ML algorithms.

V. Limitations, Future Research & Implications

One limitation of this study is that it used a small dataset. Due to time constraints, the dataset developed to train/test the ML model contained only 1020

signals. With more training data, the accuracy of the ML algorithms would improve.

Another limitation is that this study cannot definitively identify the SVM algorithm as the most accurate ML model for the classification of EMG signals in an SSI. This is because only PR and CNNs were tested in the scope of this study. There are many types of ML algorithms, such as Artificial Neural Networks, that weren't tested in this study due to time constraints and complexity of models.

Future studies can address these limitations by evaluating/comparing more types of ML algorithms. Additionally, future studies could also develop datasets for a multitude of words from the English language. This would allow for the development of a more complete SSI that can truly be used in the real world.

In this study, vowels were used for training the silent speech recognition algorithms. Silently speaking vowels require the activation of similar muscle groups in the submental triangle and therefore would have impacted the results (decreased classification accuracy). Better results could possibly be achieved with the use of various consonants. Future studies could explore the differences in accuracy when classifying consonants and vowels.

This study's findings can help improve the accuracy of future SSIs by showing that SSIs can achieve better accuracy by using an SVM model. By identifying the SVM model as the best ML model for EMG classification for an SSI, this study could benefit many future EMG classification tasks. For instance, computational resources could be effectively saved by running the less-intensive SVM model - which produces better results. This would result in faster results for devices that rely on signal classification. Additionally, the findings of this study can push researchers to develop SSIs without the use of electromyographs as this study was able to achieve good results using a low-cost muscle sensor.

References

Advancer Technologies. (n.d.). *MyoWare Muscle Sensor Kit.* https://learn.sparkfun.com/tutorials/myoware-muscle-sensor-kit/all.

Alaskar, H. (2018). *Convolutional Neural Network Application in Biomedical Signals.* Journal of Computer Science and Information Technology, 6(2). https://doi.org/10.15640/jcsit.v6n2a5

Arduino. (n.d.). *Arduino Mega 2560 rev3.* Retrieved December 18, 2020, from https://store.arduino.cc/usa/mega-2560-r3

Arslan, Z., Mutlu, T., Elif, O., & Mehmet, B. (2006). *Observations on the characteristics of EMG signals recorded at different depths.*

Ayodele, T. (2010). *Types of Machine Learning Algorithms.* New Advances in Machine Learning. https://doi.org/10.5772/9385

Beukelman, D., & Garrett, K. (1988). *Augmentative and alternative communication for adults with acquired severe communication disorders.* Augmentative and Alternative Communication, 4(2), 104–121. https://doi.org/10.1080/07434618812331274687

Bishop, C., & Nasrabadi, N. Pattern Recognition and Machine Learning. Journal of Electronic Imaging, 16(4), 049901. https://doi.org/10.1117/1.2819119

Bonaccorso, G. (2017, July 24). *Machine learning algorithms.*

Brown, S. A. (2000). *Swallowing and Speaking Challenges for the MS Patient.* International Journal of MS Care, 2(3), 4–14. https://doi.org/10.7224/1537-2073-2.3.4

Char, D., Chung, T., Mckee, A., & Pai, A. (2018, June). *The human keyboard.*

Cohen, M. (2020, November 30). *Signal processing problems, solved in MATLAB and in python [MOOC]. Udemy.* Retrieved August 27, 2020, from https://www.udemy.com/course/signal-processing/.

Denby, B., Schultz, T., Honda, K., Hueber, T., Gilbert, J. M., & Brumberg, J. S. (2010). *Silent speech interfaces.* Speech Communication, 52(4), 270–287. https://doi.org/10.1016/j.specom.2009.08.002

Eremenko, K., & Ponteves, H., DataScience, S. (n.d.). *Machine learning A-Z: Hands-on python & R in data science [MOOC]. Udemy.* Retrieved December 17, 2020, from https://www.udemy.com/course/machinelearning/

Hartman, K. (n.d.). *Getting started with Myoware muscle sensor.* Retrieved December 18, 2020, from https://learn.adafruit.com/getting-started-with-myoware-muscle-sensor

Kapur, A., Kapur, S., & Maes, P. (2018, March 5). *AlterEgo: A personalized wearable silent speech interface.* Retrieved

October 25, 2020, from https://www.media.mit.edu/publications/alterego-IUI/

Kapur, A., Sarawgi, U., Wadkins, E., & Wu, M. (2019). *Non-invasive silent speech recognition in multiple sclerosis with dysphonia.*

Kareem, F., Azeem, M., & Sameh, A. (2017, October). Classification of EMG signals of lower arm (forearm\hand) motion patterns used to control robot hand movement.

Karlık, B. (2014). *Machine Learning Algorithms for Characterization of EMG Signals.* International Journal of Information and Electronics Engineering, 4(3). https://doi.org/10.7763/ijiee.2014.v4.433

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). *ImageNet classification with deep convolutional neural networks.* Communications of the ACM, 60(6), 84–90. https://doi.org/10.1145/3065386

MathWorks. (n.d.). *Googlenet.* Retrieved December 18, 2020, from https://www.mathworks.com/help/deeplearning/ref/googlenet.html

MathWorks. (n.d.) *Classification learner app.* Retrieved December 18, 2020, from https://www.mathworks.com/help/stats/classification-learner-app.html

Mohanty, S. P., Hughes, D. P., & Salathé, M. (2016). *Using Deep Learning for Image-Based Plant Disease Detection.* Frontiers in Plant Science, 7. https://doi.org/10.3389/fpls.2016.01419

Muhammad, U., Wang, W., Chattha, S. P., & Ali, S. (2018). *Pre-trained VGGNet Architecture for Remote-Sensing Image Scene Classification.* 2018 24th International Conference on Pattern Recognition (ICPR). https://doi.org/10.1109/icpr.2018.8545591

Mwebaze, E., & Owomugisha, G. (2016). *Machine Learning for Plant Disease Incidence and Severity Measurements from Leaf Images.* 15th IEEE International Conference on Machine Learning and Applications (ICMLA). https://doi.org/10.1109/icmla.2016.0034

Samuel, A. L. (1959). *Some Studies in Machine Learning Using the Game of Checkers.* IBM Journal of Research and Development, 3(3), 210–229. https://doi.org/10.1147/rd.33.0210

Schultz, T., & Wand, M. (2010). *Modeling coarticulation in EMG-based continuous speech recognition.* Speech Communication, 52(4), 341–353. https://doi.org/10.1016/j.specom.2009.12.002

Scikit-learn. (n.d.). *Scikit-learn machine learning in Python.* Retrieved December 17, 2020, from https://scikit-learn.org/stable/

Tang, P., Wang, H., & Kwong, S. (2017). *G-MS2F: GoogLeNet based multi-stage feature fusion of deep CNN for scene recognition.* Neurocomputing, 225, 188–197. https://doi.org/10.1016/j.neucom.2016.11.023

Wood, T. (2019, May 17). *F-score.* Retrieved December 17, 2020, from https://deepai.org/machine-learning-glossary-and-terms/f-score.

Zia ur Rehman, M., Gilani, S., Waris, A., Niazi, I., Slabaugh, G., Farina, D., & Kamavuako, E. (2018). *Stacked Sparse Autoencoders for EMG-Based Classification of Hand Motions: A Comparative Multi Day Analyses between Surface and Intramuscular EMG.* Applied Sciences, 8(7), 1126. https://doi.org/10.3390/app8071126