Phonetic Classification Using Autoencoder Extracted Features of Low-density EEG Signals

B150864

Word Count: 7997

Masters' Dissertation

Speech and Language Processing

School of Philosophy, Psychology and Language Sciences

University of Edinburgh

2020

Abstract

In this research we explore phonetic classification from EEG signals using the publicly available KARA One and FEIS datasets. We lend particular focus to the feature extraction step in the classification pipeline and compare 3 methods of feature extraction. Our baseline is an expert specified set of mathematical features. Our additional 2 features sets are automatically extracted from the bottleneck layers of a Dense Stacked Autoencoder and a Convolutional Stacked Autoencoder. Additionally, we offer some preliminary insights on the effects of channel placement in low-density EEG processing tasks.

Our comparison between feature sets indicates that manually specified features may be better suited to classifying phonetic information for these datasets. We suspect this outcome is highly influenced by the small size of these datasets, as well as the methods used to gather them, including stimulus design and channel placement. As such, we suggest some future improvements to the data gathering process in order to better use neural network architectures to learn phonetic information from EEG signals.

Keywords

Phonetic Classification, Electroencephalography, Signal Processing, Autoencoders, Feature Extraction

Acknowledgements

I would like to thank my advisors for their time, insight and encouragement. Their guidance has helped me learn an incredible amount in the last 10 weeks. I also thank my fellow MSc student, who worked alongside me in the earlier stages of the project and remained unfailingly supportive thereafter. Lastly, a sincere thanks to the 2019 MSc students who gathered the FEIS dataset and made this project possible.

Table of Contents

1. Introduction

Machine learning for signal processing has made impressive strides in recent years. We have reached a point where decoding imagined speech directly from brain activity is feasible. Researchers in (1), (2), and (3) have managed to reconstruct speech from intracranial electrodes readings.

Electroencephalography (EEG) offers a non-invasive alternative to intracranial electrodes by way of wearable devices which place electrodes against the user's scalp. We are witnessing a rise in commercially available EEG devices, which unlike their medical counterparts, are more portable and easier to use for non-experts. These devices are lower density with regard to electrode count and have to contend with a lower signal-to-noise-ratio than intracranial techniques, but may offer user friendly communication aids for those suffering from neurodegenerative disorders.

Progress toward decoding speech from EEG signals includes classification tasks where we attempt to label unseen segments of an EEG with their phonetic information. A key component of classification is representing the data as some set of features which allow for categorical distinctions to be made between data samples. This research offers an investigation of autoencoder based methods for automatic feature extraction from EEG signals. Autoencoders are a neural network based architecture which allow for unsupervised feature learning. These methods are compared to a supervised baseline of hand selected features. To train our models we use 2 publicly available datasets, KARA One (4) and FEIS (5), which contain EEG recordings of 'imagined' speech.

2. Background

2.1 Electroencephalography

Electroencephalography (EEG) is the measure of electrical activity in the brain. By placing electrodes directly on the scalp, EEG recording devices are able to pick up the differences in electric potential produced by neuron activity in the cortical layers of the brain. This offers a noninvasive technique for mapping brain function which can be used to study a wide array of processes. (6) (7)

Brain activity is distributed in a 3-dimensional space which changes over time. EEG is considered to have high temporal resolution; devices are able to take readings on the order of milliseconds (6). This makes EEG a useful alternative to fMRI which has poor temporal resolution as hemodynamic response time is much slower than the brain's neural processes, causing a blurring effect (8).

2.2 Brain-Computer Interfaces

Brain-Computer Interfaces (BCIs) are alternative methods of human-computer interaction which allow brain activity to be directly translated into commands (9). BCIs can assist those with neuromuscular impairments, such as locked-in syndrome or ALS, which prevent them from using other communication aids that require muscle movement (10).

Ideally, we will be able to produce BCIs which offer direct speech synthesis from readings of user brain activity. This would have major advantages over current spelling approaches, which require users to select individual letters, resulting in slow output. Direct speech synthesis would allow users to communicate at a more natural rate and could capture prosodic elements of speech that are not present in text representations (11).

2.3 Electrocorticography

Electrocorticography (ECoG) is a technique similar to EEG for electrophysiological monitoring, but the electrodes are placed directly on the exposed surface of the brain rather than the scalp. The electrode implantation necessary for ECoG is only undertaken by patients who already require invasive neurosurgical treatment, such as those with epilepsy, brain tumors, or Parkinson's disease (12). Consequently there are far fewer subjects available to participate in ECoG research than are available for EEG research.

Relative to EEG readings, ECoG offers higher spatial resolution in addition to high temporal resolution (13). Due to the direct contact of electrodes with the brain, ECoG signals also provide a better signal-to-noise ratio than EEG (12). These improvements have allowed for some impressive success with ECoG based systems. This includes high-quality speech reconstruction from ECoG recordings of auditory stimulus (1), spoken words (2) and spoken sentences (3). However, The invasive nature of ECoG recordings means these methods do not generalize well. EEG offers a noninvasive measurement which can be similarly utilized in BCIs. The question remains, to what extent are the more practical EEG systems viable for speech applications?

3. Issues with Learning from EEG Data

3.1 Signal to Noise Ratio

Unlike other familiar signals, such as speech and images, EEG signals have a comparatively low signal-to-noise ratio (SNR). The relevant information carried by the signal is confounded by heavy interference from other factors. EEG readings are subject to physiological artifacts, environmental artifacts, and unique hardware issues associated with recording devices (7). This makes extracting features from an EEG signal more difficult than for signal types with a higher SNR (14). We have included explanations of some common issues, but this is by no means a complete list.

3.1.1 Environmental Artifacts

Environmental artifacts include influences from other electrical phenomena or devices during recording. For example, power line interferences can cause spikes in the data at certain frequencies (15). Devices are often alleged to automatically filter these interferences, but reality is not necessarily as advertised. Figure 1 presents a power density spectrum of a reading taken from FEIS participant data gathered using the Emotiv Epoc+. It demonstrates a clear spike at 50 Hz, despite the device claiming to filter this interference. This spike does not occur in all participants, so the filtering does appear to work some of the time.

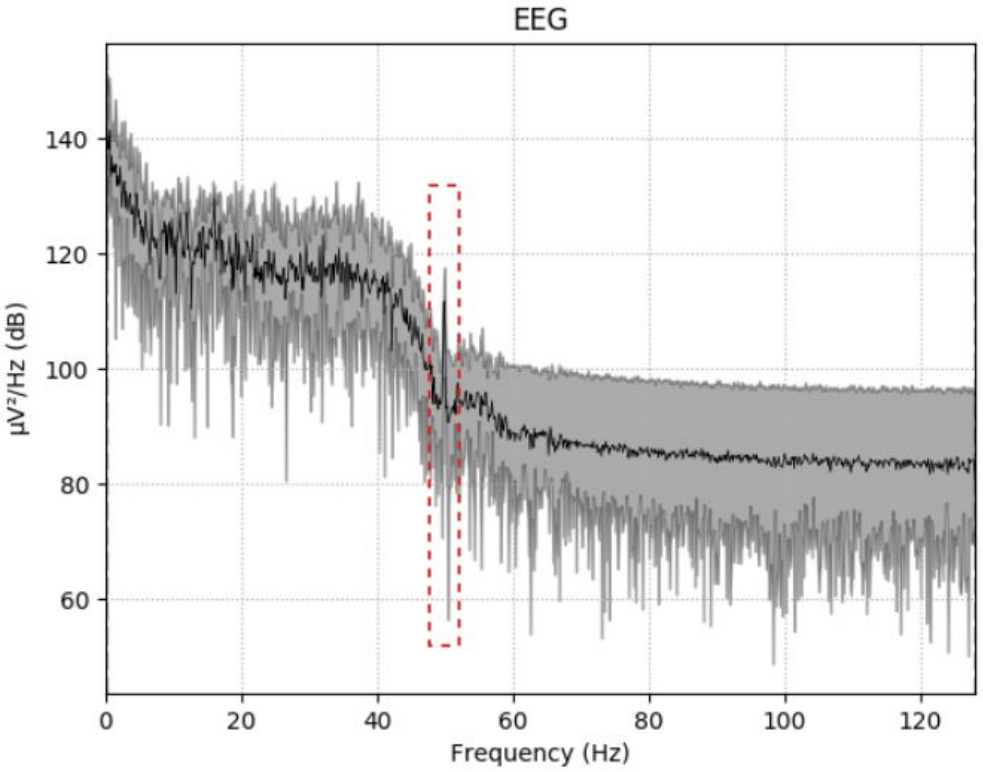

Figure 1. The power density spectrum of a reading taken from participant 6 in the FEIS dataset. The noticeable peak at 50 Hz indicates electrical interference from the surrounding environment at the time of recording.

3.1.2 Physiological artifacts

Physiological artifacts are the result of other biological processes occurring in the participant, which influence the EEG signal. Examples of physiological artifacts include eye movement (such as blinking), cardiac activity, respiratory activity, or any kind of muscle movement (16). Ocular artifacts, in particular, are very common in EEG data and have such a substantial effect on the EEG spectrum, that removal of blinking artifacts has become an important area of research in EEG signal preprocessing (17).

3.1.3 Signal Smearing

Part of the reason EEG offers a less informative signal than ECOG is that EEG signals have to contend with passing through matter in the brain, skull, and scalp before reaching the

electrodes. These factors work to spatially smooth the signals produced by the brain, which are mixed when they finally arrive at the sensor. This signal smearing results in electrodes receiving highly correlated information (6). Finding methods to unmix these signals is an active area of research (18,19).

3.2 Inter-subject Variation

While subject-independent systems are the ideal endpoint for most EEG research, subject-dependent systems are sometimes a more feasible starting point due to extreme inter-subject variation. This variation is the result of different brain physiologies between participants. A classic example of this is the lateralization of language processing in the brain and its relationship to a person's dominant hand (20). More recently, researchers have found evidence that a speaker's proficiency in a language affects the distribution of language processing activity in their EEG recordings (21).

We must also consider the interaction between our recording devices and the shape of a subject's skull. With some devices, such as the aforementioned Emotiv Epoc+, electrode placement is predetermined by the design of the device. These sorts of devices will inevitably fit differently on different subjects.

Lastly, due to the extreme sensitivity of EEG recordings, different recording sessions with the same individual can vary. A classifier built on subject dependent data recorded at one period in time could generalize poorly to data recorded at another period in time due to small inconsistencies between sessions (7). Variation between recording sessions is a difficult challenge to overcome because, while we need large datasets to better train our models, we can not record for so long that we exhaust our participants.

3.3 Limited Training Data

Currently, there are not many publicly available datasets with which to build models for EEG tasks. According to a survey of studies using EEG data performed in 2019, 42% of the papers

reviewed did not use publicly available data (7). Further, gathering EEG data is an arduous and time consuming process for both the researchers and participants. As a result, the datasets which are available tend to have only small amounts of data, especially for participant-dependent tasks. Moreover, this data is gathered on a variety of different equipment and using variable experimental designs. Consequently, it is very difficult to find pre-existing data which is suited to a given EEG task, especially in the realm of speech, where research is quite new.

4. Motivation and Research Questions

In this research we attempt to classify between phonetic categories using imagined speech from 2 publicly available datasets. Our EEG analysis involves 3 broad steps: (1) data segmentation; (2) feature extraction; (3) classification.

For this research, we are primarily interested in the feature extraction step. The goal of feature extraction is to obtain meaningful information from the EEG signals which can be used to distinguish between classes at classification time. Features can be handcrafted by an expert or learned automatically from the signal through a neural network. Handcrafted features can be time consuming and labor intensive to design as they require detailed knowledge of the task and the data. Ideally, features learned with neural networks can offer researchers an unsupervised alternative to expert specified features.

Our primary goal is to determine whether autoencoders can automatically extract viable features from low-density EEG data to use in phonetic classification tasks. We compare these features to a baseline set of manually specified features. Additionally, we pursue 3 subgoals:

1. To compare a convolutional stacked architecture and a dense stacked architecture.
2. To consider the relevance of the FEIS and KARA One datasets for this task and how data gathering methods may affect results.
3. To investigate how the scalp locations of the channels may affect the task.

5. Methods

5.1 Datasets in this Study

FEIS and KARA One are 2 publicly available datasets for imagined speech gathered using EEG devices. The sets are fairly comparable in their experimental design as the FEIS researchers took inspiration from the KARA One researchers, with some notable differences. Both datasets were designed with classification of isolated phonemes in mind. (4,5)

5.1.1 KARA One

The KARA One researchers recorded their dataset using a 64-channel Neuroscan Quick-cap. Electrode placement follows the 10-20 system. The 11 prompts used by the researchers are listed in Table 1. The collection method for a given participant consisted of roughly 15 trials for each prompt. Each trial consisted of 5 consecutive segments:

1. A 5 second resting state where the participant relaxed and cleared their mind.
2. A stimulus state where the prompt would appear as text on the screen and a single instance of the prompt was played over audio.
3. A 2 second phase where the participant moved their articulators into position as if they were going to speak the prompt.
4. A 5 second imagined speech state where the participant imagined speaking the prompt.
5. A speaking state where the speaker spoke the prompt aloud. (4)

We resampled the data from 1000 Hz to 250 Hz as suggested in (22) to be more similar to FEIS data. We then sub-selected 2 assortments of 14 channels from the overall 64 channels. The first subset contains the same 14 channels as the FEIS dataset (Set A). Selection for the alternative subset of 14 channels was inspired by (4) in which the researchers calculated Pearson correlations between electrodes and acoustic features (Set B). This effectively turns

our KARA One dataset with 64 channels into 2 separate lower density datasets, each with only 14 channels. The intent is to allow for preliminary investigation into the effect of channel location on these feature learning and classification tasks. The different assortments of channels are visualized in Figures 2 and 3.

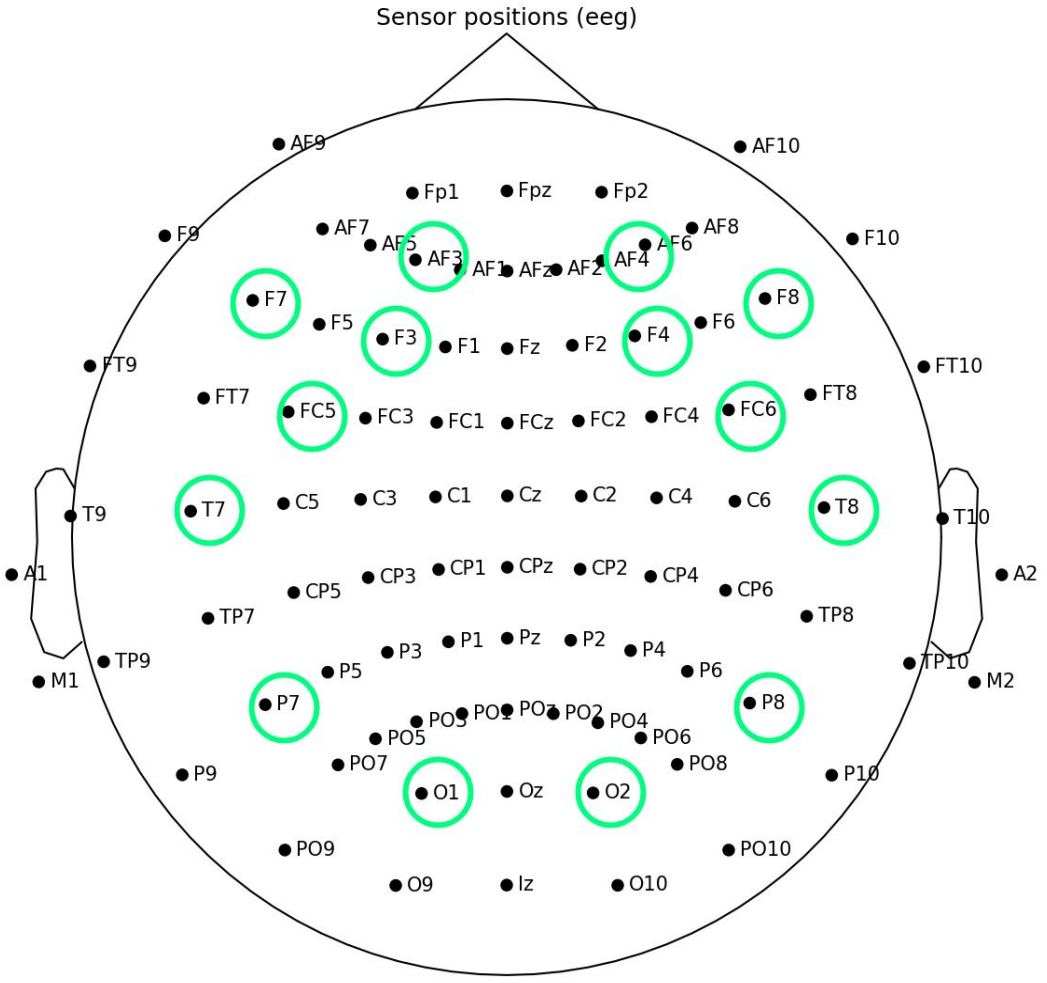

Figure 2. The 14 channels used for the FEIS data and Set A of the KARA One data. Channels are projected to approximate locations on a participant's scalp. The montage was adapted from an image in the MNE python package (23).

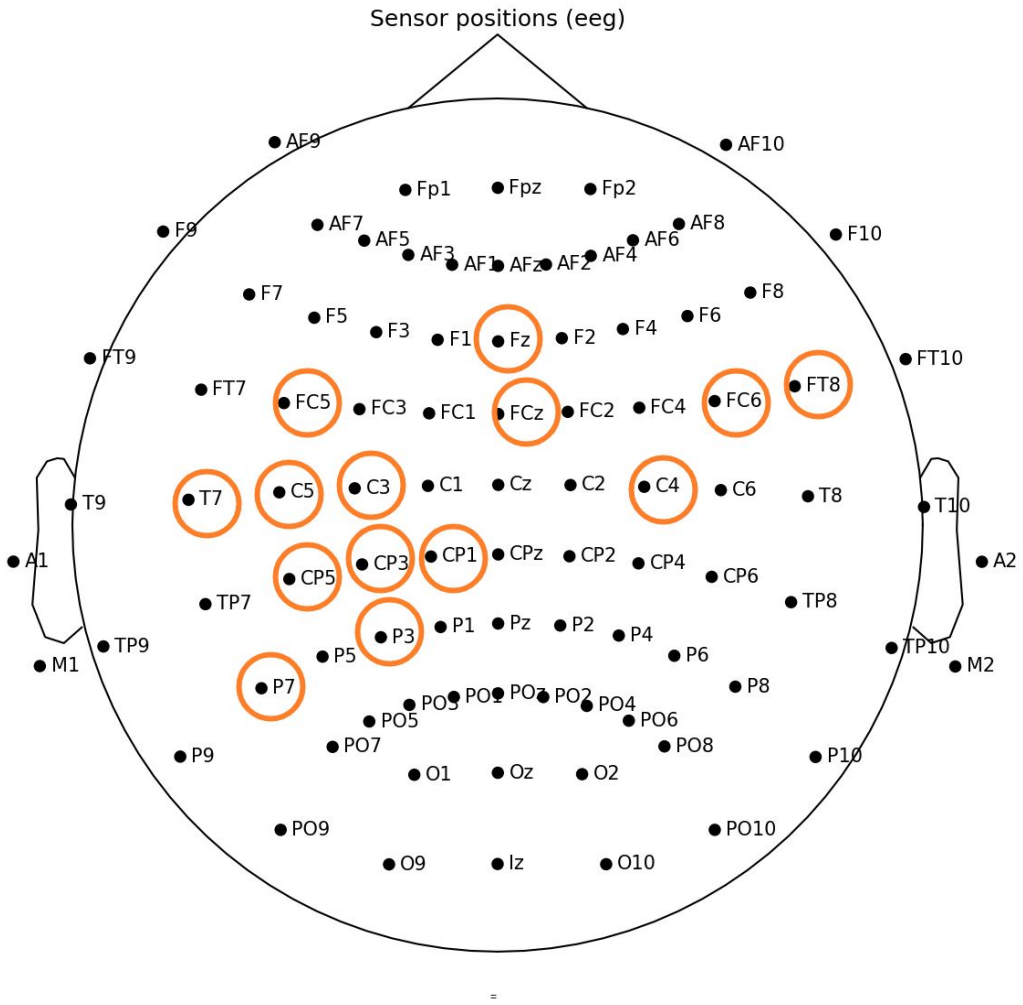

Figure 3. The 14 channels used in Set B of the KARA One data. Channels are projected to approximate locations on a participant's scalp. The montage was adapted from an image in the MNE python package (23).

5.1.2 FEIS

This dataset is recorded using a 14-channel Emotiv Epoc+ device. Electrode placement follows the 10-20 system. The FEIS data was sampled at 256 Hz, the maximum sampling rate for the device. The 16 prompts used by the researchers are listed in Table 1. The collection method for a given participant consisted of 10 trials for each prompt. Each trial consisted of 5 consecutive segments:

1. A 5 second resting state where the participant relaxed and cleared their mind.

2. A 5 second stimulus state where the prompt appeared as text on the screen and 5 instances of the prompt were played over audio at a consistent interval.
3. A 2 second phase where a focus point appeared on screen.
4. A 5 second imagined speech state where the participant imagined speaking the prompt 5 times.
5. A 5 second speaking state where the participant spoke the prompt aloud 5 times. (4)

FEIS Prompts	
Consonant Phonemes	Vowel Phonemes
/m/ /n/ /ŋ/ /f/ /s/ /ʃ/ /v/ /z/ /ʒ/ /p/ /t/ /k/	/i/ /u/ /æ/ /ɔ/

KARA One Prompts	
Isolated Sounds	Full Words
/iy/ /uw/ /piy/ /tiy/ /diy/ /m/ /n/	/pat/ /pot/ /knew/ /gnaw/

Table 1. Prompts used during data gathering of the FEIS and KARA One datasets. Note that plosives in the FEIS prompts are followed by a neutral release (ə)

5.1.3 Notable Differences Between the Datasets

1. The number of initial input channels (prior to our sub-sampling of the KARA One set).
2. The differences in prompts, which leads to different types of dataset splits in the binary tasks discussed later.
3. While KARA One data includes a ‘preparing articulators' stage, the FEIS researchers opted for a fixation point. They believed that having the participants prepare their articulators would pollute the imagined speech data with articulator movement artifacts (5).

5.1.4 Ultimate Data Representation and Dimensionality

For both the FEIS and KARA One datasets we use only the ‘thinking’ trials. This leaves us with 3 datasets of similar size, all with 14 channels and a sampling rate of either 250 Hz or 256 Hz. For a given participant, the data takes the form of a 3-dimensional matrix where the

prompt trials, the input channels, and the time domain samples each represent a dimension (see Figure 4).

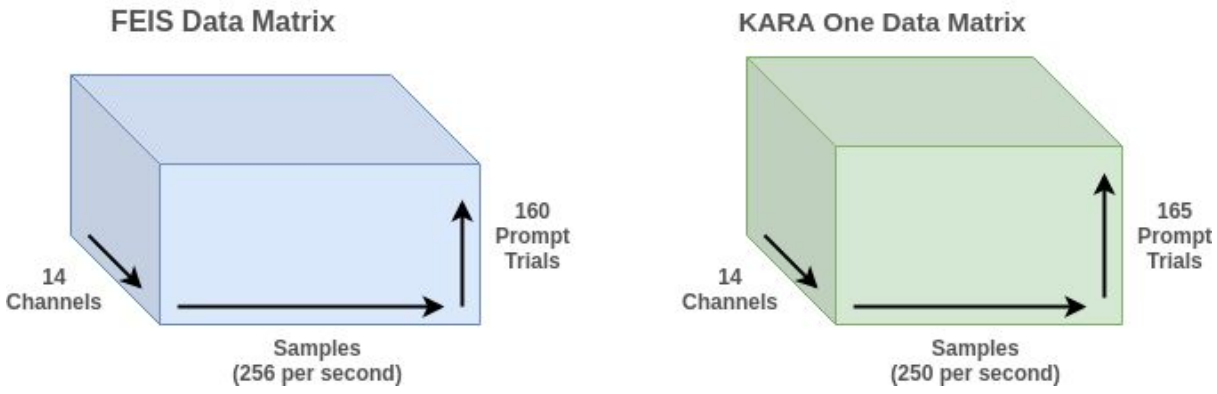

Figure 4. The dimensions for a given participant's data. Note the 'prompt trials' dimensions consist of the number of prompts multiplied by the number of experimental trials per prompt.

5.2 Signal Preprocessing

Given the low SNR of EEG data, many methods of preprocessing have been developed to clean the signal before further processing. Prior to our experiments with feature sets, we investigated some automatic preprocessing techniques. These methods were informed by the aforementioned research of common artifacts in EEG data. We tested them using the FEIS dataset and our baseline feature selection method (section 5.5.1).

5.2.1 High-pass Filter

First we explored a simple high-pass filter at 1 Hz. This filter is commonly used in order to remove slow drift from the signal by blocking the affected frequencies (in this case a frequency below 1 Hz). However, high-pass filtering should be applied with caution, as it may easily create artifacts of its own as shown in (24). With this data, a highpass at 1 Hz had variable effects across all subjects, resulting in improved baseline accuracies for some participants and poorer scores for others. This is unsurprising, given how much data can vary between participants.

5.2.2 Bandpass Filter

Similar to the highpass filter at 1 Hz, we also explored the use of a bandpass filter, which was set to allow only frequencies between 1 Hz and 50 Hz. This filter was inspired by (4) in which the researchers applied it to the KARA One dataset prior to further processing. When applied to the FEIS data we see similarly mixed results as occur with the highpass at 1 Hz.

5.2.3 ICA to Remove Ocular Artifacts

Independent Component Analysis (ICA) is a technique used to separate multiple source (component) signals from a complex mixed signal. When preprocessing EEG data, ICA is often used to isolate ocular artifacts and remove them from the signal prior to further processing. Typically this process is aided by designated EOG channels on the device which are placed near the eyes to best detect ocular movement. However, the Emotiv Epoc+ device used to collect the FEIS dataset does not have these EOG channels. The next best option is to temporarily use the 2 frontal channels (AF3 and AF4) to approximate EOG channels for the sake of removing ocular artifacts.

We implemented ICA using the methods outlined in the MNE python package (23), which detects EOG artifacts based on a Pearson correlation between the data samples and the EOG channels. This implementation requires the specification of a variance threshold by the researcher, which defines the variance above which a component is classified as an outlier. Setting this parameter, especially on subject independent data, is a difficult judgment to make. If the threshold is set too high, ocular artifacts will be ignored, while setting the threshold too low can lead to unrelated components being removed. We argue there is no perfect threshold for the entire FEIS dataset, as performance is highly subject dependent. For our test, the threshold was set to the default of 3 and any EOG components found by the ICA were automatically removed.

This method works with some success, though the overall effect of attempting to remove EOG artifacts from the FEIS data led to similarly mixed results across participants. For some

participants, no EOG artifacts were found. In others, the removed components led to either a decrease or increase in baseline accuracy. Results could be improved by visual inspection of the component plots to spot blink artifacts rather than reliance on the process used by MNE, but this is a time consuming process.

5.2.4 Final Input Data

Ultimately, we decided to forgo preprocessing for these feature extraction experiments. The changes in accuracy for a given preprocessing method were highly subject dependent, but the average accuracy across all participants remained roughly unchanged. These preprocessing tools are rather blunt methods. Though they achieve good results in some domains, it is difficult to say what information about speech we might be losing by indiscriminately applying these techniques to all of the data (25).

Further, artifact removal techniques usually require the intervention of a human expert. Cleaning EEG signals can quickly become a time-consuming process for a researcher. We wish to explore how our feature selection methods can perform on unprocessed, time series input. It is worth seeing whether neural network architectures can learn from this noisy, raw data, prior to attempting more advanced preprocessing techniques (7).

5.3 Windowing the Data

Classification tasks require that our EEG signal be windowed into individual training samples of some length. That length could span the width of an entire trial (in this case, roughly 5 seconds of a repeated prompt), or we could select crops of this signal. For our research, training on an entire trial is unfeasible, as there is very little data for each participant. Instead, we follow the examples set in (14) and (4) and use windows shifted through the trials. This strategy removes the option of learning from the global temporal structure of the complete trial and forces the models to find features that are shared among crops of a given trial type (14).

While this is necessary given the small amount of data, it is also a rather naive approach to creating samples. Features calculated from a window of time series data will be sensitive to the consistency of the slices and their relationship to the onset of the stimuli. Unfortunately, a brain's response onset is not likely to occur at consistent intervals across the data. Without explicit labels of onset timing, we are assuming that even segmentations manage to be decent representations of trial instances. Future research may benefit from progress in adaptive windowing techniques for EEG data such as those proposed in (26).

5.3.1 Averaging

In some cases, averaging could be used to enhance the signal-to-noise ratio of EEG windows. Averaging has been shown to reduce neural noise in samples and improve reconstruction accuracy (1). Typically this would involve data where the event of interest is repeated a given number of times. Then, we could take our windows (each one centered on an instance of the repeated events) and average them into a single representation (27). However, this process is not reasonable for the FEIS and KARA datasets. First, because time domain averaging will only work well if windows are appropriately time locked to their stimulus event, which we cannot assume of our naive cropping method. Further, we have so little data to begin with, that averaging the events in a trial would reduce our data by a factor of 5.

5.3.2 Windows Used in this Research

The training instances used in these experiments are unaveraged, evenly sliced windows of the data. Manually specified features use slices as suggested by (4) and later used by (22). Windows are roughly 10% of the trial with a 50% overlap between adjacent windows.

For the autoencoder input, the windows are larger. Each trial is sliced into 5 even windows with each window theoretically accounting for a single stimulus event. As noted earlier, this assumption is an ideal, and not likely to be the reality.

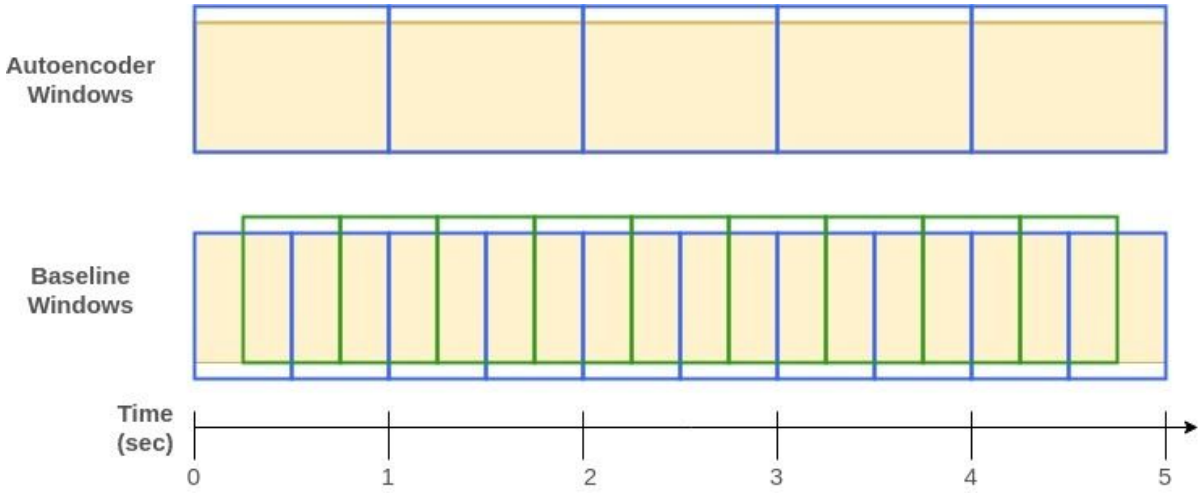

Figure 5. Our 2 window segmentation methods for a given trial of EEG data.

5.4 Classification Task and SVM Build

Our goal is to assess different feature sets for use in binary phonetic classification tasks. The classification for these experiments is always performed using a Support Vector Machine (SMV). The SVM projects the training data into a multidimensional space based on the feature set and then finds a hyperplane which separates data points of different classes with the largest margin.

Our SVM model is inspired by the build presented in (22), who adopted their architecture from (4). It has a Radial Basis Function kernel and is implemented using the Scikit-learn python package (28). The SVM build is constant across all experiments and is the sole classification method used in this research. However, tuning for values of Gamma and C parameters were undertaken separately for each experiment. We tuned for optimal values of C in the range of [1 : 100000] and for Gamma in the range of [1e-4 : 1e-9]. The tuning process used grid search with a cross-validation partition of 20/80. We tuned for optimal F-score, rather than accuracy.

Tuning for F-score highlighted some serious issues with class imbalance in the datasets. In cases where one class outnumbered another, the SVM was incapable of performing any valuable discrimination and would consistently predict the label of the larger class.

Artificially balancing the classes by sub-selecting from the prompts proved to be a simple, but functional way to solve this problem. Unfortunately, this cut down the number of training samples significantly. Perhaps, in future work, there will be better ways to balance the data that involve augmenting the smaller class rather than cutting the larger class. Table 2 contains the prompt splits used for each dataset.

FEIS Prompt Splits	KARA One Prompt Splits
Vowel / Stop	-/+ Back Vowel
Vowel / Nasal	-/+ Bilabial
Nasal / Stop	-/+ Nasal

Table 2. The prompt splits for each dataset used in our classification experiments.

5.4.1 Evaluation Metric

Accuracy is a basic measure of classifier performance. It is most informative when the classes have been balanced (same number of samples per class), which we have done with our data. We evaluate our classifier (and by extension, our feature sets) using accuracy with 5-fold cross-validation. This means our input data is partitioned 5 times into splits of 80% training data and 20% test data. For each of the 5 partitions, the SVM is trained with the training samples and accuracy is computed using the test samples. We end up with 5 accuracy scores which are averaged to get a final score. Cross-validation was implemented using the Scikit-learn Learn package (28) and partitions were kept constant across experiments.

5.5 Feature Selection for Classification

In order for our aforementioned SVM to perform phonetic classification we must represent our EEG data with some set of features. Each instance of data fed to our SVM is composed of its binary label and some feature set, which is computed from the two-dimensional EEG reading (time x channels).

There are a large number of potential features which can be used in a classification task. A larger feature set leads to a larger number of parameters our classifier is forced to optimize. Ideally, we select a set of features which is enough to be discriminative, while not forcing our SVM to handle too many dimensions. A smaller feature set can also help to reduce overfitting and improve the generalizability of our model. (9)

Among the numerous features which can be extracted from an EEG signal, only some will be relevant to a given classification task. It is the researcher's job to devise a way to select a subset of useful features.

We offer a comparison of 3 feature selection methods for our 3 datasets:

- A baseline using manually specified mathematical features
- Features extracted from a Dense Stacked Autoencoder (DSAE)
- Features selected from a Convolutional Stacked Autoencoder (CSAE)

5.5.1 Baseline Feature Selection

As a baseline method for feature selection, we calculate a set of 27 mathematical features as well as delta and double delta features for each window. Features are calculated independently for each channel and then this two-dimensional array (features x channels) is flattened into a one-dimensional feature representation. This methodology closely follows (22) and borrows from skeleton code provided by the researcher. It should be noted that (22) adopts its methodology from (4), but adds 9 additional mathematical features taken from the EntroPy python package (29). A full list of mathematical features can be found in Appendix 10.2. The ultimate representation for a window of the data is:

(27 features + 27 delta features + 27 double delta features) * (14 electrodes) = 1134

The feature set is reduced prior to its input into the SVM using a k-best selection method as provided in the Scikit-learn Learn package (28). We selected the 10 features most highly correlated with a given binary classification task on a given participant's data.

5.6 Autoencoders

At its most basic, an autoencoder is a variant of a feedforward neural network in which the target output is identical to the input. It is generally composed of symmetrical encoder and decoder structures. The purpose of the encoder is to deconstruct the input data into a hidden representation. The decoder is then responsible for taking this representation and reconstructing the original input.

The autoencoder is trained using the same techniques as other neural network architectures, namely gradient descent and backpropagation. This process trains the network parameters to minimize the difference between the input data and its reconstruction at the output. The loss between the continuous input data and the continuous target is typically measured using Mean Squared Error. (30)

5.6.1 Autoencoders for Feature Selection

Autoencoders have become useful tools for feature learning due to their ability to map data into a latent representation. This quality can be appealing relative to hand selected features as it requires less expertise in the feature design process and can be built by those with a good sense of machine learning and a good sense of the data. Further, since autoencoders are a type of neural architecture, they continue to benefit from the generalizability of many neural network techniques and will profit from further research in deep learning across all fields. We will see an example of this later, in the discussion of convolutional autoencoders (section 5.6.3).

Thus far, autoencoders have been successfully used across multiple domains to extract features from EEG data. For example, in (31), Wen and Zhang use an autoencoder with convolutional layers to perform unsupervised learning of EEG signals in epilepsy. In (32), Li et al. use multiple types of autoencoders (including an interesting Variational Autoencoder) to learn features from a multichannel EEG and then leverage those features in emotion recognition tasks.

In order to learn more useful features of the input data, autoencoders are typically constrained in some way, so that they are unable to perfectly copy the input. Ideally, this forces the model to prioritize properties of the input data which are salient and consistent across training samples. (30)

A simple constraint often applied to an autoencoder is a bottleneck layer which reduces the dimensionality of the input. This is called an 'undercomplete' autoencoder. Essentially, we train an autoencoder to reproduce some set of data, but by applying a bottleneck constraint the encoder is forced to learn a set of salient features for the data (30). We can then pass new data instances to just the encoder, taking the output of the final encoder layer to be our feature representation of the instance.

Historically this technique has been successfully used for dimensionality reduction of data (30), which, as mentioned earlier in the discussion on SVMs, can be necessary so allow models to represent information in fewer dimensions. Both the dimensionality reduction and feature learning qualities of an autoencoder make it a viable way to build a feature space for classification tasks.

Past research has determined that an undercomplete autoencoder without nonlinear activation functions is able to learn a representation of data very similar to features learned by Principal Component Analysis (33). It follows that an autoencoder with nonlinearities (provided by activation functions in our hidden layers) could learn a more complex, nonlinear generalization of Principal Component Analysis. This is true, even with a single hidden layer in each of the encoder and decoder. Yet, as with other neural network architectures, we can extend our autoencoder to have multiple layers. This stacking of layers allows for hierarchical feature learning where each deeper layer can learn a more abstract representation of the data from the previous layer. (30)

5.6.2 Dense Stacked Autoencoder

The Dense Stacked Autoencoder (DSAE) is composed of a symmetrical encoder and decoder, each with 3 fully connected (dense) layers and nonlinear activation functions. The

layers in the encoder get progressively narrower, leading to a central bottleneck layer. Ultimately, this encoder takes our large input (a flattened 1-dimensional representation which contains the samples from all 14 channels for a given window) and reduces it to a set of 10 features at the point of the bottleneck. Our decoder takes the 10 features from the encoder and, using progressively larger layers, reconstructs the data as output of the same size as our input. The initial input is raw time series data which is normalized using z-score normalization. Figure 6 visualizes the architecture and Table 3 provides the necessary parameters.

Loss function	Mean squared error
Optimizer	Stochastic gradient descent
Learning rate	0.05
Activation Function	ReLU
Epochs	200
Weight initialization	Xavier (Glorot) Uniform

Table 3. Parameters for our Dense Stacked Autoencoder

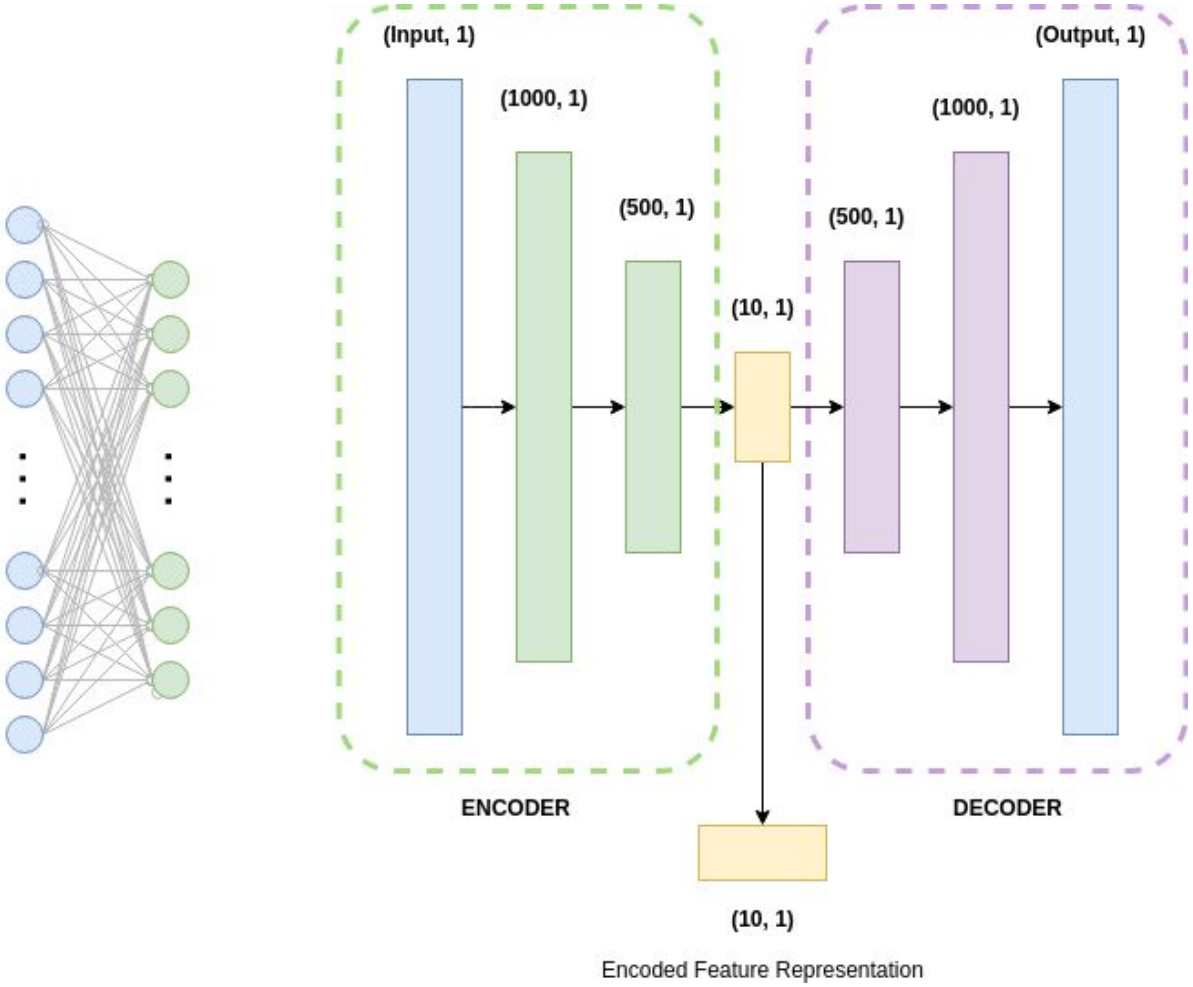

Figure 6. The architecture for our Dense Stacked Autoencoder. The figure on the left represents a slice of the encoder, demonstrating the fully connected input layer units and hidden layer units.

5.6.3 Convolutional Stacked Autoencoder

In addition to the dense stacked architecture, we offer a convolutional stacked autoencoder (CSAE). Our DSAE exclusively contains fully connected units in its layers, which forces every feature to be global as each neuron has access to the entire width of the input from the previous layer (34). Convolutional layers offer a different approach in that the neurons are constrained to view only a subset of the input from the previous layer. The width of this subset is defined by a particular kernel size provided by the researcher. Similar to the previously discussed bottleneck layers, these convolutional kernels can be viewed as a constraint to the autoencoders process of deconstructing and reconstructing input. The intuition is that this constraint will encourage the model to learn more localized features.

Thus far, convolutional autoencoders have been successfully used in many image processing tasks (34) (35) (36) (37), to extract features from audio signals (38) and even to denoise audio data (39). Stacked convolutional architectures are thought to perform well for these sorts of natural signals due to their ability to represent their inherent hierarchical structure (14). Each convolutional layer can learn higher level features from compositions of the lower level features encoded by previous convolutions. Further, recent work with convolutional architectures has demonstrated their ability to represent time series data well and in some cases even outperform RNNs for sequence modeling tasks (40). As such, convolutional layers seem a natural starting point for a more complex autoencoder architecture to use with our time series EEG data.

EEG signals have characteristics which make them different from handling typical image or audio data. Unlike a speech signal, EEG is obtained from a 3-dimensional scalp surface with multiple inputs (14) and unlike static images, EEG has a time dimension. Consequently, it can be difficult to extend the neural architectures used for these types of data to EEG based tasks.

Our CSAE structure takes inspiration from the design implemented in (31) where a similar system was used with EEG input for epilepsy detection. The architecture combines multiple convolutional layers with a dense bottleneck layer. The hope is that the convolutional layers can learn useful representations of the time series data, while the dense bottleneck is used to reduce the dimension of the output to 10 features for better use in classification. Similar to the DSAE, the input is represented as a flattened 1-dimensional representation which contains the samples from all 14 channels for a given window. Again, this raw data is normalized using z-score normalization. Figure 7 visualizes the architecture and Table 4 provides the necessary parameters.

Loss function	Mean squared error
Optimizer	Stochastic gradient descent
Learning rate	0.1
Activation Function (Convolutional Layers)	ReLU
Activation Function (Dense Layer)	None (linear)
Training epochs	20
Weight initialization	Xavier (Glorot) Uniform
Kernel Size	3
Strides	1

Table 4. Parameters for our Convolutional Stacked Autoencoder

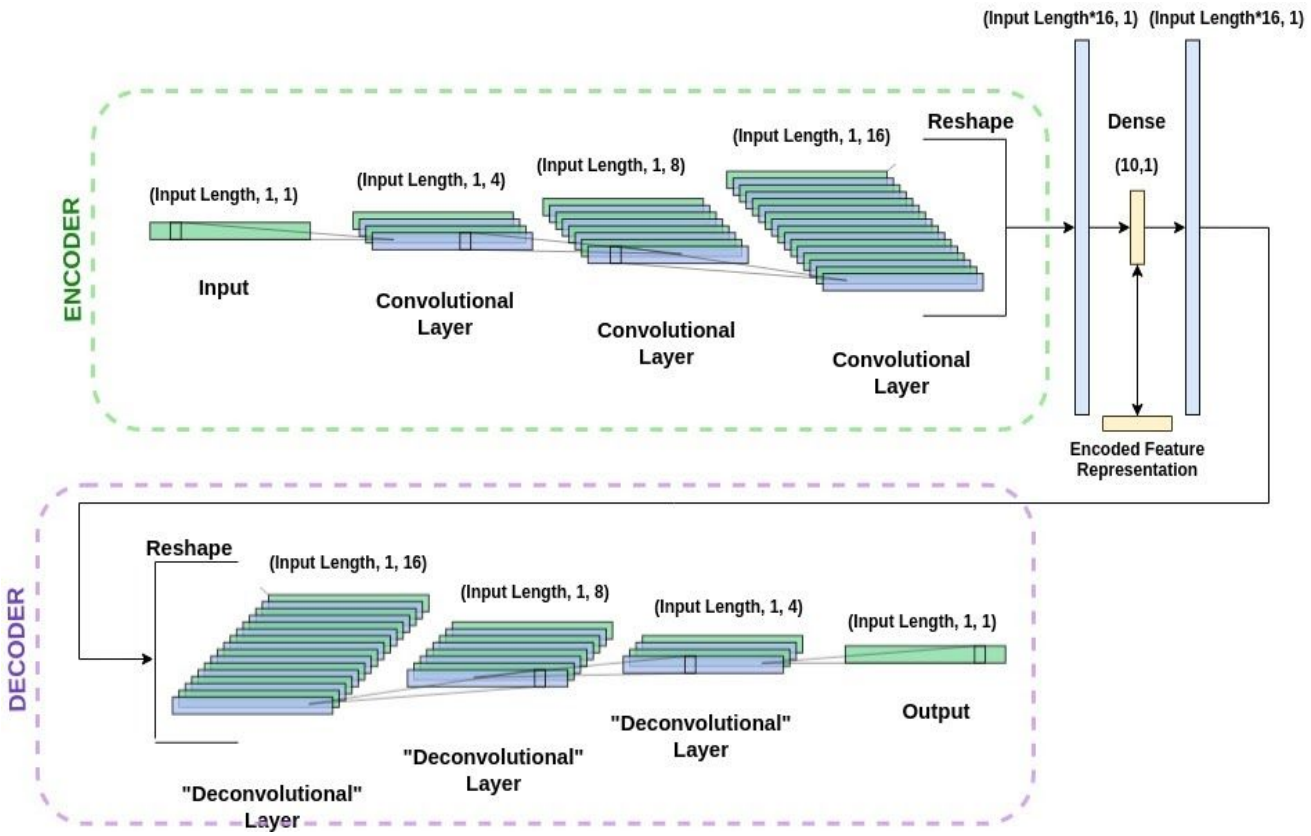

Figure 7. Architecture for our Convoltutional Stacked Autoencoder. Adapted from the NN-SVG neural network visualization toolkit (41).

5.7 Further Notes on Autoencoder Design

5.7.1 Using a Validation Set

Parameters for the aforementioned autoencoders were tuned to produce the best accuracy scores for the classification tasks. This tuning process is somewhat different for an autoencoder than it would be for an end-to-end neural network classifier. In an end-to-end classifier, a validation set is a valuable tool for tuning model parameters as at each training step it gives you an indication of a model's classification performance on unseen data. However, for an autoencoder, a validation set does not give us an indication of classification performance, but a sense of how well the decoder is able to reproduce the encoded input.

A validation set is still valuable in that it allows us to verify that the autoencoder is learning to represent the input. It is obvious when the autoencoder is not learning to represent the data as our validation loss will not lower effectively over the course of training. We can plot these validation scores to inspect the point at which the model converges. This can be useful in setting the learning rate and number of epochs. Trial and error combined with visual inspection of loss curves proved useful for understanding the point at which an autoencoder had learned effectively to represent the data without overfitting to it. For these models, we used a validation set which was 10% of the available training data for a given speaker.

After tuning with the validation set, it is necessary to process the extracted features through the SVM classification portion of the pipeline as we need to see if the autoencoder is learning valuable representations of the data for our phonetic classification tasks. We can then tune autoencoder parameters such as kernel size and stride to improve this classification output. In our research, each experiment was run for 5 constant seed values. The average accuracy across these seed values was taken as the final result.

5.7.2 Weight Initialization

In our autoencoders we use Xavier uniform weight initialization. This method sets the initial weights of our autoencoder by randomly selecting values from a uniform distribution with a mean of 0 and a variance of

$$\frac{2}{Nin + Nout}$$

where *Nin* is the number of neurons in the previous layer and *Nout* is the number in the current layer. The purpose is to ensure that the variance of the outputs is roughly equal to the variance of the inputs and, in turn, make the model more resistant to exploding and vanishing gradient problems (42).

Some literature suggests using He initialization as an alternative to Xavier initialization for layers with a ReLU activation function (43). While this is a valuable point to note, we found that this architecture performed better with this dataset when Xavier initialization was used. More study is necessary to determine the complex interactions between activation functions and weight initialization for EEG data.

5.7.3 Pooling Layers

In research with convolutional architectures (mainly for image based tasks) we tend to see convolutional layers followed by pooling layers. Pooling is commonly used as a technique to reduce overfitting and force learned features to be more generalizable to unseen data (34). Further, they act as a method of reducing the size of the input (by some specified factor).

We initially built our CSAE with max-pooling after each convolutional layer inspired by previous research (31). However, we could not readily justify its purpose in this architecture and so removed the max-pooling layers, which proved to have no ill effect on the accuracy outputs. This is not to say that pooling techniques are not valuable for handling EEG data, only that they did not appear to be relevant for these data, architectures, and tasks. It would

be worthwhile in future research to investigate pooling techniques and how they interact with an autoencoder's ability to represent EEG data.

6. Results

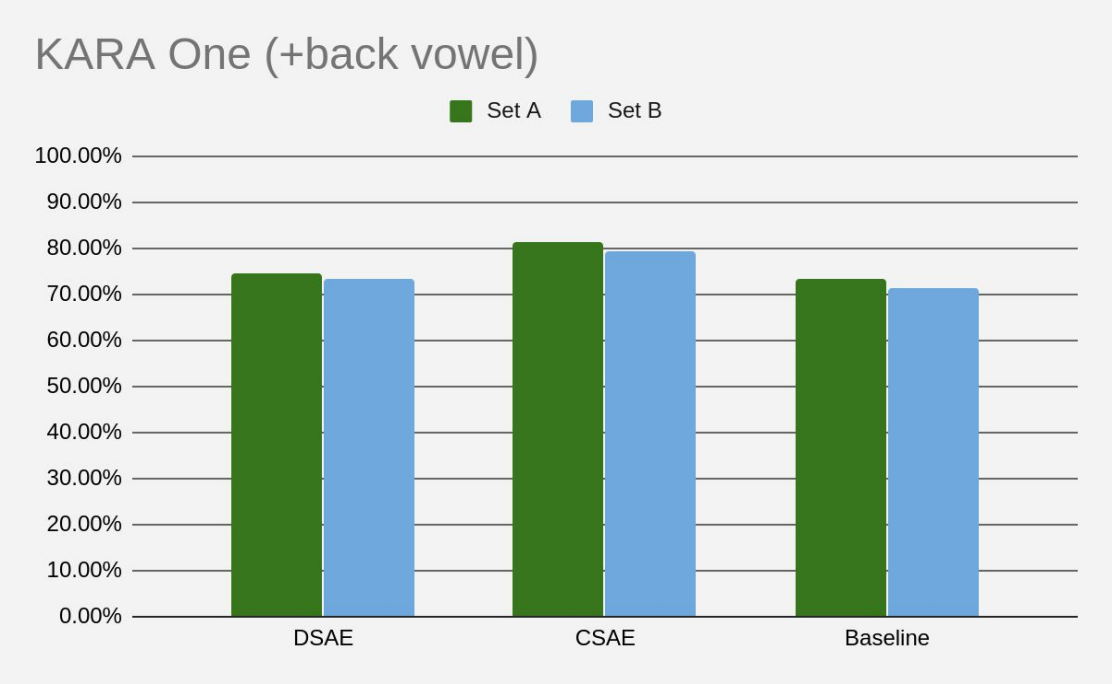

Figure 8. Average accuracy results across the 9 KARA One participants for the 'presence of a back vowel' classification task. Complete table of values in Appendix 10.1.

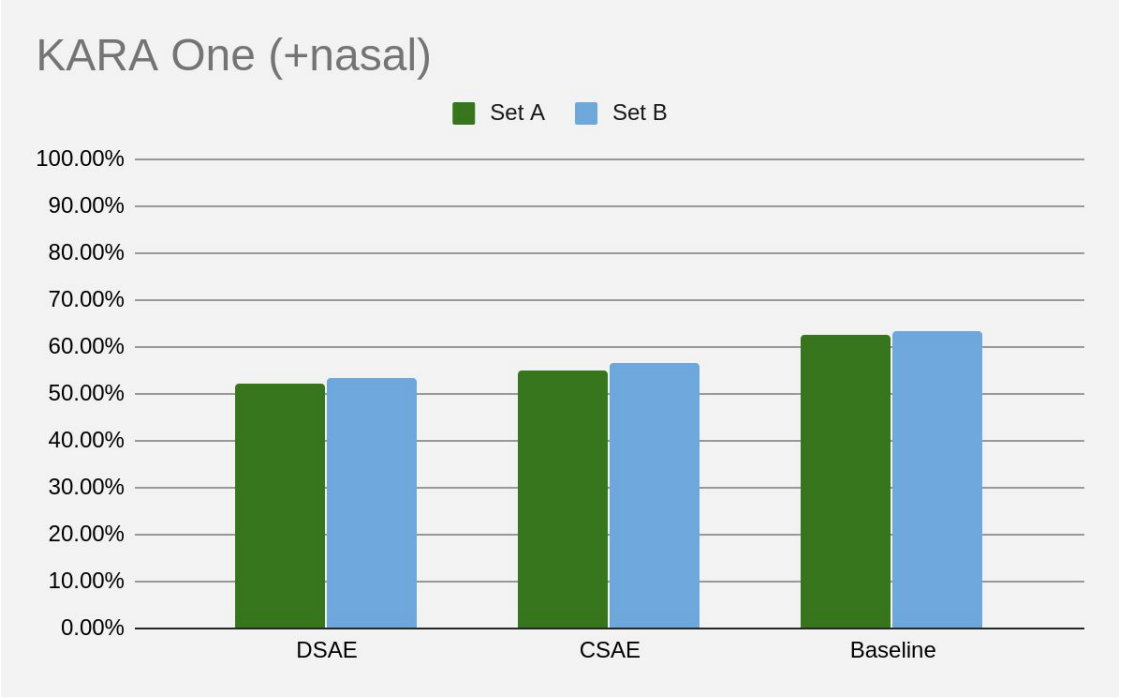

Figure 9. Average accuracy results across the 9 KARA One participants for the 'presence of a nasal' classification task. Complete table of values in Appendix 10.1.

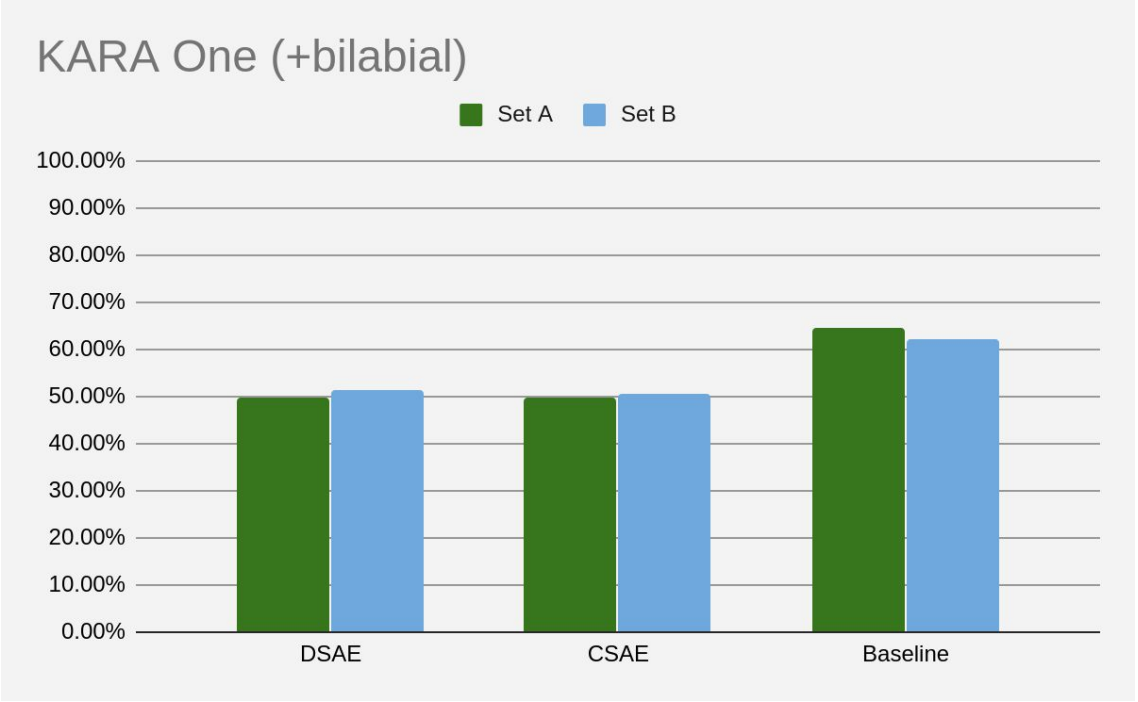

Figure 10. Average accuracy results across the 9 KARA One participants for the 'presence of a bilabial' classification task. Complete table of values in Appendix 10.1.

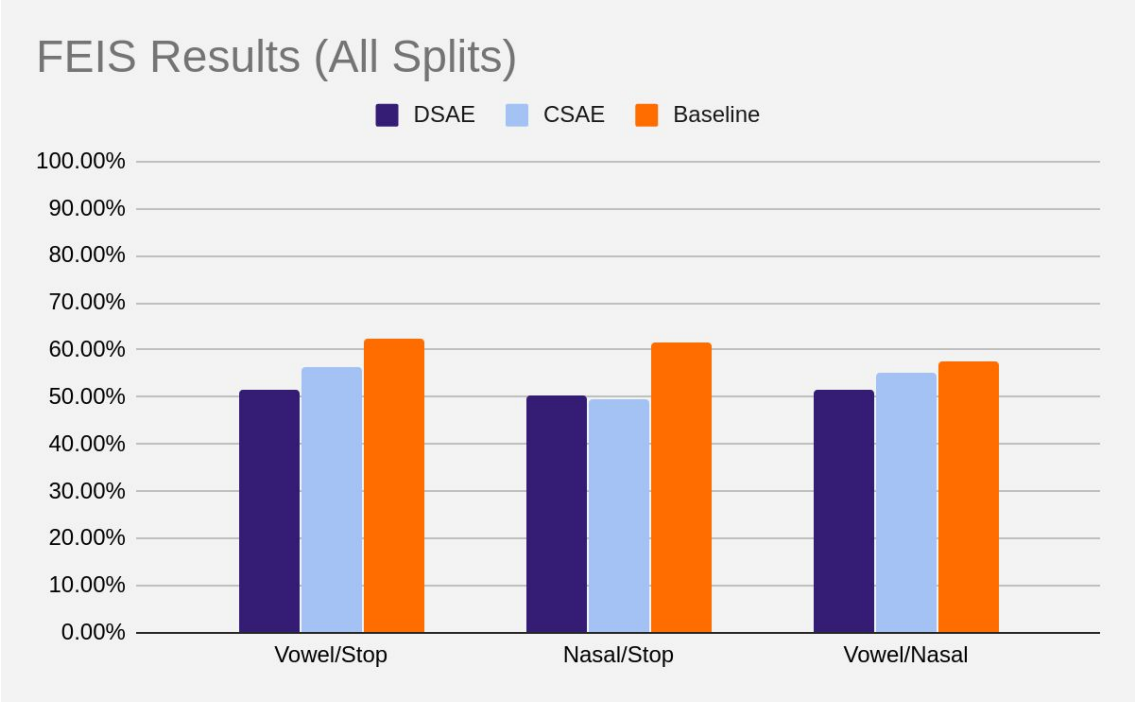

Figure 11. *Average accuracy results across 7 FEIS participants for all classification tasks. Complete table of values in Appendix 10.1.*

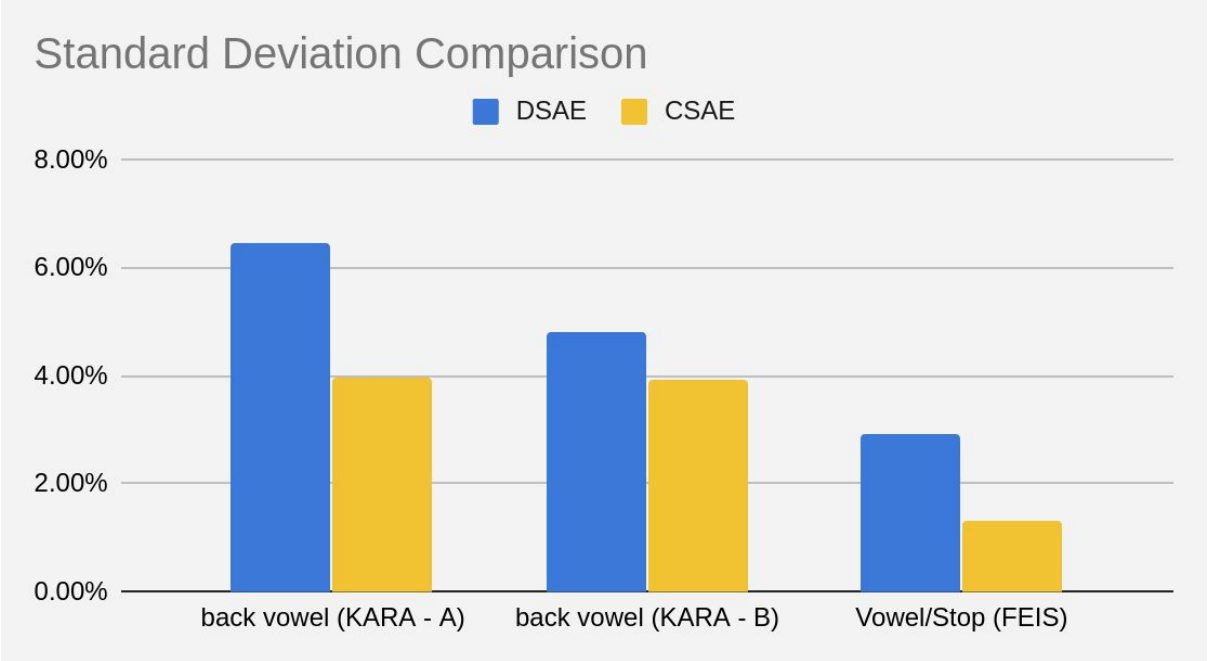

Figure 12. Standard deviation results (taken from our 5 seed values) averaged across all participants in a given dataset. We compare the DSAE and CSAE for the 3 classification tasks with the highest average accuracy.

7. Discussion

Autoencoder extracted features for most of the tested phonetic splits fail to reach an accuracy threshold of 70%, which precludes them from real world use (44). Further, autoencoder extracted features consistently perform worse during SVM classification than baseline hand selected features. That said, both autoencoder feature sets perform at above chance classification levels for certain splits of the data.

7.1 Comparison of CSAE and DSAE

Across experiments our CSAE tended to outperform our DSAE for experiments which achieved above chance accuracy. It is possible that our CSAE architecture is able to encode some sense of the timeseries quality of the data which is more easily lost in the fully connected architecture. This would be in line with what is suggested in (31) and (40). However, we also note that, in the case of better performing splits, the accuracies for the CSAE features tend to be less variable across seed values for the same experiment than those for the DSAE (see Figure 12). Perhaps the narrow convolutional kernel is an effective constraint in encouraging the DSAE to learn invariant representations of the data as suggested in (14). Alternatively, the larger standard deviation in accuracies for our DSAE may be the result of sensitivity to the model's weight initialization. We noted in our experiments that for some seed values (weight initializations) our DSAE suffered from so-called 'dying neurons'. In other words, neurons in ReLU layers during training only output 0 for every input. When this occurs, one or more of the features in our feature representation becomes 0 for all encoded data samples, which is akin to losing that feature, as it is no longer offering discriminative information at classification time.

In the future, researchers might consider using an activation function which does not have a threshold at 0 (as with ReLU), making it possible for these neurons to recover. Some examples include leaky ReLU, SELU, and ELU. Another option for handling this issue would be to adjust the weight initialization. For example, recent research in (45) has proposed

a new ‘randomized asymmetric’ initialization method designed to prevent dying neurons in ReLU layers.

7.2 Channel Selection

From our preliminary investigation into channel selection using the KARA One dataset, we note that the set of channels which results in higher accuracies is participant dependent. Inter-subject variability can explain why set A and set B perform differently. The method for selecting channels was based on a correlation using all of the available data. Due to differences between participants' physiology and recording environments, the channels with highest correlation for the entire dataset may not be the best selection of channels for a given participant.

Nonetheless, the effect of channel placement is clearly an important factor in the feature learning process. In the future, it may be prudent to select channels on a speaker dependent basis. We can imagine future data collection methods which involve using a higher density EEG device to select for best channel placement, prior to recording with individualized, lower density devices.

While high density EEG readings seem intuitively more fruitful for machine learning, research suggests that placement and duration of samples is more relevant than total number of channels. Research in (46) found evidence that recording duration was more important than spatial sampling size for epilepsy diagnostic yield of EEG data. Further, (1) provides evidence that the rate of improvement for their deep learning models slows as electrode numbers increase. They suggest that only limited information is gained by adding electrodes due to limited diversity in neural response. They also note that as the number of electrodes increases the number of free parameters in the neural network increases. These additional features may not be useful given the limited duration of their recordings (i.e. limited training data).

7.3 Dataset Evaluation

In the case of the KARA One dataset, we note that the split which tests for ‘presence of a back vowel’ performs at levels akin to or better than the baseline. These splits are significantly more accurate than all other splits tested for either dataset.

We suggest this is actually an issue of data collection methods used for the KARA One dataset. As noted in section 5.1.1, the KARA One researchers chose to have their participants ‘prepare their articulators’ prior to the imagined speech trial. This alignment of the articulators (tongue, jaw, etc.) may pollute the following imagined speech trial with articulator movement artifacts as suggested in (5) by the FEIS researchers. These artifacts may be stronger in the case of a back vowel and their presence in the EEG data allows for easier classification. If this is the case, then our feature extraction methods are using information from the sensorimotor cortex (47) to represent imagined speech, when this information should not be present in imagined speech data to begin with. We suggest that data which involves the movement or tensing of articulators, but not audible speech, should be classified as mimed speech, rather than thought speech.

These results highlight the importance of how data is gathered for imagined speech tasks. While it is appealing to think that neural networks can overcome noise in EEG signals, they still require data which is suited to the task. In this case, the FEIS dataset seems to make improvements on the KARA One dataset by replacing the ‘prepare articulators' phase with a fixation point. This is unfortunate since the KARA One dataset does offer more channels and thus the opportunity to explore channel placement.Yet the unsuitability of the data to the task is undoubtedly affecting our methods of channel selection as well.

Despite the noted improvement, the FEIS dataset comes with its own set of problems. First, the FEIS researchers instructed their participants to follow stop consonants with a neutral release (+ə). For our classification tasks, having this vowel in conjunction with each stop consonant could make it more difficult to learn features which discriminate between stop consonants and pure vowels. It is worth considering recording unreleased stops in future data.

Further, the FEIS researchers recorded their data using the Emotiv Epoc+ headset. This headset is appealing in its portability and relative cost. However, researchers testing the headset for use with P300 BCI found the headset to perform 'significantly worse' than a medical grade device while using the same set of electrodes (48), indicating that the device has design flaws which are unrelated to its low density, some of which we note in section (3.1).

7.4 Further Considerations for Future Research

7.4.1 Input Dimensions

In this research, we exclusively explore a convolutional architecture in which the signals from all 14 channels are flattened into a 1-dimensional representation. Future research might benefit from exploring a convolutional architecture which maintains a two-dimensional representation (time x channels). For example, researchers in (49) used a convolutional neural network which separated its convolutions into 2 parts (in different layers), a convolution over time and a convolution over channels.

Some research has taken this a step further, formatting EEG input akin to a video. In (50), researchers represent their EEG data as a sequence of multi-spectral images and use video classification techniques to learn representations of the data.

7.4.2 Sample Similarity

Recent research has also suggested making adjustments to the loss function of the autoencoder to help it better learn features for a given classification task. Researchers in (51) propose a relational autoencoder which incorporates the relationships between data samples by adjusting the loss function to minimize the distance between a pair of reconstructed samples. Further, researchers in (52) apply a similarity constraint to their autoencoder by enforcing that 2 instances from the same class be more similar to each other than to instances from another class. Ideally, this helps the autoencoder learn features which can distinguish between the classes.

8. Conclusion

Our experiments suggest that we can use autoencoders to extract features from low density EEG input. Further, we can use these features to achieve above chance accuracy in some phonetic classification tasks. However, we do not reach accuracies suitable for BCI application and note high variability in our results between both phonetic splits and participants. We cannot discount the value of expert specified features, which consistently perform better at classification time. We suggest hand made features may be more suitable to phonetic classification tasks when data is limited.

We still believe that autoencoder based feature learning is worth pursuing. Research has barely scratched the surface of using autoencoders to learn from EEG signals, particularly in the speech domain. However, the field is sorely lacking in useful datasets which are suitable for training neural models. In order to better explore autoencoder features, future data gathering processes will need to be scrutinized. The experimental setup for data gathering should be done with the task in mind, paying particular attention to the intended segmentation, feature extraction, and classification methods.

Lastly, our preliminary investigation of channel selection indicates that this would be a valuable avenue for future research in imagined speech, especially given a dataset more suited to imagined speech tasks. We suggest that further data gathering seeks to increase the duration of participant dependent data, rather than gathering shorter readings from more participants.

9. Works Cited

1. Akbari H, Khalighinejad B, Herrero JL, Mehta AD, Mesgarani N. Towards reconstructing intelligible speech from the human auditory cortex. Sci Rep. 2019 Jan 29;9(1):874.

2. Angrick M, Herff C, Mugler E, Tate MC, Slutzky MW, Krusienski DJ, et al. Speech synthesis from ECoG using densely connected 3D convolutional neural networks. J Neural Eng. 2019 Jun;16(3):036019.

3. Anumanchipalli GK, Chartier J, Chang EF. Speech synthesis from neural decoding of spoken sentences. Nature. 2019 Apr;568(7753):493–8.

4. Zhao S, Rudzicz F. Classifying phonological categories in imagined and articulated speech [Internet]. 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). 2015. Available from: http://dx.doi.org/10.1109/icassp.2015.7178118

5. Jonathan Clayton, Scott Wellington, Cassia Valentini-Botinhao, Oliver Watts. Decoding imagined, heard, and spoken speech: classification and regression of EEG using a 14-channel dry-contact mobile headset. In The University of Edinburgh, SpeakUnique Limited; [forthcoming].

6. He B, Sohrabpour A, Brown E, Liu Z. Electrophysiological Source Imaging: A Noninvasive Window to Brain Dynamics. Annu Rev Biomed Eng. 2018 Jun 4;20:171–96.

7. Roy Y, Banville H, Albuquerque I, Gramfort A, Falk TH, Faubert J. Deep learning-based electroencephalography analysis: a systematic review. J Neural Eng. 2019 Aug 14;16(5):051001.

8. Kim SG, Richter W, Uğurbil K. Limitations of temporal resolution in functional MRI. Magn Reson Med. 1997 Apr;37(4):631–6.

9. Lotte F, Bougrain L, Cichocki A, Clerc M, Congedo M, Rakotomamonjy A, et al. A review of classification algorithms for EEG-based brain-computer interfaces: a 10 year update. J Neural Eng. 2018 Jun;15(3):031005.

10. Birbaumer N, Cohen LG. Brain-computer interfaces: communication and restoration of movement in paralysis. J Physiol. 2007 Mar 15;579(Pt 3):621–36.

11. Dichter BK, Breshears JD, Leonard MK, Chang EF. The Control of Vocal Pitch in Human Laryngeal Motor Cortex. Cell. 2018 Jun 28;174(1):21–31.e9.

12. Chiong W, Leonard MK, Chang EF. Neurosurgical Patients as Human Research Subjects: Ethical Considerations in Intracranial Electrophysiology Research. Neurosurgery. 2018 Jul 1;83(1):29–37.

13. Engel AK, Moll CKE, Fried I, Ojemann GA. Invasive recordings from the human brain:

clinical insights and beyond. Nat Rev Neurosci. 2005 Jan;6(1):35–47.

14. Schirrmeister R, Springenberg J, Fiederer L, Glasstetter M, Eggensperger K, Tangermann M, et al. Deep learning with convolutional neural networks for brain mapping and decoding of movement-related information from the human EEG. 2017 Mar;

15. Leske S, Dalal SS. Reducing power line noise in EEG and MEG data via spectrum interpolation. Neuroimage. 2019 Apr 1;189:763–76.

16. Britton JW, Frey LC, Hopp JL, Korb P, Koubeissi MZ, Lievens WE, et al. Electroencephalography (EEG): An Introductory Text and Atlas of Normal and Abnormal Findings in Adults, Children, and Infants. St. Louis EK, Frey LC, editors. Chicago: American Epilepsy Society; 2016.

17. Hagemann D, Naumann E. The effects of ocular artifacts on (lateralized) broadband power in the EEG. Clin Neurophysiol. 2001 Feb;112(2):215–31.

18. Hansen ST, Hemakom A, Gylling Safeldt M, Krohne LK, Madsen KH, Siebner HR, et al. Unmixing Oscillatory Brain Activity by EEG Source Localization and Empirical Mode Decomposition. Comput Intell Neurosci. 2019 Mar 14;2019:5618303.

19. Biscay RJ, Bosch-Bayard JF, Pascual-Marqui RD. Unmixing EEG Inverse Solutions Based on Brain Segmentation. Front Neurosci. 2018 May 15;12:325.

20. Pujol J, Deus J, Losilla JM, Capdevila A. Cerebral lateralization of language in normal left-handed people studied by functional MRI. Neurology. 1999 Mar 23;52(5):1038–43.

21. Reiterer S, Pereda E, Bhattacharya J. Measuring second language proficiency with EEG synchronization: how functional cortical networks and hemispheric involvement differ as a function of proficiency level in second language speakers [Internet]. Vol. 25, Second Language Research. 2009. p. 77–106. Available from: http://dx.doi.org/10.1177/0267658308098997

22. Clayton J. Towards phone classification from imagined speech using a lightweight EEG brain-computer interface [Speech and Language Processing]. University of Edinburgh; 2019.

23. Gramfort A, Luessi M, Larson E, Engemann DA, Strohmeier D, Brodbeck C, et al. MNE software for processing MEG and EEG data. Neuroimage. 2014 Feb 1;86:446–60.

24. van Driel J, Olivers CNL, Fahrenfort JJ. High-pass filtering artifacts in multivariate classification of neural time series data [Internet]. Available from: http://dx.doi.org/10.1101/530220

25. Sreeja SR, Sahay RR, Samanta D, Mitra P. Removal of Eye Blink Artifacts From EEG Signals Using Sparsity. IEEE J Biomed Health Inform. 2018 Sep;22(5):1362–72.

26. Azami H, Anisheh SM, Hassanpour H. An Adaptive Automatic EEG Signal Segmentation Method Based on Generalized Likelihood Ratio [Internet]. Artificial Intelligence and Signal Processing. 2014. p. 172–80. Available from:

http://dx.doi.org/10.1007/978-3-319-10849-0_18

27. Mouraux A, Iannetti GD. Across-trial averaging of event-related EEG responses and beyond. Magn Reson Imaging. 2008 Sep;26(7):1041–54.

28. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, et al. Scikit-learn: Machine Learning in Python. J Mach Learn Res. 2011;12(85):2825–30.

29. McNair N. Entropy python package [Internet]. raphaelvallat.com. Available from: https://pypi.org/ project/EntroPy-Package/.

30. Goodfellow I, Bengio Y, Courville A. Deep Learning. MIT Press; 2016.

31. Wen T, Zhang Z. Deep Convolution Neural Network and Autoencoders-Based Unsupervised Feature Learning of EEG Signals [Internet]. Vol. 6, IEEE Access. 2018. p. 25399–410. Available from: http://dx.doi.org/10.1109/access.2018.2833746

32. Li X, Zhao Z, Song D, Zhang Y, Pan J, Wu L, et al. Latent Factor Decoding of Multi-Channel EEG for Emotion Recognition Through Autoencoder-Like Neural Networks. Front Neurosci. 2020 Mar 2;14:87.

33. Plaut E. From Principal Subspaces to Principal Components with Linear Autoencoders. ArXiv. 2018;abs/1804.10253.

34. Masci J, Meier U, Cireşan D, Schmidhuber J. Stacked Convolutional Auto-Encoders for Hierarchical Feature Extraction [Internet]. Lecture Notes in Computer Science. 2011. p. 52–9. Available from: http://dx.doi.org/10.1007/978-3-642-21735-7_7

35. Geng J, Fan J, Wang H, Ma X, Li B, Chen F. High-Resolution SAR Image Classification via Deep Convolutional Autoencoders [Internet]. Vol. 12, IEEE Geoscience and Remote Sensing Letters. 2015. p. 2351–5. Available from: http://dx.doi.org/10.1109/lgrs.2015.2478256

36. Ramos MP, Ramos VP, Fabian AL, Mamani EO. A Feature Extraction Method Based on Convolutional Autoencoder for Plant Leaves Classification [Internet]. 2019 IEEE Colombian Conference on Applications in Computational Intelligence (ColCACI). 2019. Available from: http://dx.doi.org/10.1109/colcaci.2019.8781985

37. Polic M, Krajacic I, Lepora N, Orsag M. Convolutional Autoencoder for Feature Extraction in Tactile Sensing [Internet]. Vol. 4, IEEE Robotics and Automation Letters. 2019. p. 3671–8. Available from: http://dx.doi.org/10.1109/lra.2019.2927950

38. Elhami G, Weber RM. Audio Feature Extraction with Convolutional Neural Autoencoders with Application to Voice Conversion [Internet]. 2019 May [cited 2020 Aug 12]. Report No.: CONF. Available from: https://infoscience.epfl.ch/record/261268?ln=en

39. Grais EM, Plumbley MD. Single channel audio source separation using convolutional denoising autoencoders [Internet]. 2017 IEEE Global Conference on Signal and Information Processing (GlobalSIP). 2017. Available from:

http://dx.doi.org/10.1109/globalsip.2017.8309164

40. Bai S, Kolter J, Koltun V. An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling. 2018 Mar;

41. LeNail A. NN-SVG: Publication-Ready Neural Network Architecture Schematics. Journal of Open Source Software. 2019 Jan 15;4(33):747.

42. Glorot X, Bengio Y. Understanding the difficulty of training deep feedforward neural networks. In: AISTATS. 2010.

43. He K, Zhang X, Ren S, Sun J. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification [Internet]. 2015 IEEE International Conference on Computer Vision (ICCV). 2015. Available from: http://dx.doi.org/10.1109/iccv.2015.123

44. Kübler A, Kotchoubey B, Kaiser J, Wolpaw JR, Birbaumer N. Brain-computer communication: unlocking the locked in. Psychol Bull. 2001 May;127(3):358–75.

45. Lu L, Shin Y, Su Y, Karniadakis GE. Dying ReLU and Initialization: Theory and Numerical Examples [Internet]. arXiv [stat.ML]. 2019. Available from: http://arxiv.org/abs/1903.06733

46. Bach Justesen A, Foged MT, Fabricius M, Skaarup C, Hamrouni N, Martens T, et al. Diagnostic yield of high-density versus low-density EEG: The effect of spatial sampling, timing and duration of recording. Clin Neurophysiol. 2019 Nov;130(11):2060–4.

47. Pulvermüller F, Huss M, Kherif F, Moscoso del Prado Martin F, Hauk O, Shtyrov Y. Motor cortex maps articulatory features of speech sounds. Proc Natl Acad Sci U S A. 2006 May 16;103(20):7865–70.

48. Duvinage M, Castermans T, Petieau M, Hoellinger T, Cheron G, Dutoit T. Performance of the Emotiv Epoc headset for P300-based applications. Biomed Eng Online. 2013 Jun 25;12:56.

49. Schirrmeister RT, Springenberg JT, Fiederer LDJ, Glasstetter M, Eggensperger K, Tangermann M, et al. Deep learning with convolutional neural networks for EEG decoding and visualization. Hum Brain Mapp. 2017 Nov;38(11):5391–420.

50. Bashivan P, Rish I, Yeasin M, Codella N. Learning Representations from EEG with Deep Recurrent-Convolutional Neural Networks [Internet]. arXiv [cs.LG]. 2015. Available from: http://arxiv.org/abs/1511.06448

51. Meng Q, Catchpoole D, Skillicom D, Kennedy PJ. Relational autoencoder for feature extraction [Internet]. 2017 International Joint Conference on Neural Networks (IJCNN). 2017. Available from: http://dx.doi.org/10.1109/ijcnn.2017.7965877

52. Stober S, Sternin A, Owen AM, Grahn JA. Deep Feature Learning for EEG Recordings [Internet]. arXiv [cs.NE]. 2015. Available from: http://arxiv.org/abs/1511.04306

10. Appendices

10.1 Results Tables

FEIS (vowel / stop)

Feature Set	1	2	3	4	5	6	7
DSAE	0.526	0.494	0.518	0.5453333333	0.502	0.5086666667	0.52
CSAE	0.578	0.578	0.5826666667	0.5753333333	0.55	0.5746666667	0.52
Baseline	0.634313725	0.512745098	0.6637254902	0.6323529412	0.6264705882	0.681372549	0.6117647059

FEIS (nasal / stop)

Feature Set	1	2	3	4	5	6	7
DSAE	0.4853333333	0.5093333333	0.5006666667	0.5146666667	0.5113333333	0.4893333333	0.5
CSAE	0.522	0.4906666667	0.4853333333	0.4613333333	0.486	0.5066666667	0.5033333333
Baseline	0.638235294	0.5166666667	0.6784313725	0.5519607843	0.6245098039	0.6490196078	0.6529411765

FEIS (vowel / nasal)

Feature Set	1	2	3	4	5	6	7
DSAE	0.53	0.5146666667	0.5106666667	0.5153333333	0.512	0.5046666667	0.5306666667
CSAE	0.554	0.5693333333	0.536	0.558	0.5546666667	0.56	0.518
Baseline	0.660784313	0.5068627451	0.5421568627	0.5666666667	0.5911764706	0.6117647059	0.5431372549

Table 5. Accuracy scores for each participant in the FEIS dataset for each experiment.

Table 6. Accuracy scores for each Participant in the KARA One dataset for each experiment.

KARA One (+ back vowel)

Feature Set	Channels	mm05	mm08	mm09	mm10	mm12	mm15	mm16	mm19	mm20
DSAE	Set A	0.804	0.68	0.845	0.6916666667	0.9105555556	0.7661111111	0.6666666667	0.6733333333	0.685
CSAE	Set A	0.8377777778	0.8294444444	0.8977777778	0.715	0.9227777778	0.8505555556	0.715	0.8022222222	0.7655555556
Baseline	Set A	0.8830065359	0.5819739043	0.9075075276	0.5950886584	0.9575309468	0.9166677819	0.5885312814	0.5950886584	0.5754165273
DSAE	Set B	0.6911111111	0.7166666667	0.8438888889	0.7061111111	0.9144444444	0.8333333333	0.6144444444	0.6055555556	0.6638888889
CSAE	Set B	0.7271111111	0.8522222222	0.8633333333	0.7761111111	0.9311111111	0.8361111111	0.6055555556	0.7405555556	0.8033333333
Baseline	Set B	1	0.5893509535	0.592629642	0.5950886584	0.586072265	0.5819739043	0.5811542322	0.926390097	0.9714018066

KARA One (+ nasal)

Feature Set	Channels	mm05	mm08	mm09	mm10	mm12	mm15	mm16	mm19	mm20
DSAE	Set A	0.5746666667	0.4375	0.5691666667	0.5091666667	0.505	0.5275	0.4791666667	0.5216666667	0.5566666667
CSAE	Set A	0.5573333333	0.4766666667	0.5958333333	0.4741666667	0.6016666667	0.5791666667	0.51	0.545	0.6225
Baseline	Set A	0.6833333333	0.5475460123	0.6544665569	0.617589406	0.6814753853	0.6557010325	0.5377899147	0.5451069879	0.7097935059
DSAE	Set B	0.5733333333	0.47	0.5025	0.5183333333	0.6158333333	0.5758333333	0.4933333333	0.5308333333	0.525
CSAE	Set B	0.5366666667	0.5475	0.5516666667	0.515	0.6066666667	0.6533333333	0.505	0.555	0.6216666667
Baseline	Set B	0.6852941176	0.5304728415	0.7010474338	0.5231557684	0.6913137812	0.5414484513	0.7255050127	0.5414484513	0.752401616

KARA One (+ bilabial)

Feature Set	Channels	mm05	mm08	mm09	mm10	mm12	mm15	mm16	mm19	mm20
DSAE	Set A	0.4813333333	0.4772222222	0.55	0.4766666667	0.5088888889	0.4783333333	0.5383333333	0.4744444444	0.5038888889
CSAE	Set A	0.4591111111	0.5005555556	0.5605555556	0.455	0.5427777778	0.485	0.5327777778	0.4811111111	0.4577777778
Baseline	Set A	0.6516339869	0.49836734693	0.6274573436	0.6617530947	0.6674573436	0.6356206089	0.6405754433	0.6494981599	0.636440281
DSAE	Set B	0.5182222222	0.4738888889	0.4916666667	0.4844444444	0.5766666667	0.5433333333	0.5116666667	0.4894444444	0.5177777778
CSAE	Set B	0.5111111111	0.5338888889	0.53	0.4611111111	0.5455555556	0.435	0.5177777778	0.5055555556	0.5205555556
Baseline	Set B	0.6483660131	0.5196788223	0.6552659752	0.6274573436	0.6527567748	0.5229575109	0.643777183	0.6503211777	0.6699130144

10.2 Hand Selected Features (Used in Baseline)

1. Mean average
2. Mean of absolute values
3. Maximum value
4. Max of absolute values
5. Minimum value
6. Min of absolute values
7. Approximate entropy
8. Maximum value + minimum value
9. Maximum value - minimum value
10. Curve length
11. Detrended fluctuation analysis
12. Energy (sum of squared amplitudes)
13. Higuchi Fractal Dimensions
14. Simpson Integral
15. Katz Fractal Dimensions
16. Kurtosis
17. Nonlinear energy
18. Permutation Entropy
19. Petrosian Fractal Dimensions
20. Root mean square
21. Skew
22. Sample entropy
23. Spectral entropy
24. Standard deviation
25. Spectral value decomposition entropy
26. Sum of all amplitudes
27. Variance

Table 7. This list has been adapted from appendix 7.1 in (22)