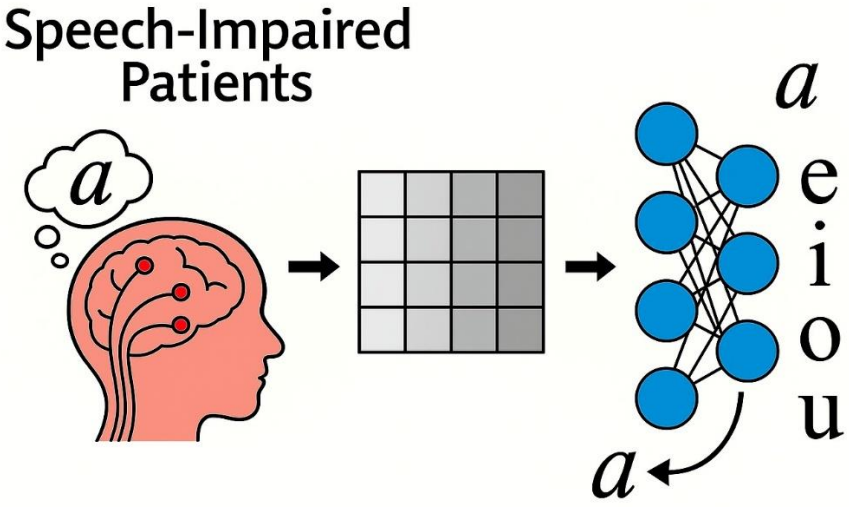
Final Project Report – Decoding of Speech Features from Single Neuron Recordings from the Human Brain

By: Eadan Schechter,
Idan Kanat
Supervisors: Dr. Ariel Tankus
Functional Neurosurgery Unit
Tel Aviv Sourasky Medical Center ("Ichilov"),
Department of Neurology and Neurosurgery, School of Medicine, Tel Aviv University, Tel Aviv, Israel,
Sagol School of Neuroscience, Tel Aviv University, Tel Aviv, Israel
Prof. Neta Rabin
School of Industrial & Intelligent Systems Engineering, Tel Aviv University, Israel

October 2024 - July 2025



Background and Objective

Abstract

Individuals with neurological disorders, e.g. ALS, brain stem stroke or brain injury, may experience significant impairments in their ability to speak, leaving them unable to communicate even their most basic needs. In this project, we aimed to tackle this important problem by developing a model that decodes speech directly from electrical brain signals

Our project aimed to develop and compare Deep Learning (DL) models for offline decoding of vowel articulations directly from the electrical activity of single neurons in the brain of epilepsy patients.

Experimental results show, on the one hand, that there is no universal "best" model, out of the models that we tested, across all individuals; on the other hand, they demonstrate the technical feasibility of decoding speech elements from single neuron recordings.

This project was conducted as a collaboration between the School of Industrial & Intelligent Systems Engineering at Tel Aviv University and Dr. Tankus from Sourasky Medical Center (Ichilov). The work is built upon earlier research by Dr. Tankus, which demonstrated the classification of two vowel sounds from the neural recordings of a single patient. Our project took this further, extending it to more phonemes and more patients. Ultimately, this project could pave the way for brain-computer interfaces (BCIs) that will restore speech communication in completely paralyzed patients.

Scientific Background

Speech impairments caused by neurological disorders - including amyotrophic lateral sclerosis (ALS), epilepsy, and Parkinson's disease - can lead to devastating loss of communication[1,2]. Brain–computer interfaces (BCIs) aim to restore this ability by decoding neural signals directly into text or synthesized speech. Early studies demonstrated that single-neuron recordings from speech motor cortex could provide enough information to synthesize vowel sounds or classify intended phonemes in patients with paralysis[3,4].

Advances in high-density intracranial recordings, such as electrocorticography (ECoG) and microelectrode arrays, have enabled researchers to decode not just isolated phonemes but entire words and sentences by extracting features like high-gamma power that reflect ensemble neuronal activity[5-8]. Recent clinical breakthroughs have shown that speech BCIs can restore practical communication abilities in patients with severe paralysis or anarthria. For instance, Moses et al. (2021)[2] and Willett et al. (2023)[9] demonstrated that paralyzed patients could achieve real-time decoding of intended speech at word error rates approaching those required for conversation.

The rapid progress in this field is largely due to the integration of deep learning methods. Convolutional neural networks (CNNs) and recurrent neural networks (RNNs), including

architectures like GRU[16] and LSTM, are now commonly used for end-to-end decoding of neural time series[10-12]. Compact models such as EEGNet have proven that interpretable and parameter-efficient architectures can generalize across diverse BCI tasks[10]. Additionally, recent studies have shown that speech-relevant neural activity can be decoded not only from classical speech motor areas but also from non-traditional regions such as the hippocampus and orbitofrontal cortex, expanding the potential of BCIs[1].

Clinically, epilepsy patients undergoing intracranial monitoring provide a unique and invaluable setup for speech BCI research, as the clinical necessity of electrode implantation enables direct, high-resolution study of human brain networks underlying speech in awake, communicating individuals[1,5]. While our current study utilizes data exclusively from epilepsy patients, the ultimate goal of this line of research is to develop neuroprosthetic solutions for restoring communication in individuals with severe speech impairments, such as those with ALS, Parkinson's disease, and other related conditions in which speech is profoundly affected[2,13].

Building on this background, our project evaluates and compares deep learning models (CNN, GRU, EEGNet) for vowel classification from multi-channel single-neuron recordings in epilepsy patients.

Scientific Problem

Individuals with severe neurological conditions often experience profound speech impairments that prevent them from communicating even simple needs. Brain–computer interfaces (BCIs) offer hope for restoring communication by decoding speech-related electrical signals directly from the brain. However, a significant technical challenge remains - accurately translating single-neuron recordings into meaningful speech elements in real time, across various individuals with diverse neurological conditions and levels of impairment.

Project's Goal and Long-Term Goal

In this project, our goal is to develop and compare between deep learning models for offline decoding of vowel articulations directly from the electrical activity of single neurons in the brain of epilepsy patients. This project paves the way for brain-computer interfaces (BCIs) that will restore speech communication in completely paralyzed patients.

The long-term goal of this research is to develop a model that can decode speech from single-neuron activity in any individual with complete paralysis, including those with total loss of speech - thereby paving the way for clinically viable BCIs that restore naturalistic communication in this population.

Methods

The Dataset

Overview

The data was recorded from 7 neurosurgical epilepsy patients, who agreed to participate in the speech decoding research, which is not related to epilepsy. Each patient underwent neurosurgical procedures involving the implantation of invasive electrodes in different regions of their brain according to the patient's clinical need. These electrodes, equipped with microwires, recorded electrical brain activity at the single-neuron level. Each such microwire is referred to as a channel. Both the locations of these electrodes and the number of channels varied across patients. It is important to note that not all channels were associated with speech production.

Following implantation, each patient was instructed to articulate one of the phonemes: a /ä/, e /ę/, i /i/, o /o̜/, u /u/ (for simplicity, we used the English rather than IPA transcription throughout the report) after hearing a *beep* sound. Each articulation constitutes a trial, with each trial labeled according to the vowel spoken. Neuronal activity was recorded for one second before and one second after *beep* onset in each trial, to capture the full temporal activity of speech articulation. The number of trials per patient varied between 200 to 370, leading to differing amounts of data per patient. Additionally, all but one patient had a uniform distribution of vowel articulations, this highlights the variability in both trial counts and distribution across patients.

Data Cleaning and Preparation

Each patient's dataset included two components:

- A labels file: Labelled each trial of a patient with the corresponding vowel - *a*, *e*, *i*, *o*, *u* respectively.
- A neural data file: Contains the neuronal activity recorded during all trials of a patient. The raw signal was recorded at 30 kHz. This enabled the detection of brief bursts of electrical activity. From these signals, action potentials ("spikes") were extracted.

  These spikes are counted within fixed, equally sized time bins of 100 milliseconds, forming what is referred to as the firing rate - the number of spikes detected in each time bin. Since all bins are of equal duration, the firing rate directly corresponds to the spike count per bin and can also be measured in spikes per second (Hz) for standardization. For example, if a neuron fired 3 spikes in a 100-millisecond time bin, the firing rate would be 30 Hz, since 3 spikes per 0.1 seconds equals on average to 30 spikes per second.

  Additionally, each trial spans 20 equally sized time bins of 100 milliseconds - 10 bins before and 10 after the *beep* sound that signaled for the patient to articulate a vowel. Trial's neural data is represented as a matrix. Each entry in the matrix thus reflects the firing rate of a specific channel, during a specific time bin, in that specific trial. The dimensions of each

matrix are $20 \times C$, where 20 corresponds to the 20 time bins mentioned before, and C is the number of channels per patient.

This process constituted the feature extraction stage for our model. Instead of using the raw electrical recordings directly, we represented each trial as a $20 \times C$ matrix of binned firing rates, making it suitable for downstream modeling. Figure 1 illustrates the variability in neuronal data across patients: Each of the 2 example matrices represents the data of a generic single trial from different patients. In both matrices, the rows correspond to time bins, which are fixed at 20 and remain consistent across all patients and trials. However, the number of columns differ: for each patient, they represent neuronal activity recorded from a unique set of brain locations, determined by the electrodes implanted in that individual. The number of channels - and thus the number of columns - varies from patient to patient.

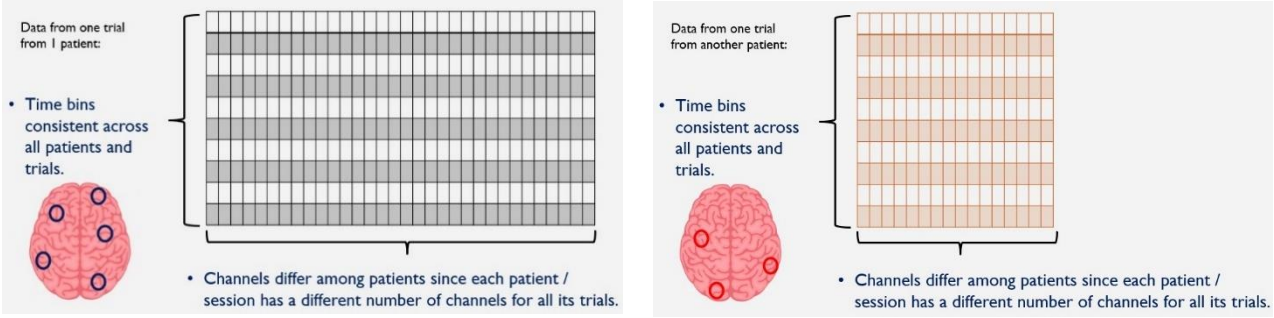


Figure 1: Variability in neuronal data across two different patients. To the left of each image - a schematic brain diagram. Red or blue circles mark distinct electrode placements in the two patients. To the right of each image, examples of data matrices - one in orange for a patient with fewer electrodes, and one in gray for a patient with more electrodes - highlighting how the matrix width reflects the number of recorded neurons in each trial for each patient.

Exploratory Data Analysis (EDA)

To better understand the structure and variability of the neuronal data, we conducted a series of exploratory analyses across patients, channels, vowels, and time. Below is a summary of the key visualizations and the insights they revealed. These analyses were performed for all seven patients, but we decided to focus here on two representative examples, Patient 3 and Patient 5, for illustrating the core trends and challenges.

Figure 2 plots the average firing rate per time bin and channel per patient (Heatmap):
For each patient, we visualized the average firing rate across all trials using a heatmap indexed by time bins (x-axis) and channels (y-axis) , with the color scale indicating firing intensity. Patient 3 exhibited widespread and consistently high activity across many channels, whereas Patient 5 showed sparser activation, with only a few channels contributing strong neuronal signals.

By design, the 10th time bin corresponds to the *beep* sound (time 0 to time 100 ms) that prompted the patient to begin articulating a vowel. The blue rectangle in each heatmap highlights the pre-cue (pre-beep) window - time bins preceding the beep - which serves as a baseline reference for neuronal activity. The green rectangle marks the post-cue (post-beep) window, which immediately follows the auditory cue. The most notable feature in Patient 3's heatmap is the pronounced increase in firing rate, particularly in the most active channels, within the post-cue (green) window. This surge in activity is time-locked to the auditory cue and likely reflects speech related processes. In contrast, the pre-cue (blue) window demonstrates relatively lower activity, reinforcing its role as a baseline for comparison. For Patient 5, increases in firing rate post-cue are less pronounced and limited to a smaller subset of channels, indicating that fewer channels contributed strong speech-related signals during task performance.

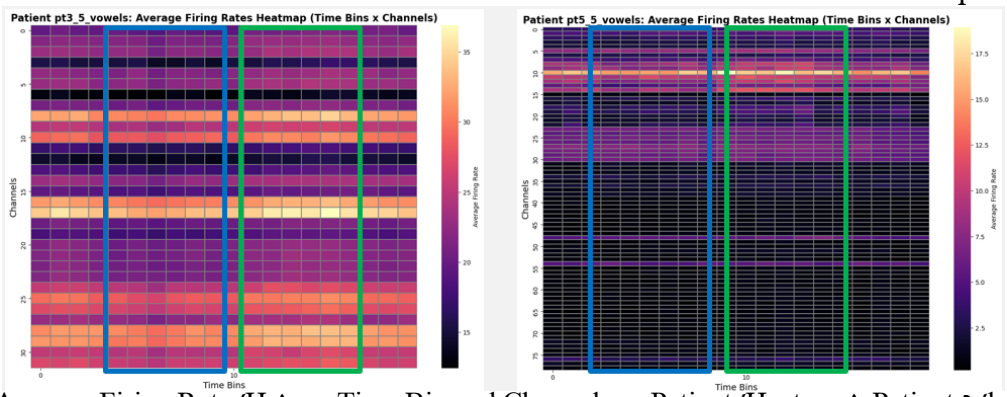


Figure 2: Average Firing Rate (Hz) per Time Bin and Channel per Patient (Heatmap). Patient 3 (left), Patient 5 (right).

Figure 3 plots the average firing rate per patient per channel (±standard error): We computed and visualized the average firing rate for each channel, aggregated across all trials and time bins, along with the corresponding standard error. The firing rate distributions differed substantially between patients. It shows that patient 3 exhibited consistently high firing rates across most channels, with peak values around 30 spikes/sec. The firing rate standard errors were small and stable across channels - indicating broadly consistent and reliable activity. In contrast, patient 5 showed much lower overall activity, with peak values closer to 16 Hz. Additionally, he showed uneven activation: a few channels had notably higher firing rates than the rest, while others remained mostly inactive. This pattern indicates that in patient 5, only a subset of channels show higher overall firing rates, which may reflect differences in electrode placement or engagement of speech-related brain areas. However, it is important to note that a neuron's contribution to speech decoding cannot be determined by firing rate alone. A neuron may have a low firing rate and be fully engaged in a specific task.

Additionally, we observe for patient 5 that channels with higher average firing rates also tended to have higher standard errors, reflecting greater trial-to-trial variability. While this could indicate

inconsistent activation across trials, it may also result from channels that exhibit low baseline firing rates with significantly increased firing rates for short periods of time, during task-related responses, such channels may be particularly informative for speech decoding. Therefore, high variability should not be interpreted solely as a negative property but rather may indicate difference between baseline and response neuronal activity. Importantly, since the electrodes were implanted in different anatomical regions for each patient, channels are not spatially or functionally aligned across individuals - meaning that channel no. 10 in patient 3 may record from a completely different brain region than channel no. 10 in patient 5. This spatial heterogeneity underscores the difficulty of aligning features across patients and highlights the need for individualized modeling or architecture designs that are robust to variable input configurations.

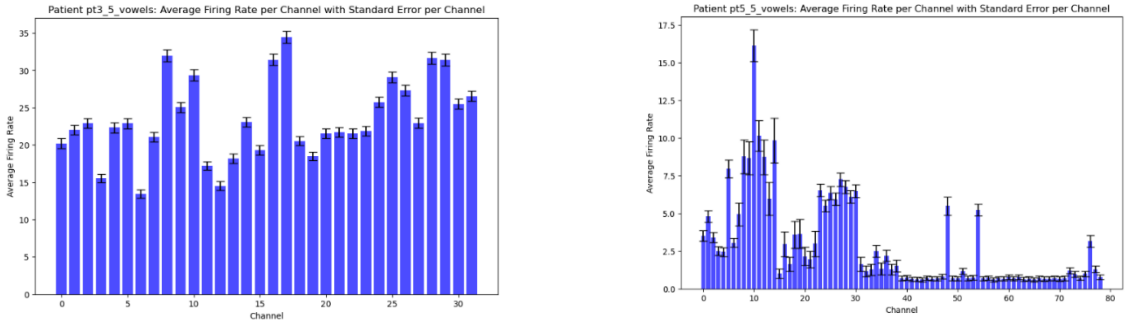


Figure 3: Average Firing Rate (Hz) per Patient per Channel (±Standard Error). Patient 3 (left), Patient 5 (right).

Figure 4 plots the average firing rate per patient per channel grouped by vowel: By grouping trials per vowel, we examined whether certain channels were particularly sensitive to specific vowels. For both patients, the overall firing rate distributions remained mostly consistent across vowels, with similar patterns of channel activation regardless of the vowel articulated. This indicates that, at the average firing rate level, most channels do not exhibit strong discrimination for individual vowels. As a result, decoding vowels in this dataset is likely to require leveraging information from more complex patterns, rather than relying solely on pronounced channel-specific selectivity.

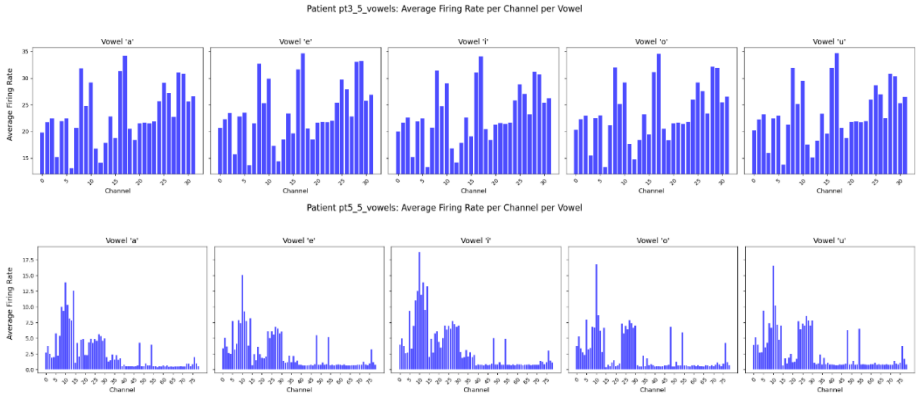


Figure 4: Average Firing Rate (Hz) per Patient per Channel Grouped by Vowel. Patient 3 (top), Patient 5 (bottom).

Figure 5 shows the control charts - average firing rate per patient per time bin (± standard deviation), grouped by vowel: To better understand variability over time, we plotted control charts showing the mean (blue line) and ±1 standard deviation (shaded) of firing rates over time, grouped by vowel. For patient 3, all vowel curves fluctuated around a similar mean value (~24 Hz), and the variability was consistent across vowels, with tight and stable standard deviation. This indicates reliable and uniform firing behavior across vowels. In contrast, patient 5 vowel curves are as well centered around a lower mean (~3 Hz), but the variability differs noticeably between vowels. Specifically, vowel i exhibited wider standard deviation area than others, such as o or u.

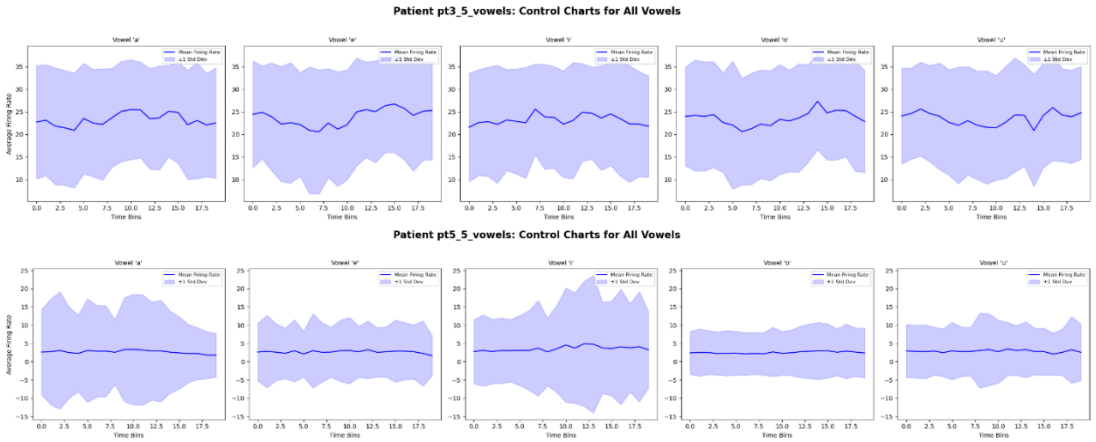


Figure 5: Control Charts showing the Average Firing Rate (Hz) per Patient per Time Bin (± Standard Deviation), Grouped by Vowel. Patient 3 (top), Patient 5 (bottom).

Dimensionality Reduction

To visually assess the feasibility of vowel classification and examine whether trials corresponding to different vowels exhibit distinguishable structure, we applied dimensionality reduction techniques to project the data into a shared feature space. This enabled us to evaluate the separability of vowel categories based on their neural representation.

We applied two dimensionality reduction techniques - Principal Component Analysis (PCA)[14] and t-Distributed Stochastic Neighbor Embedding (t-SNE)[15], into a shared three-dimensional space. PCA is a linear technique that projects data into directions that preserve the highest variance. In contrast, t-SNE is a non-linear method that aims to preserve local neighborhood structure. Results on patients 3 and 5 are shown in Figure 6 below.

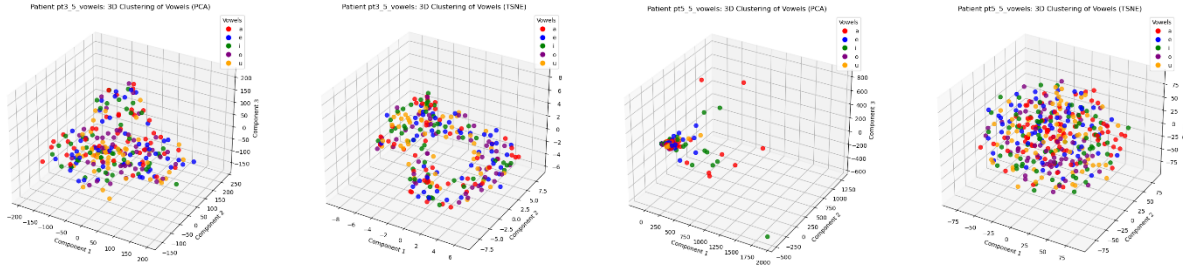


Figure 6: PCA and t-SNE Results (3D) for patients 3 and 5. Patient 3 (left), Patient 5 (right). For each patient the left image represents the PCA projection, and the right image represents the t-SNE embedding . Each color represents a vowel.

Figure 6 reveals that for patients 3 and 5, neither method revealed clear separation or clustering of trials by vowel label. This suggests that even in the reduced space, trials associated with different vowels remain entangled - with no obvious structure that distinguishes them from one another.
These results highlight the challenge of learning a shared representation across patients and vowels using simple projection techniques. They also raise important questions about whether more flexible modeling approaches will be able to uncover meaningful structure in the data - assuming such structure exists.

Algorithms and Methodology

Background and Rationale

To assess the feasibility of classifying speech phonemes from single-neuron recordings, we adopted a patient-specific modeling approach. This allowed us to evaluate the performance of several deep learning architectures on the five-vowel classification task, where the input dimensionality remained consistent within each patient.

Stepwise Methodological Approach

1. Initial 5-Class Evaluation

We began by training all models on the full five-vowel classification problem for each patient. This step was crucial for understanding the baseline capabilities of each model and identifying vowel classes for each patient where the models could extract any meaningful signal. A standard train/validation/test split was used for this stage. Initially, 90% of the data was used for training and 10% for testing. Then, 5-fold cross-validation was applied to the training set, resulting in an effective split of 72% training, 18% validation, and 10% test. Performance was assessed via confusion matrices with class-wise accuracies, and overall accuracy in comparison to the chance level.

2. Selection for Binary Classification

Given the limited success on the 5-class task, we refined our approach to focus on simpler, pairwise binary classification problems. The CNN models were used in order to select the binary pairs that were compared. This was done using the class accuracy achieved for each specific phenome, as seen in the confusion matrices applied at this stage. Among the pairs collected from the CNN models we used the GRU model as another comparable model for these pairs. This patient-specific and data-driven selection process ensured that further binary analyses targeted class pairs with evidence of discriminative neural encoding.

3. Model Architectures

Three deep learning models were developed and systematically evaluated: a Convolutional Neural Network (CNN), a Gated Recurrent Unit (GRU) network, and an EEGNet-inspired CNN architecture. Each model was adapted to handle the high-dimensional, temporally binned neural data, and the number of input channels, which varied by patient. Moreover, all models were implemented manually via Pytorch.

Full details of each architecture, including design rationale and implementation, are described in the subsections below.

4. Training and Evaluation Protocol

For 5 class classification, models were trained using a standard split (as explained in step 1). For binary classification, we employed a rigorous Double Cross-Validation (nested cross-validation) framework to obtain robust and unbiased estimates of generalization performance and to support statistical inference, that is described below.

5. Statistical Analysis

We summarized test accuracy performance with descriptive statistics: mean, standard deviation, and 95% confidence intervals (based on the t-distribution). To assess whether model performance exceeded chance, we conducted a one-sided t-test with the null hypothesis that the mean accuracy equals random guessing (H_0: $\mu = 0.5$, H_1: $\mu > 0.5$ for all binary tasks) and reported its p-value. All training and validation metrics were tracked using the Weights & Biases platform to ensure transparency and reproducibility.

Model Architectures

Before delving into model design, it is important to highlight a fundamental challenge: the nature of our data. Unlike typical tabular datasets, our input consists of high-dimensional neuronal activity matrices, representing firing rates over time across a varying number of implanted channels for

each patient. This inherently non-tabular and variable-sized format required us to move beyond conventional methods and approaches and adopt custom neural architectures capable of handling such complexity.

To address this, we systematically evaluated three distinct deep learning models, each tailored to the structure of our neuronal recordings: a Convolutional Neural Network (CNN) [1], a Gated Recurrent Unit (GRU) based network[2], and an EEGNet-inspired CNN architecture[10]. All models were specifically designed to process the temporally binned, channel-varying firing rate matrices derived from individual patient recordings. We focused on determining whether these deep learning approaches could effectively distinguish between the vowel classes within each patient, leveraging the consistency of input dimensions within patient-specific datasets.

Below, we detail the rationale and design of each model architecture.

Before developing our own models, we first considered the foundational work of Dr.Tankus[1], who demonstrated that CNN models can decode speech-related information directly from single-neuron brain activity (appendix 1). This model served as both a conceptual and practical starting point for our own CNN implementation.

Convolutional Neural Network (CNN) Architecture

Given the channel-wise and temporal structure of our input - firing rate matrices that capture neuronal activation over time and across channels, CNNs are a natural choice. (Note: for simplicity, we refer to channel-wise dimension as spatial throughout the entire report). CNNs are especially effective for detecting local patterns in structured data. In our case, the CNN architecture remains relatively lightweight, which is particularly important given the limited size of our training data. Ultimately, these considerations made the CNN architecture a suitable and practical choice for our problem setting.

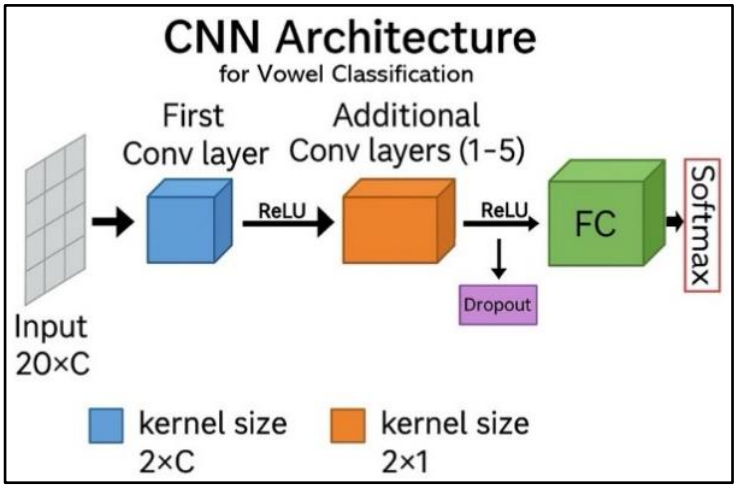


Figure 7 – CNN Architecture

As shown in Figure 7, the input of each trial was represented as a 20×C matrix, where the 20 rows correspond to temporally binned firing rates across 20 consecutive 100 ms windows (spanning a 2-second interval, 1 second before and after the *beep* onset), and each of the C columns corresponds to a recording channel.

Adapted from Tankus et al., (2024)[1], the model architecture begins with a 2D convolutional layer which serves as a feature extractor, capturing local temporal dynamics across the entire set of channels. This layer uses a kernel of size $(2 \times C)$ and outputs a number of filters, referred to as

hidden dimensionality which is a HP. This is followed by a ReLU activation function for non-linearity.

Following this, the network has an additional 1–5 convolutional layers, each employing a kernel of size (2 × 1) to extract deeper temporal features, after the initial channel-level aggregation that was performed earlier. Each of these layers was followed by a ReLU activation function for non-linearity. These deeper layers progressively reduce the temporal dimension, thereby examining a wider range of temporal dependencies. The number of layers is treated as a hyperparameter and optimized via Optuna. This layered configuration enables the model to capture increasingly complex features while preserving a relatively lightweight architecture - an essential consideration given the limited size of our dataset.

After the convolutional layers, a dropout layer is applied for regularization during training. The resulting feature maps are flattened and passed through a fully connected linear layer that outputs logits. These logits are converted to class probabilities using a SoftMax function. Softmax is not included in the model architecture, as our chosen loss function - Cross Entropy - applies it internally to the model's output scores (raw logits).

The overall architecture - including the number of convolutional layers, dropout rate, hidden dimensionality, and other parameters - is dynamically optimized through hyperparameter tuning.

EEGNet-Inspired CNN Architecture

The EEGNet-inspired architecture used in this study was adapted from Lawhern et al. (2018)[10], who designed a compact convolutional neural network specifically designed for EEG-based classification across a range of paradigms. Our PyTorch implementation preserves the core structure of EEGNet while adapting it to single neuron recordings with varying numbers of input channels per patient. Each input is represented as a 4D tensor of shape (batch size, 1, 20, C), where 20 denotes the number of 100 ms temporal bins and C represents the number of neural channels.

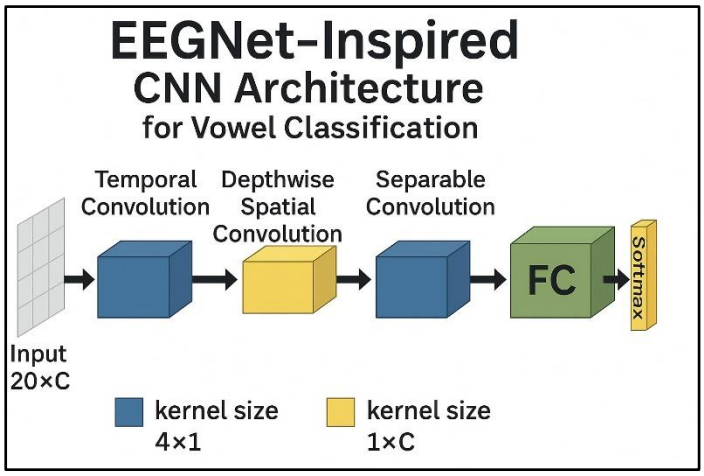


Figure 8 – EEGNet-Inspired CNN Architecture

As shown in Figure 8, the architecture begins with a temporal convolution using a kernel size of (4×1), which learns local temporal features without mixing across channels, producing F1 activation maps. This is followed by a depthwise spatial convolution with kernel size (1×C), designed to capture spatial (channel-wise) patterns independently for each temporal filter. The result is F1 × D activation maps, where each of the F1

temporal filters is paired with D depthwise spatial filters. Both F1 (number of temporal filters) and D (depth multiplier) are hyperparameters optimized via cross-validation via Optuna.

A final separable convolution layer with kernel size (4×1) further models local temporal dependencies and interactions between previously extracted features for more representational efficiency, producing F2 activation maps.

These convolutional blocks are followed by batch normalization, an ELU activation function for non-linearity, and average pooling, which reduces the temporal dimension while preserving relevant features. Regularization is incorporated through dropout layers during training, applied after both the depthwise and separable convolution stages to mitigate overfitting.

The resulting feature maps are flattened and passed through a fully connected linear classifier to output class logits. These logits are processed by a SoftMax layer during training via the cross-entropy loss function.

Overall, this architecture combines domain specific convolution operations – temporal filtering, and spatial separation – enabling it to effectively model both temporal and spatial patterns in neuronal data without relying on handcrafted features.

GRU Architecture

Unlike the convolutional models, which focused on extracting spatial-temporal features, this recurrent model leverages gated recurrent units (GRUs) to capture sequential temporal dependencies directly from the multichannel input. The GRU-based architecture was designed to classify vowels from single neuron recordings using the same input as prior models. The architecture was adapted from Willett et al (2023)[2].

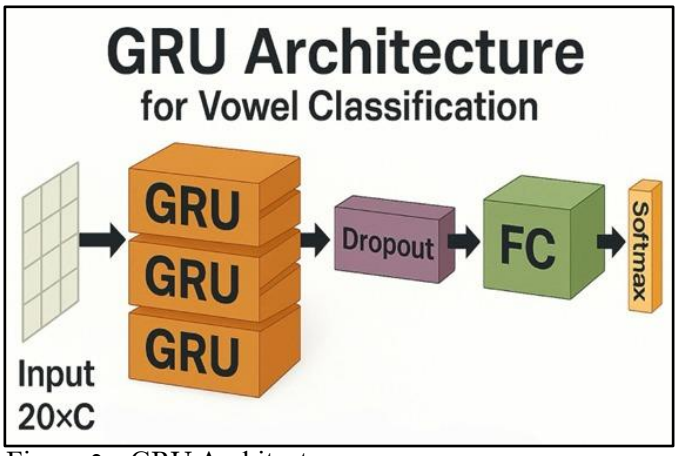


Figure 9 – GRU Architecture

As shown in Figure 9, the architecture begins by processing input matrices in the shape: (batch size, 20, C), aligning with the input convention for PyTorch RNN modules. A stack of 3 to 7 unidirectional GRU layers (standard gated recurrent units with update and reset gates, as implemented in PyTorch) is used. Each layer contains a fixed number of hidden units (ranging from 128 to 1024), all of which are selected via inner loop hyperparameter optimization.

The last hidden state of the final GRU layer serves as a compact summary of the entire input sequence. A dropout layer is applied for regularization during training, mitigating overfitting given the limited sample size. This is followed by a fully connected output layer that maps the hidden representation to class logits, corresponding to the vowel categories. These logits are interpreted via a SoftMax operation internally through the use of cross-entropy loss during training.

The model was trained using the Adam optimizer with tuned parameters ($\beta_1 = 0.9$, $\beta_2 = 0.999$, $\varepsilon = 0.1$) as stated in the paper, and L2 weight decay for regularization. The overall architecture - including the number of GRU layers, hidden size, dropout probability, and learning rate - were selected by Optuna-based hyperparameter tuning during inner cross-validation

This minimalist and sequence-focused architecture is well suited for the short, high-channel recordings available in this dataset, enabling efficient temporal pattern extraction without reliance on spatial filtering or convolution.

Having described each model architecture, we now detail the training and evaluation protocol used for assessment across all these models.

Training and Evaluation Framework - Double Cross Validation

To ensure robust model selection and unbiased generalization performance estimation, we implemented a Double Cross-Validation (Nested CV) framework, illustrated in Figure 10.

As seen in the figure, this approach consists of two nested loops – an inner and an outer loop. In the outer loop, a 10-fold StratifiedKFold is used to partition the data into 90% training and 10% testing subsets. For each outer training split, we perform an inner 5-fold cross-validation, which is designated for hyperparameter tuning using Optuna. Once the optimal hyperparameters are identified, the model is retrained from scratch on the entire outer training fold and evaluated on the held-out outer test fold. This procedure is repeated across all 10 outer folds, yielding 10 independent test accuracy scores. By isolating hyperparameter tuning from final test evaluation, this Nested CV framework effectively prevents information leakage and provides an unbiased estimate of the model's true predictive and generalization capabilities.

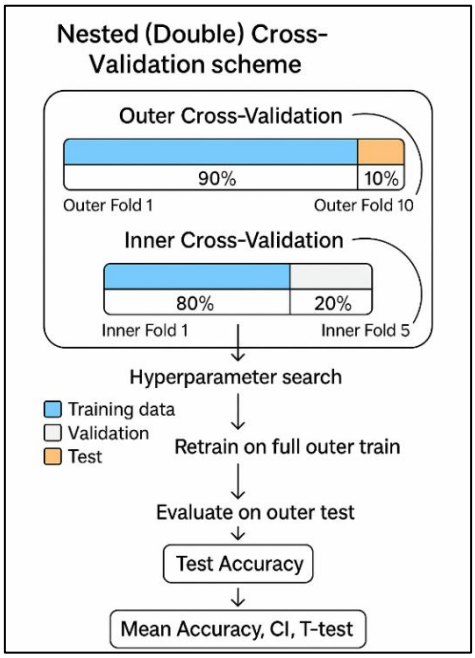


Figure 10 – Nested (Double) Cross-Validation scheme

We report descriptive statistics including the mean and standard deviation of the test accuracies, a 95% confidence interval based on the t-distribution, and the p-value of the one-sided t-test against chance-level performance (H_0: $\mu = 0.5$, H_1 $\mu > 0.5$).

More specifically, to guide model optimization within the inner loop, it explores a defined search space of hyperparameters (e.g, learning rate, weight decay, early stopping patience, batch size, dropout rate and number of epochs) from a study that consists of 50 Optuna trials. This extensive study is used to thoroughly explore the search space and then selects the configuration that achieves

the highest average validation accuracy across the inner folds. These parameters also reflect our use of multiple regularization techniques during training - specifically dropout, L2 weight decay, and early stopping - to prevent overfitting, especially given the limited training data. We also optimized the number of training epochs to improve the model's performance over time iteratively. The ranges were chosen after early experimentation to ensure they covered diverse yet reasonable configurations.

Each model architecture was trained using the Adam optimizer, due to its adaptive learning rate, momentum-based updates, and stable convergence.

We used the Weights & Biases API to track key training and validation performance metrics, such as loss and accuracy, throughout the entire training scheme.

Results

Exclusion of Patients from Analysis

In preliminary experimental runs, patient 3 exhibited worse-than-random performance on the five-vowel classification task (0.2) using the EEG-inspired CNN model. Although performance improved to above random chance on the five-class task with the alternative CNN model, all pairwise binary classifications for this patient yielded p-values above 0.05, indicating a lack of statistically significant discrimination. Consequently, test results for patient 3 are not reported.

Similarly, patient 7 exhibited worse than random performance on the five-vowel classification task (0.2) using the CNN model. Although performance improved above random chance on the five-class task with the EEG-inspired CNN model, all pairwise binary classifications for this patient also yielded p-values above 0.05, indicating a lack of statistical significance. Thus, test results for patient 7 were likewise not reported.

Additionally, patient 5 was the only patient with a non-uniform distribution of vowel classes. Given the already limited amount of data per patient and the lack of a principled way to correct this imbalance, we opted not to discard any data points for this patient. However, to ensure valid model comparison and robust statistical evaluation across subjects, we excluded patient 5 from the modeling process. This decision was made to maintain consistency and comparability across the dataset, as inclusion of a patient with an imbalanced class distribution could bias overall model evaluation.

In conclusion, five patients (1, 2, 4, 6a, 6b) remained and were included in all comparative analyses.

Methodology Overview

For classification of the five vowels, we initially evaluated both CNN-based models and a GRU-based model. The CNN models consistently demonstrated superior performance and were able to

learn meaningful features for the five-class classification task. In contrast, the GRU model failed to converge and could not learn meaningful representations for the 5-class problem. Consequently, we did not derive binary classification pairs directly from the GRU model. However, the GRU model was still included for comparison on the binary pairs that were identified as significant from the CNN models.

Approach to Model Comparison and Reporting

Given the strong performance of the CNN models, we focused our detailed analysis on results derived from these two architectures. For each possible vowel pair derived using the modeling methodology, we conducted binary classification tasks using both CNN models, retaining only those results with statistically significant performance (p-value < 0.05 for accuracy exceeding chance level).

To facilitate direct comparison across all models, we also included the corresponding binary results from the EEG-inspired CNN model wherever it exhibited statistically significant performance ($p < 0.05$). This allowed us to comprehensively fill the comparison table for each vowel pair where at least one CNN model achieved significance. For all binary pairs meeting this approach, we reported the corresponding results from all three models, allowing for a comprehensive comparison across all three architectures where applicable.

This rigorous filtering process ensured that only robust, statistically meaningful results were included in the final analysis, excluding many non-significant runs. As a result, the reported comparisons represent a substantial computational effort but ensure that only reliable findings are presented.

Patient Specific Results

Note that the lines emphasized in each table correspond to the model with the highest mean accuracy for each specific phoneme pair. Additionally, the reported results are based on the double CV training scheme, and all accuracies mentioned in this section represent mean accuracy.

Table 1 summarizes the performance of all three models - CNN, EEGNet-Inspired CNN, and GRU - on binary classification tasks involving *a* versus each of the other four vowels for patient 1. The CNN model consistently achieved the highest or near-highest mean accuracy for most phoneme pairs, notably reaching 85.0% accuracy ($p < 0.001$) for the *a* vs. *u* pair. The GRU model performed comparably (numerically on the mean accuracy) to the CNN and even outperformed it for the *a* vs. *o* pair, achieving an accuracy of 86.3% ($p < 0.001$). This represents the highest mean accuracy achieved in our project across all patients and phoneme pairs. The EEGNet-Inspired CNN generally showed lower accuracy across all pairs. These findings suggest that, for patient 1, both the CNN and GRU models are effective for classifying between vowel pairs involving *a*, while the EEGNet-

Inspired CNN is less robust. Generally, test accuracy exceeded chance level with statistical significance for nearly all models and vowel pairs ($p < 0.05$).

Table 1 - Patient 1: Binary Vowel Classification Results.

Phoneme Pair	Model	Mean Accuracy (%)	Standard Deviation (%)	95% CI (%)	p-value
a vs. *e*	**CNN**	**0.8125**	**0.1473**	**[0.7071, 0.9179]**	**0.0000**
	EEGNet-Inspired CNN	0.5750	0.1208	[0.4886, 0.6614]	0.0406
	GRU	0.7625	0.1713	[0.6400, 0.8850]	0.0005
a vs. *i*	**CNN**	**0.7750**	**0.0986**	**[0.7045, 0.8455]**	**0.0000**
	EEGNet-Inspired CNN	0.5000	0.1021	[0.4270, 0.5730]	0.5000
	GRU	0.7375	0.1497	[0.6304, 0.8446]	0.0004
a vs. *o*	CNN	0.7750	0.1646	[0.6573, 0.8927]	0.0003
	EEGNet-Inspired CNN	0.5875	0.1324	[0.4928, 0.6822]	0.0331
	GRU	**0.8625**	**0.1243**	**[0.7736, 0.9514]**	**0.0000**
a vs. *u*	**CNN**	**0.8500**	**0.1291**	**[0.7576, 0.9424]**	**0.0000**
	EEGNet-Inspired CNN	0.6750	0.1467	[0.5700, 0.7800]	0.0022
	GRU	0.8125	0.1350	[0.7159, 0.9091]	0.0000

As shown in Table 2, the EEGNet-Inspired CNN achieved the highest mean accuracy among the three models for both the *e* vs. *i* and *i* vs. *u* vowel pairs, outperforming both the CNN and GRU. However, the mean accuracy across all models remained slightly above chance accuracy (ranging from 0.55 to 0.62), indicating that these phoneme pairs were more challenging to classify for patient 2. Statistically significant p-values were observed for the models on the *e* vs. *i* task, while for *i* vs. *u*, only the EEGNet-Inspired CNN achieved statistical significance ($p = 0.0242$). These results suggest that, in this patient, discrimination between these vowels was generally difficult, with limited performance even from the best-performing model.

Table 2 - Patient 2: Binary Vowel Classification Results

Phoneme Pair	Model	Mean Accuracy (%)	Standard Deviation (%)	95% CI (%)	p-value
e vs. *i*	CNN	0.5900	0.1524	[0.4810, 0.6990]	0.0473
	EEGNet-Inspired CNN	**0.6200**	**0.1476**	**[0.5144, 0.7256]**	**0.0151**
	GRU	0.5800	0.1229	[0.4921, 0.6679]	0.0349
i vs. *u*	CNN	0.5500	0.1509	[0.4420, 0.6580]	0.1611
	EEGNet-Inspired CNN	**0.6100**	**0.1524**	**[0.5010, 0.7190]**	**0.0242**
	GRU	0.5600	0.1075	[0.4831, 0.6369]	0.0557

Table 3 presents the binary vowel classification performance for patient 4 across five vowel pairs and all three models. The GRU model achieved the highest accuracy for the *a* vs. *e* (77.0%) and *e* vs. *i* (70.0%) pairs, both with strong statistical significance ($p \leq 0.0001$). For other pairs, performance was more variable between the CNN and EEGNet-Inspired CNN models, which outperformed the GRU. The EEGNet-Inspired CNN attained 68.0% accuracy for *e* vs. *o* and 65.0% for *i* vs. *o*, with statistical significance ($p < 0.002$), outperforming the other models. For the pair *a* vs. *o*, the CNN reached the highest accuracy among the models with 77% and strong statistical significance. These findings indicate that, for patient 4, all model architectures provided robust classification for specific vowel pairs.

Table 3 - Patient 4: Binary Vowel Classification Results

Phoneme Pair	Model	Mean Accuracy (%)	Standard Deviation (%)	95% CI (%)	p-value
a vs. *e*	CNN	0.6900	0.1524	[0.5810, 0.7990]	0.0017
	EEGNet-Inspired CNN	0.6600	0.1578	[0.5471, 0.7729]	0.0054
	GRU	**0.7700**	**0.1418**	**[0.6686, 0.8714]**	**0.0001**
e vs. *i*	CNN	0.6300	0.1252	[0.5405, 0.7195]	0.0047
	EEGNet-Inspired CNN	0.5700	0.1337	[0.4743, 0.6657]	0.0662
	GRU	**0.7000**	**0.0943**	**[0.6326, 0.7674]**	**0.0000**
e vs. *o*	CNN	0.6000	0.1155	[0.5174, 0.6826]	0.0114
	EEGNet-Inspired CNN	**0.6800**	**0.1476**	**[0.5744, 0.7856]**	**0.0019**
	GRU	0.5800	0.1398	[0.4800, 0.6800]	0.0519
a vs. *o*	**CNN**	**0.7700**	**0.1059**	**[0.6942, 0.8458]**	**0.0000**
	EEGNet-Inspired CNN	0.6800	0.1476	[0.5744, 0.7856]	0.0019
	GRU	0.6300	0.0483	[0.5954, 0.6646]	0.0000
i vs. *o*	CNN	0.6200	0.1687	[0.4994, 0.7406]	0.0255
	EEGNet-Inspired CNN	**0.6500**	**0.1179**	**[0.5657, 0.7343]**	**0.0015**
	GRU	0.6400	0.1506	[0.5323, 0.7477]	0.0082

Table 4 summarizes the binary vowel classification results for patient 6a (patient 6 underwent two distinct experimental sessions with varying electrode configurations and is thus denoted as 6a and 6b for clarity) across three vowel pairs. For this patient, the CNN and EEGNet-Inspired CNN models both achieved statistically significant classification accuracy for *a* vs. *i* (70.0% and 64.0%, respectively, both $p < 0.01$). The EEGNet-Inspired CNN was the only model to achieve significance for the other two pairs (*a* vs. *o* at 59.0%, and *i* vs. *o* at 66.0%). In contrast, the GRU model failed to converge on any pair, as evidenced by its poor performance (accuracies near or below chance and non-significant p-values across all pairs). These results indicate that, for patient 6a, convolutional

architectures (CNN and EEGNet-Inspired CNN) provided more reliable classification than the GRU model, which was not able to learn meaningful vowel distinctions in this case.

Table 4 - Patient 6a: Binary Vowel Classification Results

Phoneme Pair	Model	Mean Accuracy (%)	Standard Deviation (%)	95% CI (%)	p-value
a vs. *i*	**CNN**	**0.7000**	**0.0943**	**[0.6326, 0.7674]**	**0.0000**
	EEGNet-Inspired CNN	0.6400	0.1174	[0.5560, 0.7240]	0.0022
	GRU	0.4200	0.1033	[0.3461, 0.4939]	0.9816
a vs. *o*	CNN	0.5500	0.1509	[0.4420, 0.6580]	0.1611
	EEGNet-Inspired CNN	**0.5900**	**0.1101**	**[0.5113, 0.6687]**	**0.0147**
	GRU	0.5200	0.0919	[0.4543, 0.5857]	0.2543
i vs. *o*	CNN	0.5300	0.1494	[0.4231, 0.6369]	0.2707
	EEGNet-Inspired CNN	**0.6600**	**0.1897**	**[0.5243, 0.7957]**	**0.0129**
	GRU	0.4900	0.1101	[0.4113, 0.5687]	0.6098

Table 5 presents binary vowel classification results for patient 6b across five vowel pairs. The CNN model achieved the highest accuracy for the *e* vs. *i* pair (69.2%, p = 0.0003), while the EEGNet-Inspired CNN reached its best performance on the *i* vs. *u* pair (66.7%, p = 0.0008). For several pairs, both CNN and EEGNet-Inspired CNN models provided statistically significant results, but mean accuracies were generally moderate, indicating some classification challenge for this patient. Notably, the GRU model failed to converge on all pairs - reflected in accuracies at or near chance and non-significant p-values - even for *i* vs. *u*, where it achieved 65.8% accuracy (p = 0.0009), Weights & Biases logs indicate that this was not due to model convergence, but rather reflects good performance despite a lack of true learning stability. These findings show that, for patient 6b, convolutional architectures again offered more reliable performance overall, with limited contribution from the GRU.

Table 5 - Patient 6b: Binary Vowel Classification Results

Phoneme Pair	Model	Mean Accuracy (%)	Standard Deviation (%)	95% CI (%)	p-value
e vs. *i*	**CNN**	**0.6917**	**0.1182**	**[0.6071, 0.7762]**	**0.0003**
	EEGNet-Inspired CNN	0.5667	0.1511	[0.4586, 0.6748]	0.0982
	GRU	0.5000	0.0962	[0.4312, 0.5688]	0.5000
e vs. *o*	CNN	0.6250	0.1195	[0.5395, 0.7105]	0.0046
	EEGNet-Inspired CNN	**0.6417**	**0.1245**	**[0.5526, 0.7308]**	**0.0029**
	GRU	0.5583	0.1115	[0.4786, 0.6381]	0.0662

e vs. *u*	**CNN**	**0.6167**	**0.1676**	**[0.4968, 0.7366]**	**0.0276**
	EEGNet-Inspired CNN	0.5750	0.1441	[0.4719, 0.6781]	0.0671
	GRU	0.5333	0.1255	[0.4436, 0.6231]	0.2113
i vs. *o*	CNN	0.4667	0.1372	[0.3685, 0.5648]	0.7690
	EEGNet-Inspired CNN	**0.5833**	**0.0962**	**[0.5145, 0.6522]**	**0.0114**
	GRU	0.4750	0.1524	[0.3660, 0.5840]	0.6918
i vs. *u*	CNN	0.6167	0.1809	[0.4873, 0.7461]	0.0359
	EEGNet-Inspired CNN	**0.6667**	**0.1179**	**[0.5824, 0.7510]**	**0.0008**
	GRU	0.6583	0.1142	[0.5766, 0.7400]	0.0009

Summary and Conclusions of Results

Across all included patients and vowel pairs, the comparative evaluation of three neural network architectures - CNN, EEGNet-Inspired CNN, and GRU - revealed several consistent patterns in model performance.

First, the convolutional architectures (CNN and EEGNet-Inspired CNN) generally provided the most reliable and statistically significant results. For most patients and vowel pairs, the CNN achieved either the highest or near-highest mean accuracy, frequently accompanied by low p-values ($p < 0.05$), indicating robust discrimination between vowel pairs. The EEGNet-Inspired CNN also delivered competitive results, particularly performing best on most reported vowel pairs (9 out of 19) and for specific patients (e.g., patient 2), sometimes outperforming the CNN.

Second, the GRU model exhibited highly variable performance across patients. While it matched or exceeded CNN accuracy for selected pairs (such as *a* vs. *o* for patient 1 and *a* vs. *e* for patient 4), it failed to converge or to deliver meaningful results for others, notably for both patient 6a and patient 6b. In these cases, GRU classification accuracies hovered near chance and were accompanied by non-significant p-values, further confirmed by Weights & Biases logs indicating a lack of convergence during training.

Third, there was noticeable heterogeneity in classification difficulty across vowel pairs and patients. In some cases, such as for patient 1 and patient 4, accuracies frequently exceeded 70%, demonstrating that, for certain individuals and phoneme contrasts, the neural signal contained strong, discriminative information. However, for other patients (e.g., patient 2 and patient 6b), mean accuracies were closer to chance level, and significant performance was less consistently achieved. This variability likely reflects both inter-patient differences in the extent to which recorded neurons are involved in speech processing, and intrinsic differences in vowel pair discriminability.

Fourth, none of the models tested emerged as a universal "best" model; rather, performance varied considerably depending on the specific patient and phoneme pair under study. For example, the EEGNet-Inspired CNN performed best on all vowel pairs for patient 2, but never did so for patient

1. Similarly, no single model consistently outperformed the others across all vowel pairs for more than one specific patient.

These findings underscore the importance of considering individual variability in electrode placement and brain anatomy. As a result, further analysis and larger datasets will be required to fully account for these sources of heterogeneity and to optimize performance across diverse patient populations.

Overall, our project demonstrates the technical feasibility of decoding speech features from single-neuron recordings using deep learning models - specifically CNN and GRU-based neural network architectures - providing a foundation for the future development of communication-restoring BCIs. Our study also extends the existing literature by evaluating deep learning models that have not previously been applied to single-neuron activity recorded in the human frontal lobe for speech decoding. These results provide strong evidence that deep learning methods, particularly convolutional architectures, can effectively support the development of neural speech prostheses in clinical populations. Nevertheless, continued methodological refinement, patient-specific adaptation, and broader validation are required before translation to real-world BCI applications. These results directly address our project goal of benchmarking deep learning methods for decoding vowel articulations from single-neuron recordings of epilepsy patients.

Ethical Considerations

This project, which explored decoding of vowel articulations from single-neuron recordings using DL based architectures, lies at the intersection of neuroscience, artificial intelligence, and human communication. While this setup offers significant potential for BCI applications, particularly for individuals with speech impairments - they also introduce several ethical dilemmas that merit careful examination.

First and foremost is the issue of neural privacy. Decoding internal states such as speech intentions from brain activity raises concerns about the boundaries between thought and action. As decoding models become more accurate and generalized, there exists a risk that neural data could be used to infer mental content beyond the individual's consent or awareness. This is especially important in clinical or research settings where participants may be vulnerable, such as those with neurological or communication disorders. Ensuring robust data governance policies, explicit informed consent procedures, and clearly defined limits on data usage is therefore essential.

Second, data ownership and informed consent present critical challenges. Neuronal data, by nature, is highly sensitive and person specific. The question of who owns the neural data - patients, institutions, or researchers - has profound implications for consent, data sharing, and potential commercialization. Any deployment of models trained on such data must respect the autonomy of participants and safeguard against unclear uses of their neural information.

Third, lack of model explainability poses a critical challenge. While the deep learning architectures employed in this project achieved promising classification accuracy, it sacrifices interpretability. This reflects an inherent trade-off in DL models: CNNs, while powerful, often operate as "black boxes," making it difficult to understand precisely which neural patterns contribute to specific classification decisions, especially in a way that's meaningful for clinical or patient-facing contexts. As neural decoding moves toward clinical use, model transparency becomes increasingly important. Future work could incorporate explainability techniques - such as gradient-based saliency maps or channel-wise importance analysis - to identify which time points or neural channels contribute most to the model's decisions. Ensuring that models are accompanied by interpretable representations, or post-hoc explainability tools, is essential not only for debugging and model trustworthiness but also for establishing ethical transparency in sensitive neural applications. This could improve both scientific understanding and ethical deployment by ensuring that the model's predictions are interpretable and aligned with intended outcomes.

Finally, there is an overarching risk of bias and inequality. Neural decoding models may not generalize across individuals due to differences in brain anatomy, electrode placement, or neural variability. If training data are limited to certain demographic or clinical groups, downstream systems may fail to perform equitably across populations. As such, fairness in model development, transparent evaluation, and inclusive data representation are vital to prevent the marginalization of underrepresented groups in future neurotechnological applications.

To summarize, while the project demonstrates promising technical capabilities, it must be situated within a framework that prioritizes fair and inclusive development, long-term social responsibility and individual autonomy. Anticipating these dilemmas and embedding ethical safeguards early in the research pipeline is essential to ensure that neural-AI systems are both effective and aligned with human values.

Appendices

Appendix 1 - Dr. Tankus CNN Model

The following section is adapted from the original work of Tankus et al., (2024)[1].

Dr.Tankus developed a Convolutional Neural Network (CNN) to classify two vowel phonemes - a and e - based on spike-sorted single-neuron recordings from a single epilepsy patient implanted with depth electrodes.

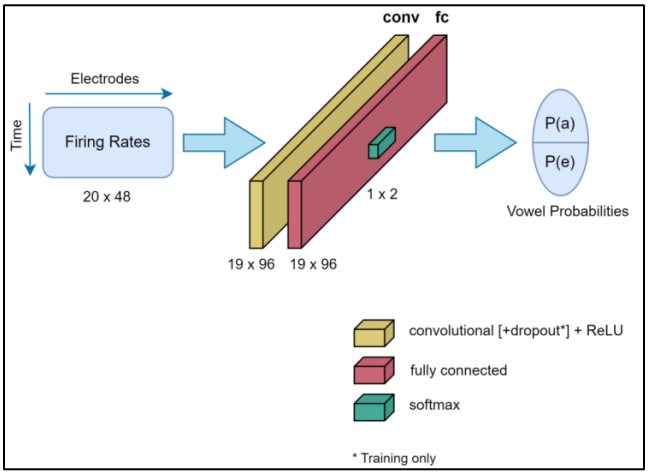


Figure 11 – Dr. Tankus' CNN Architecture

As seen in Figure 11 and described in Tankus et al., (2024)[1], each trial (articulation) was represented as a 20×48 matrix, encoding firing rates across 20 time bins and 48 neuronal channels, which corresponded to different brain locations where depth electrodes were implanted. The CNN architecture included a 2D convolutional layer which used a 2×48 kernel, as well as 96 activation maps, spanning both temporal and spatial dimensions, followed by ReLU activations, dropout regularization (during training), and a fully connected output layer.

Training was performed using the Adam optimizer and Cross-Entropy loss, with 20-fold cross-validation over the 40 trials (articulations) per vowel (80 total). The model achieved a classification accuracy of 95.0% ± 2.2% (95% CI), demonstrating that CNNs can reliably capture vowel-discriminative neural patterns from high-resolution brain signals - an insight that directly motivated our effort to extend this approach to multiple vowels and patients.

References

1. Tankus, A., Stern, E., Klein, G., et al. (2024). A Speech Neuroprosthesis in the Frontal Lobe and Hippocampus: Decoding High-Frequency Activity into Phonemes. Neurosurgery, 96(2): 356–364
2. Willett, F. R., Kunz, E. M., Fan, C., et al. (2023). A high-performance speech neuroprosthesis. Nature, 620(7973): 117–124
3. Guenther, F. H., Brumberg, J. S., Wright, E. J., et al. (2009). A wireless brain–machine interface for real-time speech synthesis. PLoS ONE, 4(12): e8218.
4. Brumberg, J. S., Wright, E. J., Andreasen, D. S., et al. (2011). Classification of intended phoneme production from chronic intracortical microelectrode recordings in speech motor cortex. Frontiers in Neuroscience, 5: 65.
5. Kellis, S., Miller, K., Thomson, K., et al. (2010). Decoding spoken words using local field potentials recorded from the cortical surface. Journal of Neural Engineering, 7(5): 056007.
6. Pei, X., Barbour, D. L., Leuthardt, E. C., et al. (2011). Decoding vowels and consonants in spoken and imagined words using electrocorticographic signals in humans. Journal of Neural Engineering, 8(4): 046028.
7. Mugler, E. M., Patton, J. L., Flint, R. D., et al. (2014). Direct classification of all American English phonemes using signals from functional speech motor cortex. Journal of Neural Engineering, 11(3): 035015.
8. Herff, C., Heger, D., de Pesters, A., et al. (2015). Brain-to-text: decoding spoken phrases from phone representations in the brain. Frontiers in Neuroscience, 9: 217.
9. Moses, D. A., Metzger, S. L., Liu, J. R., et al. (2021). Neuroprosthesis for decoding speech in a paralyzed person with anarthria. New England Journal of Medicine, 385(3): 217–227.
10. Lawhern, V. J., Solon, A. J., Waytowich, N. R., et al. (2018). EEGNet: a compact convolutional neural network for EEG-based brain–computer interfaces. Journal of Neural Engineering, 15(5): 056013.
11. Makin, J. G., Moses, D. A., & Chang, E. F. (2020). Machine translation of cortical activity to text with an encoder–decoder framework. Nature Neuroscience, 23: 575–582.
12. Anumanchipalli, G. K., Chartier, J., & Chang, E. F. (2019). Speech synthesis from neural decoding of spoken sentences. Nature, 568(7753): 493–498.
13. Dashtipour, K., Chen, J. J., & Nemeth, E. F. (2018). Speech disorders in Parkinson's disease: pathogenesis, characteristics, and management. Journal of Neurology, 265(9), 2163–2172.
14. Pearson, K. (1901). On lines and planes of closest fit to systems of points in space. The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science, 2(11), 559–572.

15. Van Der Maaten L & Hinton G. (2008). Visualizing data using t-SNE. *Journal of Machine Learning Research.* 2008; 9(Nov): 2579–2605.
16. Cho et al. (2014). Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 1724–1734