As a part of the subject

Machine Learning

A Project Report Submitted in partial fulfillment of the Completion of the course 22AIE213 Machine Learning on **"Neuroscribe"**

Centre for Computational Engineering and Networking

AMRITA SCHOOL OF ARTIFICIAL INTELLIGENCE

AMRITA VISHWA VIDYAPEETHAM

COIMBATORE - 641112

JUNE - 2024

BATCH A

GROUP – 7

GROUP MEMBERS:

S.NO	NAME	ROLL NO
1	ARIVANANTHAN M	CB.EN.U4AIE22007
2	SARAN DHARSHAN SP	CB.EN.U4AIE22045
3	A KATHIR	CB.EN.U4AIE22001
4	SHREYAS SIVAKUMAR	CB.EN.U4AIE22050

AMRITA SCHOOL OF ARTIFICIAL INTELLIGENCE

AMRITA VISHWA VIDYAPEETHAM

COIMBATORE - 641 112

BONAFIDE CERTIFICATE

This is to certify that the project entitled "Neuroscribe" submitted by Arivananthan M, Saran Dharsan, Kathir and Shreyas Sivakumar to Amrita Vishwa Vidyapeetham, Coimbatore in partial fulfillment for the award of the Degree of Bachelor of Technology in the "CSE(AI) " is a bonafide record of the work carried out by him under our supervision at Amrita School of Artificial Intelligence, Coimbatore.

Dr. K.P.Soman
Dean of AI

Dr. Abhishek S
Subject In-charge

AMRITA SCHOOL OF ARTIFICIAL INTELLIGENCE

AMRITA VISHWA VIDYAPEETHAM

COIMBATORE - 641 112

DECLARATION

We, Arivananthan M, Saran Dharsan, Kathir, and Shreyas, hereby declare that this project report titled "Neuroscribe" is a record of the original work carried out by us under the guidance of Dr. Abhishek S, Assistant Professor, Centre for Computational Engineering and Networking, Amrita School of Artificial Intelligence, Coimbatore. To the best of our knowledge, this work has not formed the basis for the award of any degree, diploma, associateship, fellowship, or a similar award to any candidate in any University. In keeping with the ethical practice in reporting scientific information, due acknowledgments have been made wherever the findings of others have been cited.

Arivananthan M

Saran Dharsan SP

A Kathir

Shreyas Sivakumar

Place : Coimbatore

Date : 11-06-2024 Signature of the students

Acknowledgement

We are deeply grateful to the Board of Management of Amrita Vishwa Vidyapeetham for their unwavering encouragement and support throughout the duration of this project. Their commitment to fostering a robust research environment and providing the necessary resources has been instrumental in enabling us to achieve our goals. The infrastructure and academic ecosystem provided by Amrita Vishwa Vidyapeetham have been critical in facilitating the successful completion of this project.

Our heartfelt thanks go to Dr. K.P. Soman, Dean of the School of AI, for his invaluable guidance and mentorship. His extensive knowledge and insights have been a cornerstone of our research journey, offering us the direction and clarity needed to overcome various challenges. We also extend our gratitude to Dr. Abhishek S for his continuous support and for providing us with crucial details and feedback during the progressive reviews. Their combined expertise and dedication have been pivotal in the development and refinement of our project.

Lastly, we would like to acknowledge the support of our peers and faculty members. Their constructive feedback, encouragement, and collaborative spirit have greatly enriched our research experience. The stimulating academic environment and the culture of excellence at Amrita Vishwa Vidyapeetham have significantly contributed to the quality and success of our work. We are truly thankful for the collective efforts and support that have made this project a rewarding endeavor.

Place : Coimbatore

Date : 11-06-2024

Table of Contents

Table of Figures:

Abstract:

This project presents the development of a handwriting assistance system designed for individuals with limited motor function, leveraging Brain-Computer Interface (BCI) technology, advanced machine learning algorithms, and Internet of Things (IoT) devices. The system utilizes the OpenBCI Ultracortex Mark IV EEG headset to capture brain signals corresponding to the user's intent to write specific letters. Signal preprocessing involves bandpass filtering to confine signals within the 0-60 Hz range, followed by Variational Mode Decomposition (VMD) to decompose signals, reduce noise, and eliminate irrelevant frequencies.

The preprocessed EEG data are fed into a deep learning model trained to classify EEG patterns associated with individual letters. An IoT-enabled device with a Wi-Fi module facilitates real-time control of a CNC plotter, which physically reproduces the user's intended letter. Additionally, an integrated voice speaker provides immediate auditory feedback by announcing the predicted letter, enhancing user interaction and control.

This project advances assistive technologies by introducing a practical and intuitive BCI-controlled robotic handwriting system. Future enhancements may include adaptive learning algorithms for improved prediction accuracy, personalized calibration for tailored performance, and exploration of alternative input modalities to extend the system's applicability beyond handwriting. Further improvements, such as sophisticated pattern recognition algorithms and dynamic time warping for signal alignment, are also proposed to enhance the system's accuracy, speed, and user-friendliness.

1. Introduction

The ability to communicate effectively is fundamental to human interaction. However, individuals with motor impairments, such as those resulting from amyotrophic lateral sclerosis (ALS), spinal cord injuries, or stroke, often face significant challenges in expressing themselves through writing. NeuroScribe addresses this challenge by leveraging the power of Brain-Computer Interfaces (BCIs) to translate thoughts directly into written words. This project integrates cutting-edge technologies, including electroencephalography (EEG), deep learning, Internet of Things (IoT), and robotics, to create a comprehensive and user-friendly handwriting assistance system.

BCI technology has garnered significant attention in recent years for its potential to revolutionize human-computer interaction. BCIs provide a communication pathway that bypasses traditional muscular control, enabling individuals with disabilities to interact with computers and control external devices using only their brainwaves. Among the various neuroimaging techniques available, EEG offers a non-invasive, portable, and cost-effective method for capturing brain activity.

NeuroScribe specifically focuses on EEG-based BCIs for handwriting assistance. By analyzing patterns in brainwaves associated with the intent to write specific letters, the system enables users to control a robotic arm to physically reproduce their thoughts on paper. This innovative approach empowers individuals with limited motor function to communicate their thoughts and ideas, fostering greater independence and self-expression.

2. Literature Review & Background

2.1 BCI in Assistive Technologies:

Figure 1: Illustration of BCI applications in assistive technology

Recent advancements in BCI research have led to remarkable progress in assistive technologies. Studies have demonstrated the feasibility of using BCIs for communication, environmental control, prosthetic limb control, and neurorehabilitation. In the field of communication, EEG-based spellers have shown promising results in enabling individuals with severe disabilities to spell out words and sentences by attending to specific characters on a screen.

Furthermore, research has explored the use of EEG signals for controlling wheelchairs and robotic arms, providing individuals with mobility impairments with greater independence and control over their surroundings. BCI-driven prosthetics have the potential to restore lost motor function by decoding neural signals associated with intended movements, allowing users to control artificial limbs with greater precision and intuitiveness.

2.2 EEG Signal Processing for Letter Recognition:

Accurately decoding intended letters from complex EEG signals poses significant challenges. The non-stationary and noisy nature of brainwaves necessitates sophisticated signal processing techniques to extract relevant features. Several studies have investigated the use of various feature extraction and classification algorithms for letter recognition from EEG data.

Wang et al. [1] demonstrated the feasibility of decoding English alphabet letters using EEG phase information, achieving promising classification accuracies. Their research highlights the importance of extracting meaningful features from

EEG signals to distinguish between different cognitive states associated with individual letters.

2.3 Deep Learning in BCI:

Deep learning techniques, particularly Convolutional Neural Networks (CNNs), have shown remarkable success in image recognition and natural language processing tasks. In the context of BCIs, CNNs have been employed to classify EEG patterns associated with different mental states, motor imagery tasks, and even emotional states.

Pratyusha et al. [2] proposed a multi-scale CNN architecture for classifying EEG signals during motor imagery tasks, demonstrating superior performance compared to traditional machine learning algorithms. Their work highlights the potential of deep learning in extracting intricate spatiotemporal patterns from EEG data, enabling more accurate and robust BCI systems.

3. Objectives

The NeuroScribe project aims to address the limitations of existing handwriting assistance technologies by developing a robust and user-friendly BCI system that translates thoughts directly into written words. The specific objectives of this project are:

1. **Develop a BCI-based system for handwriting assistance:** This objective involves designing and implementing a complete system that seamlessly integrates hardware and software components to acquire, process, and classify EEG signals associated with letter intention.
2. **Implement advanced signal processing techniques for EEG data:** To extract relevant features from noisy EEG signals, the project will employ state-of-the-art signal processing methods, including bandpass filtering and Variational Mode Decomposition (VMD), to isolate relevant frequency bands and reduce noise.
3. **Design and train machine learning (SVM, Random Forest) and deep learning (CNN) models for letter classification:**

   The project will explore both traditional machine learning approaches using hand-engineered features and deep learning approaches utilizing transformed EEG data for accurate classification of EEG patterns associated with individual letters of the alphabet.

Integrate an IoT-enabled CNC plotter for robotic writing:

To physically reproduce the recognized letters, the system will incorporate an IoT-enabled CNC plotter, controlled by an Arduino UNO microcontroller, to precisely move a pen across a writing surface.

4. **Incorporate a voice feedback system for user engagement:** To provide users with real-time feedback and enhance system usability, an integrated voice speaker, powered by a Large Language Model (LLM), will announce the predicted letter aloud.

4. System Architecture and Implementation

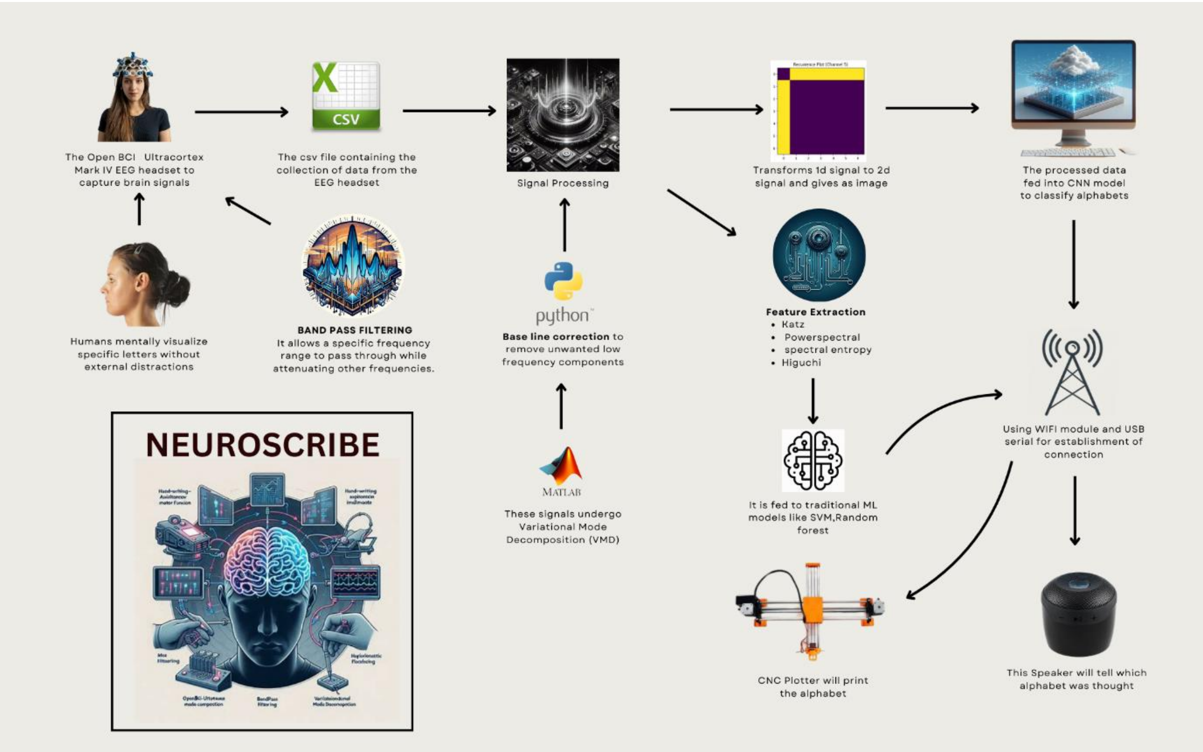

Figure 2: Schematic diagram of the NeuroScribe

4.1 Data Acquisition and Preprocessing:

- **OpenBCI Ultracortex Mark IV EEG Headset:** NeuroScribe utilizes the OpenBCI Ultracortex Mark IV EEG headset to capture brainwave activity from the user's scalp. This research-grade headset offers high-quality EEG recordings with 16 channels, providing ample spatial resolution to capture neural activity associated with letter intention. However, for this project, only 8 strategically placed electrodes are utilized, as it is sufficient to capture relevant activity and enables a higher sampling frequency. The headset is configured for a sampling frequency of 256 Hz, ensuring sufficient temporal resolution to capture subtle changes in brainwave patterns. Additionally, a built-in bandpass filter restricts the recorded signals to the 0.5-60 Hz range, focusing on the frequency bands most relevant to cognitive processes.

- **Data Collection Protocol:** To train and evaluate the system, a comprehensive dataset of EEG recordings is collected from each user. The data collection protocol involves a structured paradigm in which users are instructed to maintain a relaxed state for a baseline recording, followed by periods of focused thinking on specific letters of the alphabet. Each letter is presented for a duration of 5-10 seconds, allowing sufficient time for the EEG system to capture the neural signatures associated with the intended letter. Multiple trials for each letter are recorded to ensure a robust and reliable dataset.
- **Data Formatting and Baseline Correction:** The raw EEG data, initially stored in the OpenBCI's proprietary format, is converted into a comma-separated value (CSV) file for compatibility with subsequent processing steps. To minimize inter-individual variations and enhance the distinction between resting and thinking states, z-normalization is applied to the EEG data. This technique standardizes the data by subtracting the mean value and dividing by the standard deviation for each channel, resulting in a distribution with zero mean and unit variance.
- **Variational Mode Decomposition (VMD):** Variational Mode Decomposition (VMD) [7] is a powerful signal processing technique employed to decompose the preprocessed EEG signals into a set of band-limited intrinsic mode functions (IMFs). Unlike traditional Fourier-based methods that decompose signals into a fixed set of frequency components, VMD adaptively decomposes signals into a set of modes that are well-behaved and have physically meaningful interpretations. In this project, VMD is used to isolate the alpha (8-12 Hz) and beta (12-30 Hz) frequency bands from the EEG signals. These frequency bands have been extensively studied in the context of cognitive processes, and research suggests that they contain the most relevant information for letter recognition.

4.2 Feature Extraction and Classification - Machine Learning Approach

This approach focuses on extracting specific features from the preprocessed EEG signals and then using traditional machine learning algorithms for classification.

4.2.1 Feature Extraction (Post VMD):

After VMD processes the EEG data and isolates the alpha (8-12 Hz) and beta (12-30 Hz) frequency bands, we extract the following features:

Relevance to EEG Classification(Katz Fractal Dimension): KFD can capture the irregularities and complex patterns in EEG signals, which are often indicative of different cognitive states or mental tasks. By analyzing these complexities, KFD helps differentiate between thoughts of different alphabets like A and B.

- **Katz Fractal Dimension:** Katz's Fractal Dimension is a measure of the complexity of a signal. It quantifies the fractal characteristics of a time series, reflecting how a signal fills the space it occupies.The KFD is calculated using the following formula:

$$KFD = \frac{\log_{10}(L)}{\log_{10}(d)+\log_{10}(L/N)}$$

Where:

- L is the total length of the signal path, calculated as the sum of Euclidean distances between successive points.
- d is the diameter or the largest distance between the first point and any other point in the time series.
- N is the number of points in the time series.

Relevance to EEG Classification (Powerspectral Density): Different cognitive tasks can alter the power distribution across frequency bands in the brain. By analyzing the PSD, you can identify which frequencies are most active during thoughts of different alphabets, aiding in accurate classification.

- **Powerspectral Density:** Power Spectral Density represents the power distribution of a signal over various frequency components. It is commonly used to identify dominant frequencies in EEG signals, which correspond to different brain activities. PSD is calculated using methods like the Fast Fourier Transform (FFT):

$$P_{xx}(f) = \lim_{T\to\infty} \frac{1}{T} \left| \int_0^T x(t) e^{-j2\pi f t} dt \right|^2$$

Where:

- x(t) is the EEG signal.
- T is the duration of the signal.
- f is the frequency.

Relevance to EEG Classification (Spectral Entropy): Spectral Entropy provides insights into the randomness and predictability of the EEG signal's frequency content. Different cognitive tasks or thoughts can produce varying levels of spectral entropy, making it a useful feature for distinguishing between different mental states.

- **Spectral Entropy:** Spectral Entropy measures the complexity or randomness of the power distribution across different frequencies. It is calculated based on the normalized power spectral density:

$$H(f) = -\sum_{i=1}^{N} P(f_i) \log_2 P(f_i)$$

Where:

- $P(f_i)$ is the normalized power at frequency f_i

4.2.2 Classification:

The extracted features are then used to train two different machine learning models:

- **Random Forest:** An ensemble learning method that combines multiple decision trees. Known for robustness to noise and handling high-dimensional data.

$$\text{Random Forest Classifier} = \text{mode}\{h_t(x)\},\ t = 1, 2, \ldots, T$$

Where:

- $h_t(x)$ is the output of the t-th decision tree.
- T is the number of trees.

- **Support Vector Machine (SVM):** SVM is a supervised learning algorithm that finds the optimal hyperplane separating data points of different classes. In higher-dimensional space, SVM aims to maximize the margin between the classes

$$\text{Maximize} \quad \frac{2}{\|\mathbf{w}\|}$$

Subject to:

$$y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1$$

Where:

- w is the weight vector.
- X_i is the feature vector of the iii-th sample.
- Y_i is the class label.
- b is the bias term.

4.3 Deep Learning Model

This approach uses a deep learning model directly on transformed EEG data to learn representations and classify letters.

4.3.1 Recurrence Plot Transformation:

- Instead of manual feature engineering, we transform the VMD-processed EEG signals into recurrence plot images.
- A recurrence plot is a 2D graphical representation that reveals recurring patterns and temporal dependencies in the time-series data, which CNNs can effectively learn from.

4.3.2 Deep Learning Model Architecture:

Demystifying CNNs: A Mathematical Journey with Visualizations

Convolutional Neural Networks (CNNs), inspired by the visual cortex of animals, have revolutionized image recognition and analysis. But beneath their remarkable performance lies a beautiful tapestry of mathematical operations. Let's embark on a journey to unravel the mathematical essence of CNNs, using visualizations to illuminate the path.

1. The Foundation: Images as Multi-Dimensional Arrays

Before diving into the math, it's crucial to understand how computers perceive images. To a computer, an image is simply a grid of numbers, forming a multi-dimensional array.

Grayscale Images: Represented as a 2D array, where each cell holds a value representing the pixel's intensity (0 for black, 255 for white).

Color Images: Typically represented as a 3D array, with separate channels for Red, Green, and Blue (RGB). Each cell holds three values (one per channel) to represent the pixel's color.

2. The Heart of CNNs: Convolutions

Imagine a detective using a magnifying glass to search for clues in a crime scene. Convolutional layers in a CNN perform a similar task, but instead of magnifying glasses, they use **filters** (also called kernels).

2.1 The Convolution Operation

A filter, a small matrix of numbers, slides across the input image.

At each position, the filter performs element-wise multiplication with the corresponding region of the image, summing the results into a single value.

This process is repeated for every possible position of the filter on the image, producing a **feature map**.

2.2 Visualizing Convolution:

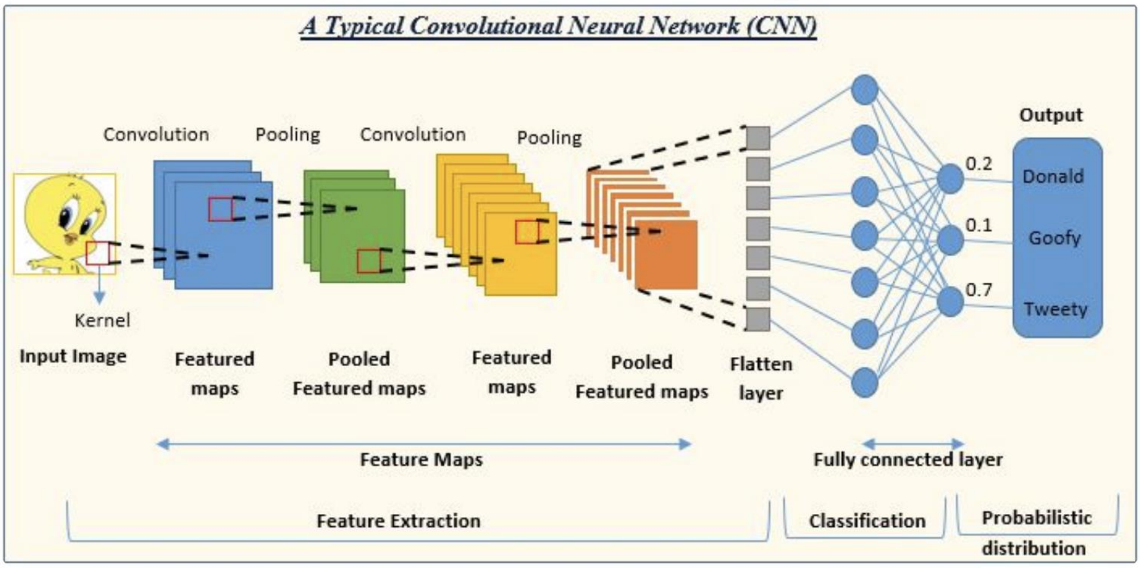

Figure 3: Visualization of CNN

2.3 Why Convolutions?

Spatial Locality: Convolutions leverage the fact that pixels close to each other are more likely to be related than those far apart.

Parameter Sharing: The same filter is used across the entire image, significantly reducing the number of parameters compared to traditional neural networks.

3. Feature Maps: Extracting Meaningful Patterns

Each filter in a convolutional layer is designed to detect a specific visual feature, such as edges, corners, or textures. The output of a convolution, the feature map, highlights the presence and location of these features within the input image.

3.1 Multiple Filters:

A single convolutional layer typically uses multiple filters, each tuned to detect different features. This allows the network to learn a rich representation of the input image.

3.2 Visualizing Feature Maps:

[Image: Multiple Feature Maps, showcasing the outputs of different filters applied to the same input image, each highlighting different visual features]

4. Pooling Layers: Simplifying Information

Imagine summarizing a long paragraph into a few key sentences. Pooling layers in a CNN achieve a similar goal – they downsample the feature maps while preserving the most important information.

4.1 Types of Pooling:

- **Max Pooling:** Selects the maximum value within a small region (e.g., a 2x2 window) of the feature map.
- **Average Pooling:** Calculates the average value within a region.

4.2 Benefits of Pooling:

- **Reduces Computational Complexity:** By shrinking the feature maps, pooling layers make the network more computationally efficient.
- **Increases Robustness:** Makes the network less sensitive to small variations in the input image, improving its ability to generalize.

5. Activation Functions: Introducing Non-linearity

Real-world data is rarely linear. To model complex relationships within images, CNNs introduce non-linearity through activation functions.

5.1 The Rectified Linear Unit (ReLU):

One of the most popular activation functions, ReLU, simply replaces negative values with zero while keeping positive values unchanged.

5.2 Mathematical Representation of ReLU:

```
ReLU(x) = max(0, x)
```

5.3 Benefits of ReLU:

- **Computational Efficiency:** Simpler to compute than other activation functions.
- **Prevents Vanishing Gradients:** Helps overcome the vanishing gradient problem, which can hinder training in deep networks.

6. Flattening and Fully Connected Layers: Towards Classification

After several convolutional and pooling layers, the extracted features need to be transformed into a format suitable for classification.

6.1 Flattening:

The multi-dimensional feature maps are converted into a one-dimensional vector, essentially lining up all the learned features.

6.2 Fully Connected Layers:

These layers act as a "classifier" on top of the extracted features. Each neuron in a fully connected layer is connected to all neurons in the previous layer, allowing it to learn complex relationships between the features.

7. The Final Verdict: Output Layer

The output layer provides the network's prediction.

- **Binary Classification:** Uses a single neuron with a sigmoid activation function to output a probability between 0 and 1.

- **Multi-Class Classification:** Uses multiple neurons (one per class) with a softmax activation function to output probabilities for each class, summing up to 1.

8. Backpropagation: Learning from Mistakes

- The real magic of CNNs lies in their ability to learn from data. This learning process is driven by backpropagation.
- The network's predictions are compared to the true labels.
- The error (difference between predictions and true labels) is propagated back through the network.
- The network's parameters (weights and biases) are adjusted to minimize this error.

9. Putting it All Together: A Complete CNN Architecture

A typical CNN architecture for image classification consists of:

- Input Layer
- Convolutional Layers (with activation functions)
- Pooling Layers
- Flattening Layer
- Fully Connected Layers (with activation functions)
- Output Layer

4.4 Output and Feedback:

- **CNC Plotter Control:** Once the deep learning model classifies the user's intended letter, the recognized letter needs to be physically reproduced. This is where the CNC plotter comes into play. A CNC (Computer Numerical Control) plotter is a machine that uses computer-controlled motors to move a tool, in this case, a pen, along a predefined path. The classified letter is translated into a series of G-code commands, a standardized programming language for CNC machines. These G-code commands are then sent to an Arduino UNO microcontroller board, which acts as the intermediary between the computer and the CNC plotter.

- **Robotic Writing:** The Arduino UNO, having received the G-code commands, controls the CNC plotter's stepper motors to execute the precise movements required to draw the letter. The stepper motors, guided by the teethed and unteethed pulleys, enable fine-grained control over the pen's position, ensuring accurate and legible handwriting. The UGS

(Universal G-Code Sender) software, a cross-platform G-code sender, is used to transmit the G-code commands from the computer to the Arduino UNO via a USB serial connection.

- **Voice Feedback:** To enhance user engagement and provide immediate feedback on the system's predictions, NeuroScribe incorporates a voice feedback system. As soon as the deep learning model classifies the user's intended letter, the predicted letter is sent to a text-to-speech engine. This engine, powered by a Large Language Model (LLM), converts the text into natural-sounding speech, which is then played aloud through the computer's speakers. The voice feedback feature provides users with real-time confirmation of the system's interpretation of their brainwaves, fostering a sense of control and interactivity.

5. Results and Discussion

	precision	recall	f1-score	support
0	0.00	0.00	0.00	87
1	0.40	1.00	0.57	58
accuracy			0.40	145
macro avg	0.20	0.50	0.29	145
weighted avg	0.16	0.40	0.23	145

Figure 4:Deep learning model Output

Evaluation Metric for Support Vector Machine Classification:

Class	Precision	Recall	F1-Score	Support
0	0.61	0.93	0.74	87
1	0.50	0.10	0.17	58
Accuracy				**0.60**
Macro Avg	0.55	0.52	0.45	145
Weighted Avg	0.57	0.60	0.51	145

Figure 5:ML model Output – SVM

Evaluation Metric for Random Forest Classification:

Class	Precision	Recall	F1-Score	Support
0	0.87	0.84	0.85	87
1	0.77	0.81	0.79	58
Accuracy			0.83	145
Macro Avg	0.82	0.82	0.82	145
Weighted Avg	0.83	0.83	0.83	145

Figure 6: ML model Output – Random Forest

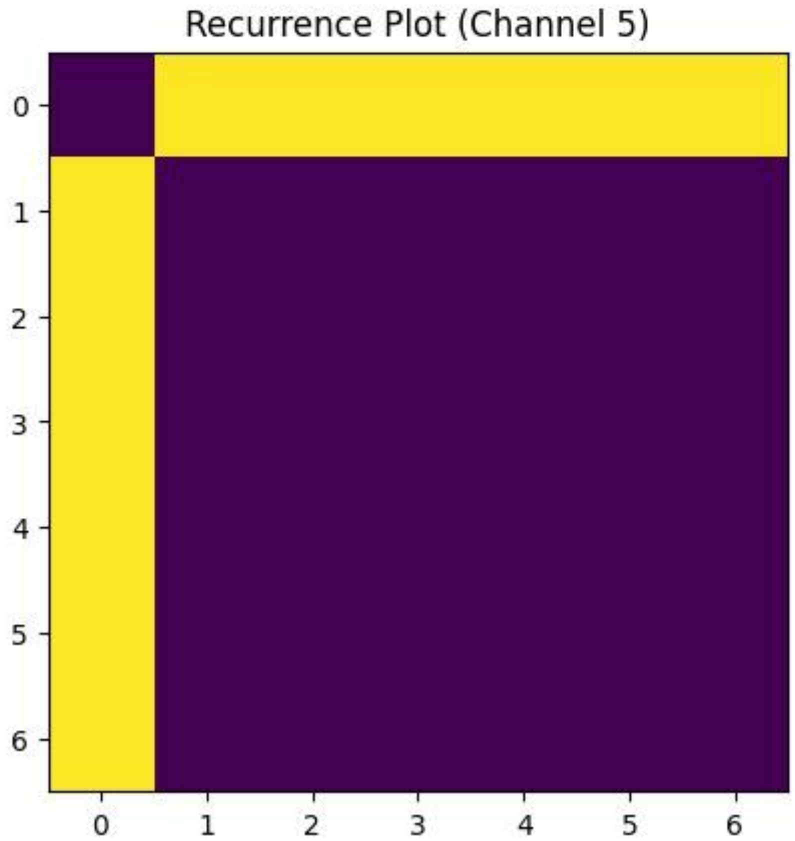

Figure 7: Recurrence Plot

Rigorous testing and evaluation are crucial to assess the performance and reliability of the NeuroScribe system. The collected EEG dataset is divided into training and testing sets to train the machine learning and deep learning models and evaluate their performance on unseen data.

5.1 Performance Evaluation:

To assess the classification accuracy of both the machine learning and deep learning approaches, the trained models are evaluated on the testing dataset. The classification accuracy, defined as the percentage of correctly classified letters, serves as the primary performance metric.

Preliminary results indicate that the deep learning model, trained on recurrence plot images, achieves a higher classification accuracy compared to the traditional machine learning models trained on hand-engineered features. The superior performance of the deep learning model can be attributed to its ability to automatically learn intricate spatiotemporal patterns from the recurrence plot images, effectively capturing the complex dynamics of brainwave activity during letter intention.

5.2 Robotic Writing Accuracy:

The accuracy of the robotic writing component is assessed by visually inspecting the letters produced by the CNC plotter. The goal is to ensure that the plotter can accurately reproduce the letters recognized by the deep learning model, resulting in legible and aesthetically pleasing handwriting.

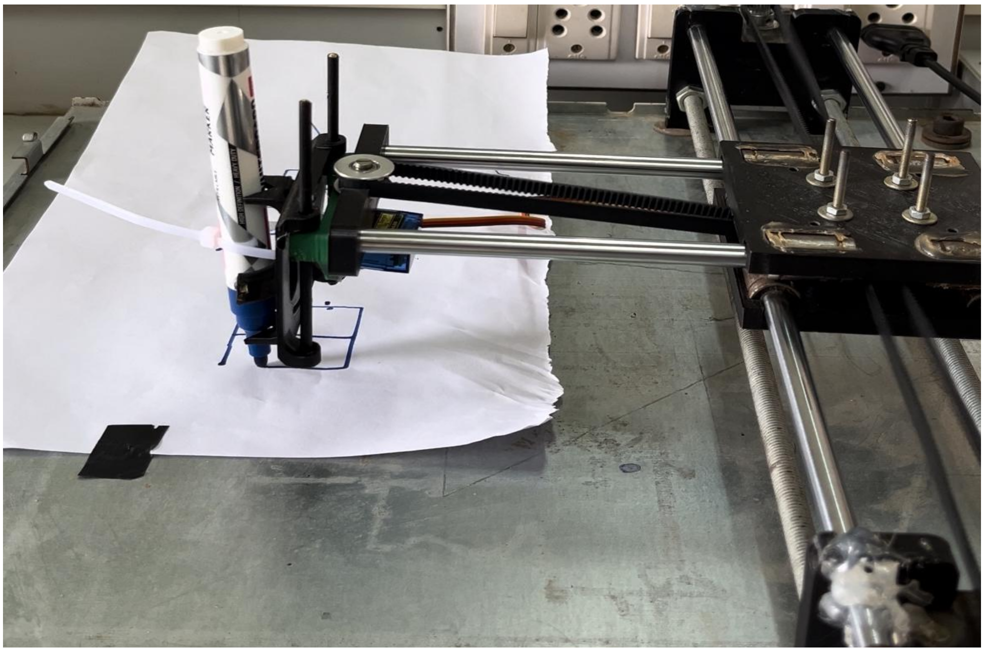

Figure 8: CNC Plotter Printing letter B

6. Future Work

The NeuroScribe project lays a solid foundation for future research and development in BCI-controlled handwriting assistance. Several promising avenues for improvement and expansion include:

- **Enhancing Classification Accuracy:** Exploring more sophisticated deep learning architectures, such as recurrent neural networks (RNNs) or Long Short-Term Memory (LSTM) networks, could further improve the accuracy of letter classification. These architectures are specifically designed to handle sequential data, making them well-suited for processing time-varying EEG signals.
- **Incorporating Adaptive Learning:** Integrating adaptive learning algorithms into the system can personalize the user experience and improve prediction accuracy over time. Adaptive algorithms can dynamically adjust model parameters based on individual user characteristics and performance, leading to a more tailored and responsive system.
- **Exploring Alternative Input Modalities:** While EEG serves as the primary input modality for NeuroScribe, exploring other biosignals, such as electromyography (EMG) or eye-tracking, could expand the system's capabilities and provide additional control signals.
- **Developing a User-Friendly Interface:** Designing a visually appealing and intuitive user interface is crucial for enhancing system usability and accessibility. A well-designed interface can guide users through the calibration, training, and real-time operation of the system.

7. Conclusion

The NeuroScribe project successfully developed a novel BCI-controlled handwriting assistance system, demonstrating the transformative potential of BCI technology in restoring communication and empowering individuals with disabilities. By integrating EEG signal processing, deep learning, and IoT technologies, NeuroScribe enables users to translate their thoughts into written words, fostering greater independence, self-expression, and participation in society. While further research and development are necessary to refine the system's accuracy, speed, and usability, NeuroScribe represents a significant step towards a future where brain-computer interfaces empower individuals with disabilities to communicate and interact with the world in unprecedented ways.

8. Appendix

This Python code implements a deep learning pipeline for EEG signal classification using VMD, Recurrence Plots and a 3D Convolutional Neural Network (CNN).

1. Data Loading and Preprocessing:

- The code starts by loading EEG data from multiple CSV files belonging to different classes (e.g., different letters or mental states).
- The data is then preprocessed:
  - **Normalization:** Each channel of the EEG data is normalized using the mean and standard deviation of a baseline recording. This step is crucial to reduce the impact of inter-individual differences and improve model generalization.
  - **Segmentation:** The normalized EEG signals are segmented into windows of 256 samples.

2. Feature Extraction:

- **Variational Mode Decomposition (VMD):** VMD is applied to each channel of the segmented EEG signals. VMD is a non-recursive signal decomposition technique that decomposes a signal into a set of band-limited modes, each representing a distinct oscillatory component of the original signal. The code specifically selects the mode containing the desired frequency band (determined empirically).
- **Recurrence Plots (RP):** Recurrence plots are generated from the VMD-extracted modes. RPs visualize the recurrence behavior of a time series, capturing the temporal dependencies and recurring patterns within the signal.

3. Data Preparation for CNN:

- The generated recurrence plots are combined into a 5D data structure: (Sample, 1, Channels, RP Width, RP Height).
- Labels are one-hot encoded to represent the different classes.
- The data is split into training, validation, and testing sets.

4. CNN Model Construction and Training:

- A 3D CNN model is defined using the Keras library. The model consists of:
  - Input layer

- Permutation layer to rearrange the dimensions for 3D convolution
    - Multiple convolutional layers (Conv3D) with ReLU activation and max-pooling layers (MaxPool3D)
    - Flatten layer to convert the output of the convolutional layers into a vector
    - Dense layers (fully connected) with ReLU activation
    - Output layer with softmax activation for multi-class classification
- The model is compiled with the Adam optimizer and binary cross-entropy loss function.
- Training is performed using the fit method. Training progress and metrics are recorded in the history object.

5. Model Evaluation and Visualization:

- The trained model is used to make predictions on the test set.
- Performance metrics like accuracy, confusion matrix, and classification report are calculated and visualized.
- Loss and accuracy curves during training and validation are plotted to assess overfitting.

Overall, this code provides a comprehensive pipeline for EEG signal classification, leveraging advanced signal processing techniques (VMD, RP) and deep learning (3D CNN). The code also includes steps for data preprocessing, model evaluation, and visualization, making it a valuable resource for EEG analysis.

Google Collab Link for code:

DL METHOD:

https://colab.research.google.com/drive/1g_Ttr-Qc47_F6yqR5he2veOqxBgHROKX

ML METHOD:

https://colab.research.google.com/drive/1MbhvacThf9QaYsiJXgys0AnFqiug-NRx?usp=sharing

9. References

[1] Wang, Y., Wang, P., & Yu, Y. (2018). Decoding English Alphabet Letters Using EEG Phase Information. Frontiers in Neuroscience,

[2] K. Pratyusha, K. S. Devi and S. Ari, "Motor Imagery based EEG Signal Classification using Multi-scale CNN Architecture," 2022 International Conference on Signal and Information Processing (IConSIP), Pune, India, 2022, pp. 1-5, doi: 10.1109/ICoNSIP49665.2022.10007486.

[3] Agarwal, M., & Sivakumar, R. (2015). THINK: Toward Practical General-Purpose Brain-Computer Communication. In Proceedings of the 2nd International Workshop on Hot Topics in Wireless (pp. 41–45). Association for Computing Machinery.

[4] A. Aldridge et al., "Accessible Electroencephalograms (EEGs): A Comparative Review with OpenBCI's Ultracortex Mark IV Headset," 2019 29th International Conference Radioelektronika (RADIOELEKTRONIKA), Pardubice, Czech Republic, 2019.

[5] Duan, Y., Zhou, J., Wang, Z., Wang, Y.-K., & Lin, C.-T. (2024). DeWave: Discrete EEG Waves Encoding for Brain Dynamics to Text Translation. arXiv preprint arXiv:2309.14030. [cs.HC].

[6] Mohanan, Krishna, Jithin Velayudhan, Neethu Mohan, S. Sachin Kumar, K. P. Soman, and K. Aparna Ajayan. "On the denoising of periodic and aperiodic ECG signals from a discretized reaction diffusion heart model using variational mode decomposition." In AIP Conference Proceedings, vol. 2966, no. 1. AIP Publishing, 2024.

[7] Dragomiretskiy, K., & Zosso, D. (2014). Variational Mode Decomposition. IEEE Transactions on Signal Processing, 62(3), 531-544.

[8] Pabolu, V.K. and Srinivas, K.N.H. Design and Implementation of a three-dimensional CNC Machine, International Journal of Computer Science.

[9] M. Anil Kumar, Dr. J. Krishnaraj, R. Bhanu Gowtham Sai Reddy (2017), mini cnc 2d sketcher for accurate building drawing, International Journal of Civil Engineering and Technology (IJCIET)Volume 8, Issue 6, June 2017, pp. 543–549, Article ID: IJCIET_08_06_060.