5-2024

Optimizing BCI for Real-Time Performance with TensorFlow

Akshay Suresh Babu

Optimizing BCI for Real-Time Performance with TensorFlow

by

Akshay Suresh Babu

A Starred Paper

Submitted to the Graduate faculty of

St. Cloud State University

in Partial Fulfillment of the Requirements

for the Degree of

Master of Science in

Computer Science

May 2024

Starred Paper Committee:
Adriano Cavalcanti, Chairperson
Akalanka Bandara Mailewa
Andrew A. Anda

Abstract

Developed a 1-Dimensional Convolutional Neural Network model and trained to translate electroencephalography (EEG) signals into drone commands. Using a 16-channel EEG headset at a 125 Hz sampling rate raw data was collected and further processed for feature extraction and training of the model to control drone movements using commands, backward, forward, left, right, land, and takeoff based on the classification of brain wave patterns. The developed model was trained on raw EEG dataset of 2,498,750 entries and achieved an accuracy of 99.27%. Though the model performed well in classifying the brain wave patterns, based on commands, it struggled slightly in differentiating 'Takeoff' and 'Forward' commands because of the non-uniform size of the dataset.

Acknowledgements

I would like to thank the faculty and staff of the Department of Computer Science and Information Technology at St. Cloud State University for their immense support and guidance, throughout my academic journey. I am greatly appreciative to Dr. Adriano Cavalcanti for his mentorship, all through the project, and to Dr. Akalanka Mailewa and Dr. Andrew A. Anda, the committee members, for their guidance.

Nevertheless, I would also want to thank my family and friends who have motivated and backed me all this while.

Table of Contents

List of Figures

I. Introduction

Electroencephalography (EEG) [1] technology plays a vital role in healthcare diagnosis, particularly in areas such as seizure diagnosis and patient care [2] and is being used on a growing basis for machine interface. The brain's electrical activity is classified into different types of brain waves based on frequency. Delta waves are associated with deep sleep and theta waves with relaxation and light sleep. Alpha waves are associated with lucid, relaxed states of consciousness and beta waves with active thought and engagement. The gamma waves are associated with awareness and advanced cognitive processes [3]. EEG has been transforming assistive technology by allowing direct contact between the human brain and external equipment. Brain-Computer Interface (BCI) systems [4], using brain signals is transforming how people connect with technology to communicate or carry out activities [5]. Science fiction-like [6] skills are becoming a reality because of this technology, which is helping industries including robotics, healthcare, and entertainment. Though there are various methods and algorithms for processing EEG data [7], the goal of my project is to develop a neural network model based on TensorFlow [8] to classify the brain waves [3] based on action and use it as a command to control the DJI Tello Edu Drone [9]. Through this project, I aim to explore whether neural networks outperform traditional models in BCI applications due to their superior ability to automatically learn and extract complex patterns from large datasets, enabling more accurate and real-time predictions across multiple classes.

II. Literature Review

A. *EEG and BCI in Medical Industry*

Electroencephalography (EEG) [1] is a type of imaging in which sensors positioned over the scalp capture the electrical activity of the brain. It helps in the diagnosis of neurological diseases like epilepsy and sleep disturbances [2]. Beyond diagnostics, EEG has become more important in Brain-Computer Interface (BCI) technology, enabling direct channels of communication between the brain and external devices. This interface technology improves the quality of life for people with severe physical disabilities by enabling control over prosthetic limbs and other assistive equipment [10], as well as supporting medical rehabilitative procedures such as assisting stroke survivors regain mobility.

B. *Existing Solutions*

The "Avatar" project [6],at Saint Cloud State University is an ongoing effort, involving contributions from multiple students [11]. The OpenBCI headset [12] used in the project captures brainwaves from users which is sent to a server and randomly renamed and dated for privacy [6]. The brain waves data is processed, and predictions (drone commands) are made based on brain wave patterns. These predictions are then transmitted to a client-side application(interface), which interprets them to control a DJI Tello Edu Drone. The client application retrieves the latest prediction and provides as an interface for real-time interaction with the drone, allowing adjustments based on the accuracy of the brainwave-interpreted commands, all while ensuring privacy and compliance with Institutional Review Board (IRB) guidelines. The equipment used as part of the "Avatar" project includes an OpenBCI Headset with a Cyton 16 Channel Board and a programmable DJI Tello Edu Drone. The current model used for classification and prediction is a random forest.

C. Gap Analysis

Though a Random Forests model is being used currently because of its superiority in machine learning, I suspect 1-Dimensional Convolutional Neural Network might perform better in tasks involving the processing of large-scale, high-dimensional data like raw EEG data.1-Dimensional Convolutional Neural Network is suitable for BCI applications because of its ability to analyze data in real-time, identify patterns, and flexibility to adapt and change weights based on unseen challenges [13]. In a few scenarios, I suspect 1-Dimensional Convolutional Neural Network performs better as they use filters to extract important features and identify complicated patterns, resulting in faster predictions when compared to the complicated decision trees utilized by Random Forests.

D. Interdisciplinary Relationships between EEG and BCI

The interdisciplinary approach of using EEG and BCI technologies connects the fields of neuroscience, engineering, and computer science. This integration allows researchers to enhance the accuracy and responsiveness of BCIs and refine the interpretation of EEG signals. In applications requiring precise interpretation of brain signals, such development of assistive devices based on brainwave patterns or equipment for people with motor disabilities, this collaborative approach broadens the capabilities of brain computer interfaces (BCIs) [14]. Advances in computing, particularly in signal processing and machine learning are crucial for BCIs to overcome current limitations and enhance accuracy and usability [15].

III. Methodology

A. *Data collection*

Using an OpenBCI headset [12] and a Cyton 16 Channel board in an isolated environment, raw EEG data was collected from volunteers. Participants were instructed to think about the command while the headset placed on their scalps for collecting data [6]. Based on neural activity, the electric potential difference is recorded, each electrode's analog signal is converted into a digital value with a precision of 24 bits [16]. The participant's privacy was ensured by further anonymizing the raw EEG data which included brain wave patterns associated with specific commands backward, forward, left, right, land, and takeoff through random renaming and dating [6]. The data was gathered for around 10 seconds at a sampling rate of 125 Hz per second for each command.

B. *Data Preprocessing*

The Primary unit of collected raw EEG data is microvolt (μV). After placing the headset on the scalp as shown in Fig. 2, one electrode is designated as reference electrode and negative and positive electric potentials are recorded at the other electrodes based on the neural activity. The raw data consists of data from the ExG (Extracellular Signals) channel. From the dataset sample as shown in Fig. 1, columns ExG Channel 0 to ExG Channel 15 represent electrodes placement on the scalp capturing electrical activity from the brain. Accel Channel 0 to Accel Channel 2 represent the three axes (X, Y, and Z) of accelerometer in the headset, which measures acceleration forces which can be used to detect head movements or orientation.

Sample Index	EXG Channel 0	EXG Channel 1	EXG Channel 2	EXG Channel 3	EXG Channel 4	EXG Channel 5	EXG Channel 6	EXG Channel 7	EXG Channel 8	EXG Channel 9	EXG Channel 10	EXG Channel 11	EXG Channel 12	EXG Channel 13	EXG Channel 14	EXG Channel 15	Accel Channel 0	Accel Channel 1	Accel Channel 2	C
0	16205.1488405	9714.693989	-27941.84869	-41379.94097	-26513.68398	30247.5205358	-187500.0224	-38628.44123	0	0	0	0	0	0	0	0	0	0	0	
224	16205.3500062	9713.7105123	-27946.00611	-41387.36175	-26519.1378	30229.549733	-187500.0224	-38641.517	-16730.48189	-28151.77627	-43491.08559	-50390.44325	-11797.47424	-26412.14	-31586.12181	-40414.61383	0.012	0.507	0.035	
226	16215.2071255	9722.7629688	-27946.58726	-41382.64553	-26519.786	30248.615771	-187500.0224	-38640.84645	-16717.8308	-28137.91819	-43476.22168	-50378.03803	-11787.3042	-26399.9136	-31582.63494	-40407.1707	0.012	0.507	0.035	
228	16220.1468610	9728.395608	-27941.82634	-41383.38314	-26518.71312	30215.378727	-187500.0224	-38648.40134	-16726.7268	-28144.4449	-43480.9379	-50382.75425	-11796.9825	-26411.73767	-31591.10625	-40416.29021	0.012	0.507	0.035	
230	16218.82811	9730.8095968	-27946.38609	-41381.05856	-26518.22138	30228.342739	-187500.0224	-38634.74442	-16722.8823	-28136.17475	-43474.72411	-50371.71249	-11796.44606	-26394.97386	-31584.0431	-40408.69062	0.012	0.507	0.035	
232	16218.6045907	9729.490844	-27935.05376	-41373.45897	-26507.49254	30220.385518	-187500.0224	-38627.41305	-16728.09025	-28135.32539	-43477.98747	-50369.32085	-11800.55878	-26395.86793	-31579.99743	-40401.91804	0.012	0.507	0.035	
234	16217.5317069	9736.3751812	-27937.84773	-41367.82633	-26504.98915	30249.867469	-187500.0224	-38612.34797	-16734.54991	-28133.80547	-43480.46851	-50369.52202	-11803.88919	-26396.47143	-31566.98871	-40395.27957	0.019	0.75950000000	0.0555	
236	16209.8650586	9729.5578991	-27929.01879	-41368.96627	-26499.24475	30255.030722	-187500.0224	-38609.86693	-16733.87936	-28130.49741	-43477.42867	-50368.73971	-11799.62001	-26393.67746	-31562.25014	-40386.54004	0.019	0.75950000000	0.0555	
238	16213.9107243	9731.882481	-27929.30936	-41367.20048	-26496.719	30265.312524	-187500.0224	-38603.47433	-16737.88032	-28137.82878	-43481.07201	-50372.65126	-11800.96111	-26395.44325	-31558.07037	-40387.83644	0.019	0.75950000000	0.0555	
0	8046.62800391	4829.2732035	-14088.03631	-20690.22753	-13201.45437	15761.846692	-93750.01118	-19274.69155	-8362.212284	-14061.52714	-21735.03747	-25179.44129	-5894.825565	-13196.33582	-15777.17999	-20187.78267	0.026	1.01	0.084	
2	16092.2501793	9657.8758547	-28178.64307	-41384.38897	-26405.61329	31551.431900	-187500.0224	-38546.38798	-16605.87092	-28061.07289	-43138.68798	-50233.10932	-11646.28704	-26495.89199	-31502.23571	-40288.34883	0.026	1.01	0.084	
4	16083.3765367	9645.5600435	-28177.50313	-41395.69895	-26409.23427	31541.530077	-187500.0224	-38553.62994	-16610.1848	-28069.14187	-43141.52666	-50241.35712	-11658.28993	-26503.93862	-31501.83338	-40289.35466	0.026	1.01	0.084	
6	16081.94603	9641.1343981	-28184.38747	-41401.48805	-26415.24689	31557.712740	-187500.0224	-38552.98174	-16617.8738	-28081.05535	-43150.82498	-50251.54951	-11661.75445	-26513.90749	-31507.39896	-40298.25065	0.026	1.01	0.084	
8	16110.6680167	9667.4871048	-28161.81221	-41380.72328	-26393.43159	31524.498048	-187500.0224	-38531.21114	-16602.94284	-28073.3887	-43137.70451	-50242.34059	-11654.0878	-26514.33218	-31507.75659	-40290.98633	0.03	1.01	0.088	
10	16107.9411039	9667.531808	-28173.301	-41387.07118	-26403.60163	31557.355112	-187500.0224	-38529.87004	-16586.13432	-28054.52383	-43117.7891	-50220.99468	-11636.89931	-26497.79189	-31488.86937	-40275.22835	0.03	1.01	0.088	
12	16105.4824120	9664.536675	-28171.28935	-41392.16738	-26406.70853	31548.883801	-187500.0224	-38537.62609	-16599.00893	-28065.38678	-43128.69676	-50232.30466	-11652.03144	-26511.69467	-31504.94027	-40288.01355	0.03	1.01	0.088	
14	16110.9362376	9671.7115845	-28171.13288	-41387.71938	-26401.18765	31551.878935	-187500.0224	-38530.20531	-16594.1586	-28059.64238	-43122.01358	-50226.94024	-11645.54943	-26506.68788	-31500.33581	-40284.30316	0.03	1.01	0.088	
16	16105.7059294	9666.101297	-28170.57409	-41388.94873	-26399.84654	31531.047109	-187500.0224	-38539.77186	-16592.37046	-28059.17299	-43121.6336	-50226.69437	-11640.72146	-26505.54794	-31505.96845	-40286.22541	0.03	1.01	0.088	
18	16116.5912290	9676.3160438	-28162.57216	-41373.74954	-26389.76591	31536.702100	-187500.0224	-38526.36081	-16593.37629	-28053.38389	-43123.64526	-50220.72646	-11640.56499	-26500.94348	-31502.07925	-40286.0913	0.024	1.008	0.084	
20	16108.1646213	9667.0847734	-28157.56537	-41375.94001	-26393.94568	31538.065557	-187500.0224	-38535.18975	-16593.97979	-28051.01461	-43125.36635	-50219.9665	-11634.8653	-26491.98043	-31501.29694	-40279.49754	0.024	1.008	0.084	
22	16114.1548888	9667.9788432	-28147.55179	-41374.62126	-26390.70468	31530.376557	-187500.0224	-38534.09452	-16594.96326	-28054.83676	-43126.61804	-50222.58165	-11642.46489	-26495.40025	-31493.31736	-40280.88335	0.024	1.008	0.084	
24	16114.1101853	9668.5152850	-28139.37105	-41372.81077	-26389.90002	31526.844981	-187500.0224	-38533.73689	-16586.60371	-28048.57827	-43118.28084	-50218.53599	-11633.81477	-26482.70446	-31492.87033	-40277.32942	0.024	1.008	0.084	
26	16105.8847434	9662.100334	-28141.11449	-41378.01872	-26392.60458	31529.638949	-187500.0224	-38537.78255	-16589.30827	-28052.06514	-43117.0962	-50219.67592	-11634.41827	-26486.90659	-31484.08609	-40278.15643	0.024	1.008	0.084	
28	16098.5310195	9656.3559360	-28150.34576	-41387.78643	-26397.27609	31531.717661	-187500.0224	-38552.17708	-16597.51136	-28058.86007	-43126.43923	-50227.76726	-11638.88861	-26495.44495	-31495.93252	-40288.77351	0.022	1.008	0.09	
30	16096.1617346	9650.1645028	-28157.76654	-41397.37533	-26400.91943	31520.318272	-187500.0224	-38561.386	-16607.05556	-28071.10883	-43139.85028	-50239.63603	-11641.45906	-26511.62762	-31504.40383	-40303.77153	0.022	1.008	0.09	
32	16101.1461736	9653.4725610	-28160.20288	-41402.85151	-26402.79697	31532.924656	-187500.0224	-38560.11195	-16603.12165	-28067.97958	-43137.25747	-50239.09959	-11640.29677	-26505.97262	-31510.21528	-40303.92799	0.022	1.008	0.09	
34	16100.7438422	9653.964299	-28166.84135	-41407.07599	-26406.01562	31526.330891	-187500.0224	-38558.2344	-16602.22758	-28069.5442	-43137.95038	-50239.81485	-11638.30747	-26513.66162	-31510.08117	-40304.86677	0.022	1.008	0.09	
36	16108.6116562	9663.5755495	-28171.13288	-41399.5658	-26407.46849	31539.451365	-187500.0224	-38550.12072	-16604.97685	-28065.99028	-43138.8221	-50239.12194	-11643.64954	-26509.5489	-31520.92177	-40306.25258	0.022	1.008	0.09	
38	16110.4444993	9665.676614	-28168.56243	-41398.8729	-26409.88248	31523.916903	-187500.0224	-38555.30632	-16596.77376	-28057.67543	-43130.90958	-50229.53304	-11632.40661	-26503.69275	-31516.45142	-40299.68116	0.024	1.006	0.09	
40	16116.3006563	9672.337433	-28160.87343	-41391.51918	-26407.26732	31545.575743	-187500.0224	-38546.34327	-16592.1693	-28055.30614	-43128.65205	-50227.72255	-11646.75643	-26496.27197	-31515.69146	-40296.46251	0.024	1.006	0.09	
42	16118.4687755	9672.2256746	-28154.88316	-41385.4395	-26403.8028	31530.376557	-187500.0224	-38546.29857	-16588.81653	-28055.32849	-43121.36538	-50223.31926	-11639.13448	-26500.58585	-31505.49907	-40286.7395	0.024	1.006	0.09	
44	16115.1830691	9670.7951630	-28156.35838	-41375.2024	-26403.57928	31556.841022	-187500.0224	-38536.97789	-16585.75435	-28054.36737	-43117.0962	-50219.60887	-11633.88182	-26494.86381	-31503.73328	-40277.73175	0.024	1.006	0.09	
46	16113.1714121	9668.4482298	-28148.40116	-41365.79232	-26402.61816	31545.79926	-187500.0224	-38529.73593	-16589.28592	-28056.7143	-43120.65013	-50220.86057	-11636.07229	-26493.88033	-31497.47479	-40270.44508	0.024	1.006	0.09	
48	16111.2491621	9668.3141193	-28153.13973	-41369.458	-26405.36742	31572.442540	-187500.0224	-38528.99832	-16597.44431	-28052.93686	-43123.37704	-50218.98302	-11637.99454	-26481.83274	-31490.74691	-40264.14189	0.026	1.008	0.09	
50	16111.7856039	9669.051727	-28138.63345	-41378.15283	-26396.98552	31548.481470	-187500.0224	-38536.19558	-16597.04198	-28054.8144	-43126.48393	-50223.56513	-11641.39201	-26490.41581	-31491.46217	-40273.55197	0.026	1.008	0.09	

Fig. 1. Sample Dataset.

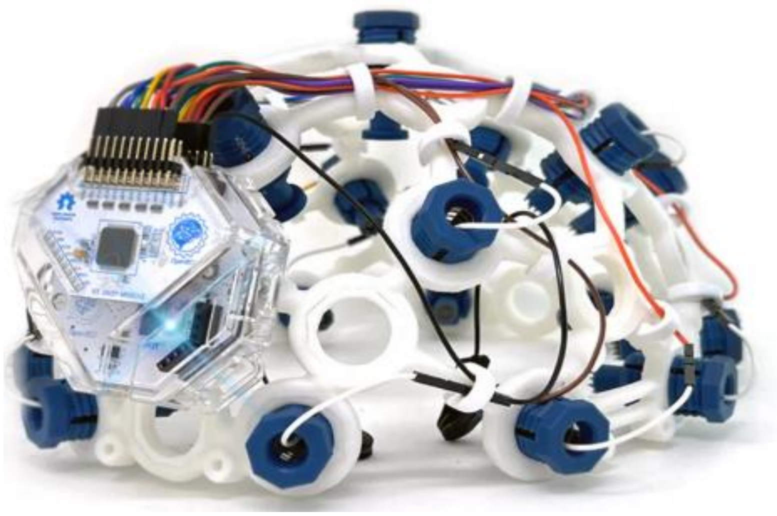

Fig. 2. OpenBCI Headset [12].

C. Feature Engineering

Initially, using python (version 3.11.5), pandas (version: 2.0.3) all individual samples were truncated or padded to a uniform length, resulting in datasets of 1250 rows or 10 seconds of data. Frequency bands for alpha, beta, and gamma waves as shown in Fig. 3 were defined and Butterworth bandpass filter was applied to isolate the EEG data into these bands using Python's signal processing libraries. Then, the data for the isolated alpha, beta, and gamma waves was processed to extract statistical features such as Mean, Standard Deviation, and Skewness and labels [0, 1, 2, 3, 4, 5] were assigned to categorize the actions as [backward, forward, left, right, land, takeoff]. The raw EEG data was compiled based on individual samples and the same labels were applied to raw EEG datasets for model testing purposes. The `StandardScaler` from the sklearn [17] library was used to standardize the features. Label encoding, a technique used to convert categorical variables into numerical format, was used to convert categorical drone command labels into a binary matrix format suitable for neural network processing, using the `to_categorical` function from Keras (library that runs on top of TensorFlow) [18] to transform integer class vectors into one-hot encoded formats for efficient multi-class classification. The `shuffle(df, random_state=42)` function was used to randomly permute the dataset, ensuring that the order of samples does not bias or affect the learning process. The reshaping operation using `X_scaled = X_scaled. reshape(X_scaled.shape[0], 16, 1)` converts the data into a three-dimensional array suitable for 1D convolutional neural networks (CNNs) processing. This structure consists of three dimensions, the first represents the number of samples, the second corresponds to the 16 EEG channels and third, set to 1, represents a single feature per channel specifically, the EEG signal amplitude. This format is crucial as it allows the CNNs to effectively process the temporal and spatial features inherent in the EEG data. The dataset used for training the model consists of 2,498,750 entries.

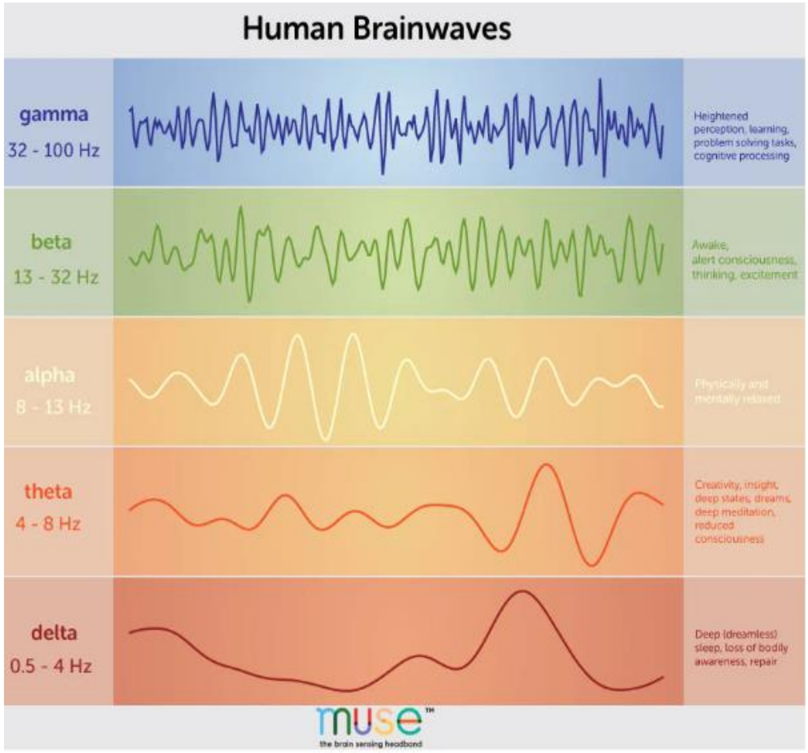

Fig. 3. Brainwave chart [3].

D. Model Development

A neural network model was constructed using a Sequential architecture in Keras as shown in Fig. 4, which includes two 1D convolutional layers [19]with 64 and 128 filters respectively, both followed by batch normalization, a technique used to normalize activations of each layer and max pooling, a down sampling operation that reduces the spatial dimensions of input data to enhance the model's learning efficiency and reduce overfitting, where the model performs poorly on unseen data due to excessive emphasis performance on training data [20]. The model also features a flattening layer to convert the data into a 1D array, a dense layer with 128 units for further processing, a dropout layer which is a regularization technique that randomly removes a portion of neurons during training to avoid overfitting, and a final dense layer with a SoftMax activation function to classify the input into one of six possible drone command categories [20].

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv1d (Conv1D)             (None, 14, 64)            256

 batch_normalization (Batch  (None, 14, 64)            256
 Normalization)

 max_pooling1d (MaxPooling1  (None, 7, 64)             0
 D)

 conv1d_1 (Conv1D)           (None, 5, 128)            24704

 batch_normalization_1 (Bat  (None, 5, 128)            512
 chNormalization)

 max_pooling1d_1 (MaxPoolin  (None, 2, 128)            0
 g1D)

 flatten (Flatten)           (None, 256)               0

 dense (Dense)               (None, 128)               32896

 dropout (Dropout)           (None, 128)               0

 dense_1 (Dense)             (None, 6)                 774

=================================================================
Total params: 59398 (232.02 KB)
Trainable params: 59014 (230.52 KB)
Non-trainable params: 384 (1.50 KB)
_________________________________________________________________
```

Fig. 4. Model Summary.

E. Training Process

The model was compiled using the `Adam optimizer,` an optimization algorithm and `categorical_crossentropy` as the loss function, which quantifies the difference between true labels and predicted class probabilities and helps optimize the neural network for accurate classification. Training involved iterating over 100 epochs with batches of 256 samples, using a validation set, which is a portion of dataset set aside during training to monitor performance. Adam optimizer was used due to its superior performance in handling adapting the learning rate, which controls the size of parameters during training. Categorical crossentropy was used as a loss function, which is standard for multi-class classification problems. It measures the "distance" between the distribution of the predicted probabilities and the true data distribution, effectively quantifying how well the model's predicted probabilities match the actual class labels. Minimizing this loss function during training leads to maximizing the accuracy of the predictions. Early

Stopping callback, a training regularization technique to prevent the model from overfitting was used to halt training if the validation loss does not improve for five consecutive epochs as shown in Fig. 5, at which point the best model weights are restored, ensuring efficiency, and preventing overfitting [20].

```
Epoch 1/100
7809/7809 [==============================] - 218s 28ms/step - loss: 0.0727 - accuracy: 0.9734 - val_loss: 0.0317 - val_accu
racy: 0.9884
Epoch 2/100
7809/7809 [==============================] - 223s 29ms/step - loss: 0.0576 - accuracy: 0.9783 - val_loss: 0.0320 - val_accu
racy: 0.9880
Epoch 3/100
7809/7809 [==============================] - 208s 27ms/step - loss: 0.0498 - accuracy: 0.9813 - val_loss: 0.0386 - val_accu
racy: 0.9857
Epoch 4/100
7809/7809 [==============================] - 207s 26ms/step - loss: 0.0449 - accuracy: 0.9831 - val_loss: 0.0314 - val_accu
racy: 0.9861
Epoch 5/100
7809/7809 [==============================] - 186s 24ms/step - loss: 0.0412 - accuracy: 0.9844 - val_loss: 0.0252 - val_accu
racy: 0.9896
Epoch 6/100
7809/7809 [==============================] - 159s 20ms/step - loss: 0.0388 - accuracy: 0.9853 - val_loss: 0.0236 - val_accu
racy: 0.9904
Epoch 7/100
7809/7809 [==============================] - 158s 20ms/step - loss: 0.0364 - accuracy: 0.9861 - val_loss: 0.0254 - val_accu
racy: 0.9904
Epoch 8/100
7809/7809 [==============================] - 149s 19ms/step - loss: 0.0350 - accuracy: 0.9868 - val_loss: 0.0249 - val_accu
racy: 0.9902
Epoch 9/100
7809/7809 [==============================] - 151s 19ms/step - loss: 0.0336 - accuracy: 0.9871 - val_loss: 0.0245 - val_accu
racy: 0.9906
Epoch 10/100
7809/7809 [==============================] - 162s 21ms/step - loss: 0.0322 - accuracy: 0.9877 - val_loss: 0.0230 - val_accu
racy: 0.9907
Epoch 11/100
7809/7809 [==============================] - 152s 19ms/step - loss: 0.0315 - accuracy: 0.9880 - val_loss: 0.0185 - val_accu
racy: 0.9925
Epoch 12/100
7809/7809 [==============================] - 155s 20ms/step - loss: 0.0305 - accuracy: 0.9883 - val_loss: 0.0210 - val_accu
racy: 0.9913
Epoch 13/100
7809/7809 [==============================] - 139s 18ms/step - loss: 0.0298 - accuracy: 0.9885 - val_loss: 0.0205 - val_accu
racy: 0.9915
Epoch 14/100
7809/7809 [==============================] - 135s 17ms/step - loss: 0.0294 - accuracy: 0.9887 - val_loss: 0.0234 - val_accu
racy: 0.9897
Epoch 15/100
7809/7809 [==============================] - 143s 18ms/step - loss: 0.0285 - accuracy: 0.9890 - val_loss: 0.0193 - val_accu
racy: 0.9921
Epoch 16/100
7809/7809 [==============================] - 138s 18ms/step - loss: 0.0282 - accuracy: 0.9892 - val_loss: 0.0181 - val_accu
racy: 0.9927
Epoch 17/100
7809/7809 [==============================] - 161s 21ms/step - loss: 0.0272 - accuracy: 0.9894 - val_loss: 0.0185 - val_accu
racy: 0.9925
Epoch 18/100
7809/7809 [==============================] - 163s 21ms/step - loss: 0.0274 - accuracy: 0.9893 - val_loss: 0.0197 - val_accu
racy: 0.9921
Epoch 19/100
7809/7809 [==============================] - 162s 21ms/step - loss: 0.0269 - accuracy: 0.9895 - val_loss: 0.0201 - val_accu
racy: 0.9923
Epoch 20/100
7809/7809 [==============================] - 168s 21ms/step - loss: 0.0262 - accuracy: 0.9897 - val_loss: 0.0263 - val_accu
racy: 0.9894
Epoch 21/100
7809/7809 [==============================] - ETA: 0s - loss: 0.0263 - accuracy: 0.9897Restoring model weights from the end
of the best epoch: 16.
7809/7809 [==============================] - 162s 21ms/step - loss: 0.0263 - accuracy: 0.9897 - val_loss: 0.0191 - val_accu
racy: 0.9918
Epoch 21: early stopping
```

Fig. 5. Training.

Early stopping with a patience of 5 halted the training after 21 epochs when there was no improvement to ensure that the model was restored to its optimal state with the lowest validation loss and helped in preventing overfitting.

F. Evaluation Metrics

I evaluated the neural network model using key metrics accuracy, precision, recall, F1-score, and confusion matrix (Figure 6). Accuracy, a measure of overall correctness of the classification model, is a ratio of number of correct predictions to the total number of predictions made. Precision is the ratio of true positives to the total number of positive predictions made. F1-score is the harmonic mean of precision and recall. A confusion matrix is a tabular representation of the number of true positive, true negative, false positive, and false negative predictions made by a classification model [21]. The model achieved an overall accuracy of 99.27% as shown in Fig. 6, demonstrating high reliability in classifying EEG signals into drone commands. The confusion matrix shows high accuracy but also some small errors, especially between the 'Takeoff' and 'Forward' commands, pointing out areas where the model can be tweaked for better results. These metrics prove that the model works well and provides a good foundation for further enhancements and practical applications.

Classification Report:

	precision	recall	f1-score	support
Backward	1.00	1.00	1.00	83277
Forward	0.97	1.00	0.98	87074
Left	1.00	1.00	1.00	82467
Right	0.99	1.00	1.00	82160
Land	1.00	1.00	1.00	82053
Takeoff	1.00	0.97	0.98	82719
accuracy			0.99	499750
macro avg	0.99	0.99	0.99	499750
weighted avg	0.99	0.99	0.99	499750

Fig. 6. Classification Report.

IV. Results

Throughout this project, I observed that the quality of input data and the type of features used to train the model significantly influenced the results and the accuracy of the predictions as shown in Fig. 7.Using statistical features like Mean, Standard Deviation and Skewness for training did not yield satisfactory results, however utilizing labeled raw data in conjunction with a 1-Dimensional convolutional neural network model led to achieving a high accuracy of 99.27%. This highlights the critical role of choosing the right data and model architecture to ensure optimal performance.

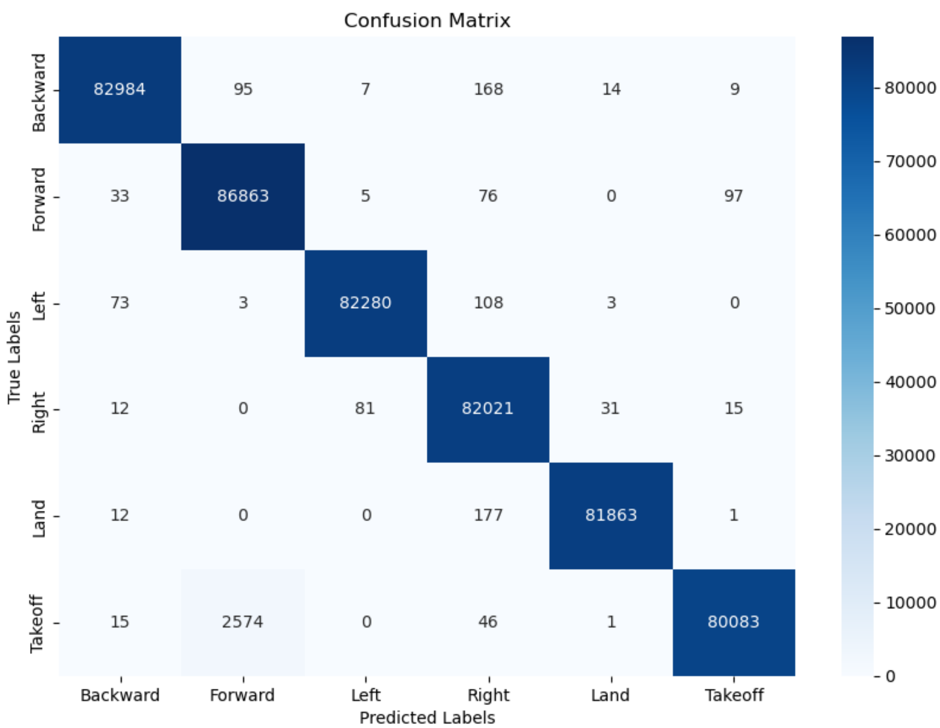

Fig. 7. Confusion Matrix.

V. Future Work

In the future, I plan to improve the model architecture and training procedures to address misclassifications and further enhance overall accuracy. Future efforts will be focused on refining the model architecture and using methods such as bandpass filter, Power Spectral Density (PSD) [13] for raw EEG data processing and feature extraction to optimize the accuracy of the model performance for enhanced real time performance. Having achieved a high accuracy rate with this model, I am now looking forward to deploying it and testing its ability to control a drone based on brain wave patterns.

References

[1] P. D. KATARZYNA BLINOWSKA, ELECTROENCEPHALOGRAPHY (EEG), Wiley encyclopedia of biomedical engineering, 2006.

[2] E. Eberhard and S. R. Beckerman, "Rapid-Response Electroencephalography in Seizure Diagnosis and Patient Care: Lessons From a Community Hospital," *Journal of Neuroscience Nursing,* vol. 55 , no. 5, 2023.

[3] "muse," [Online]. Available: https://choosemuse.com/blogs/news/a-deep-dive-into-brainwaves-brainwave-frequencies-explained-2. [Accessed 8 May 2024].

[4] J. G.-G. L.F. Nicolas-Alonso, "Brain Computer Interfaces, a Review," *Sensors,* vol. 12, no. 2, pp. 1211-1279, 2012.

[5] K. M. N. &. M. Y. Värbu, "Past, Present, and Future of EEG-Based BCI Applications," *Sensors,* vol. 22, no. 9, 2022.

[6] A. Cavalcanti, "Tutorial on TensorFlow Spark," in *Midwest Instruction and Computing Symposium 2023* , Cedar Falls, Iowa 50614, 2023.

[7] J. A. C. Saeid Sanei, EEG signal processing, John Wiley & Sons, 2013 .

[8] "TensorFlow," [Online]. Available: https://www.tensorflow.org/. [Accessed 8 May 2024].

[9] "DJI," [Online]. Available: https://store.dji.com/product/tello-edu. [Accessed 8 May 2024].

[10] D. A. M. H. K. L. Y. P. Y. W. Jetsada Arnil, "BCI-based assistive robot arm," in *7th International Symposium on Medical Information and Communication Technology (ISMICT)*, 2013.

[11] J. D. W. H. M. &. C. A. Knudson, " Kubernetes for High Performance BCI Flying Avatars," in *Minnesota State Symposium*, St. Paul, 2023.

[12] "OpenBCI," [Online]. Available: https://shop.openbci.com/products/the-complete-headset-eeg?variant=44401726161136. [Accessed 9 May 2024].

[13] F. Y. Dewi, A. Faza, P. Prajitno and S. K. Wijaya, " Stroke severity classification based on EEG signals using 1D convolutional neural network," in *Journal of Physics: Conference Series*, 2020.

[14] J. Arnil, D. Anopas, M. Horapong, K. Luangrat, Y. Punsawad and Y. Wongsawat, "BCI-based assistive robot arm," in *7th International Symposium on Medical Information and Communication Technology (ISMICT)*, 2013.

[15] F. S. P. &. V. I. Gembler, "Exploring the possibilities and limitations of multitarget SSVEP-based BCI applications," in *38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 2016.

[16] Á.-M. A. C.-P. D. C.-D. G. C.-D. G. Cardona-Álvarez YN, "A Novel OpenBCI Framework for EEG-Based Neurophysiological Experiment," *Sensors*, vol. 23, no. 7, 2023.

[17] "scikit-learn," [Online]. Available: https://scikit-learn.org/stable/. [Accessed 9 May 2024].

[18] "keras," [Online]. Available: https://keras.io/guides/. [Accessed 9 May 2024].

[19] N. O'Shea, "An Introduction to Convolutional Neural Networks," 2015.

[20] Y. B. a. A. C. I. Goodfellow, Deep Learning, MIT Press, 2016.

[21] H. D. a. H. Dalianis., "Evaluation metrics and evaluation," in *Clinical Text Mining: Secondary Use of Electronic Patient Records*, Springer, Cham, 2018, p. 45 to 53.