Original Article

Paralyzed Patients-oriented Electroencephalogram Signals Processing Using Convolutional Neural Network Through Python

Abstract

Aim: Some of the systems that use brain–computer interfaces (BCIs) that translate brain activity patterns into commands for an interactive application make use of samples produced by motor imagery. This study focuses on processing electroencephalogram (EEG) signals using convolutional neural network (CNN). It is aimed to analyze EEG signals using Python, convert data to spectrogram, and classify them with CNN in this article. **Materials and Methods:** EEG data used were sampled at a sampling frequency of 128 Hz, in the range of 0.5–50 Hz. The EEG file is processed using Python programming language. Spectrogram images of the channels were obtained with the Python YASA library. **Results:** The success of the CNN model applied to dataset was found to be 89.58%. **Conclusion:** EEG signals make it possible to detect diseases using various machine learning methods. Deep learning-based CNN algorithms can also be used for this purpose.

Keywords: *Electroencephalogram, convolutional neural network, spectrogram images*

Vedat Topuz[1], Ayça AK[1], Tülin Boyar[2]

[1]Marmara University, Vocational School of Technical Sciences, Kartal, [2]Marmara University, Faculty of Technology, Department of Computer Engineering, Maltepe, Istanbul, Turkey

Introduction

Analysis of biomedical signals has become one of the hottest topics with advances in technology and machine learning. Researchers are working to understand and classify human biosignals to more accurately diagnose diseases or develop assistive technologies for people with disabilities.[1] Electroencephalogram (EEG) brain signals are one of the most studied topics for developing noninvasive approaches to detect neurological abnormalities and brain–computer interface (BCI) technologies.[2]

Today, motor imagery (MI) EEG-based BCI is studied by many researchers due to its effectiveness in both nonmedical and medical applications. MI is accomplished by imagining executing a particular command without actually doing it. MI tasks commonly used in research are movements related to the left hand, right hand, left foot, right foot, both feet, elbows, fists, and fingers. MI-based BCI applications include clarifying the EEG signals and defining the responses to these signals in real time.[3] It has laid the groundwork for interesting studies such as the processing of EEG signals, robot movements with thought, and detection of various diseases according to brain activity.

The data set, which is frequently used in studies for the diagnosis of diseases related to brain activity, was taken from the University of Bonn database.[4] Türk and Özerdem (2017) aimed to classify the features extracted from EEG signals using the One-Dimensional Median Local Binary Pattern method with the k-nearest neighbor (k-NN) algorithm.[5] The data used in the study were taken from Bonn data set.[4] In the study, the classification performance was found to be 100% for A-E datasets, 99.00% for A-D datasets, 98.00% for D-E datasets, 99.50% for CD-E datasets, and 96.00% for A-D-E datasets. When the effect of the one dimension median local binary pattern method on classification in EEG datasets is compared with other studies, it was seen that the proposed approach gave successful results.[5] Çevik worked on an application that allows vehicle control by detecting the physical and mental states of the operators (such as fatigue, sleepiness, and inattention) through EEG signals for vehicle use. Fast Fourier transform (FFT) and power spectral density (PSD) signal processing techniques are used for feature extraction from

This is an open access journal, and articles are distributed under the terms of the Creative Commons Attribution-NonCommercial-ShareAlike 4.0 License, which allows others to remix, tweak, and build upon the work non-commercially, as long as appropriate credit is given and the new creations are licensed under the identical terms.

For reprints contact: WKHLRPMedknow_reprints@wolterskluwer.com

How to cite this article: Topuz V, Ayça AK, Boyar T. Paralyzed patients-oriented electroencephalogram signals processing using convolutional neural network through python. J Neurobehav Sci 2022;9:90-5.

Ethics committee approval: There is no need for ethics committee approval.

Received : 04-12-2022
Revised : 06-12-2022
Accepted : 11-12-2022
Published : 29-12-2022

Orcid
Vedat Topuz: {ORCID: 0000-0001-7461-1849}
Ayça AK: {ORCID: 0000-0002-3429-4962}
Tülin Boyar: {ORCID: 0000-0002-5797-1284}

Address for correspondence:
Prof. Vedat Topuz, Vocational School of Technical Sciences, Marmara University Mehmet Genç Kulliyesi, Dragos, Kartal, Istanbul 34865, Turkey.
E-mail: vtopuz@marmara.edu.tr

Access this article online

Website: www.jnbsjournal.com

DOI: 10.4103/jnbs.jnbs_33_22

Quick Response Code:

© 2022 The Journal of Neurobehavioral Sciences | Published by Wolters Kluwer - Medknow

signals.[6] Akben conducted a comprehensive research on the characteristics and diagnosis of migraine disease using EEG signals in his study (2012). The data obtained from the KSU Faculty of Medicine used in the study belong to 30 migraine patients and 30 healthy controls. The data were first analyzed with the Fourier transform on the frequency axis. The results were analyzed with the help of classification techniques and clustering techniques. As a result, some characteristic data about migraine disease have been obtained and different suggestions have been made for the automatic diagnosis of the disease.[7] Coşkun and Istanbullu analyzed EEG data from a patient under anesthesia with different methods such as band-pass filter, FFT, wavelet transform, and PSD.[8]

Olivas-Paddal and Chacon-Murguia proposed two methods for multimotor image classification. Both methods use features obtained with a variant of the discriminatory Filter Bank Common Spatial Model. One method uses convolutional neural network (CNN) classification, whereas the second method uses a modular network of four expert CNNs.[9] Hernández-Del-Toro *et al*. present five feature extraction methods based on fractal dimension, wavelet decomposition, frequency energies, empirical mode decomposition, and chaos theory properties to solve the task of detecting MI segments.[10]

In this article, the classification of EEG signals has been carried out by converting them to spectrogram images. It is explained how these operations are performed with Python. A CNN is designed for the classification process. The rest of the article is organized in the following sections. The material and method are described in Section 2. The results obtained are presented in Section 3. In Section 4, a general evaluation of the study is made.

Materials and Methods

There is no need for ethics committee approval.

Electroencephalography (EEG) is a method that provides electrical induction of brain activity. EEG recording is performed by placing conductive wires on the skull with the help of a special gel. The electrical potential changes between these placed electrons are recorded on the computer by means of an analog–digital converter. This recording can be used in studies such as the detection of diseases using various transformations.

Brain activity is related to the frequencies of EEG signals. In clinical studies, the frequencies of EEG signals in the range of 0.5–30 Hz are examined. Frequencies above 30 Hz are known as gamma signals. Because their amplitudes are so low, they are not significant and are rarely used. Table 1 shows the frequency bands and frequency ranges to which EEG signals may belong.

Delta waves are seen in infants and severe organic brain diseases. Theta waves particularly arise in children. In adults, they also occur in situations of emotional tension and frustration. Alpha waves are seen in awake, normal, and calm people. They

Table 1: Electroencephalogram signal bands and frequencies

EEG frequency band	Frequency range (Hz)	Amplitude (μV)
Delta (δ)	0.5-4	20-400
Theta (θ)	48	5-100
Alfa (α)	8-13	2-10
Beta (β)	13-30	1-5
Gama (γ)	30+	1+

+: Plus, EEG: Electroencephalogram

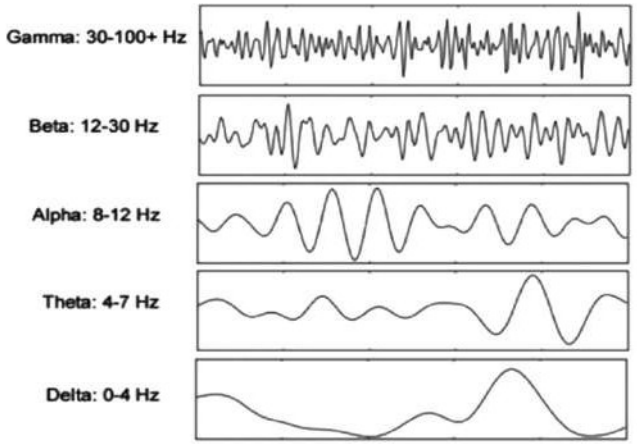

Figure 1: EEG bands.[1] EEG: Electroencephalogram

disappear into the dormant state. If the awake person directs his attention to something special, higher frequency but lower amplitude EEG signals (Beta waves) occur instead of α waves. Beta waves occur in strong activation of the central nervous system or states of tension. They disappear with increased mental activity and are replaced by low-amplitude asynchronous signs. Gamma waves are used a lot in clinical practice when their amplitudes are very small and meaningless. Waves in different frequency ranges of EEG signals are shown in Figure 1.[11]

Data reading and display

Ethics committee report is not required since it has not been tested on paralyzed patients yet. The data used were sampled at a sampling frequency of 128 Hz, in the range of 0.5–50 Hz, belonging to 29 channels. The EEG log file is opened with the Python programming language MNE library [Figure 2].

```
#read files
file = "a.edf"
raw = mne.io.read_raw_edf(file,preload=True)
print(raw)
print(raw.info)
raw.crop(tmax=60)
eeg_and_eog = raw.copy().pick_types(meg=False, eeg=True, eog=True)
print(len(raw.ch_names), '→', len(eeg_and_eog.ch_names))
print(raw.ch_names[-3:])
channel_renaming_dict = {name: name.replace(' ', '_') for name in
raw.ch_names}
raw.rename_channels(channel_renaming_dict)
print(raw.ch_names[-3:])
```

Channel signals can be accessed on the ".edf" file with the Python MNE library. Image of a specific channel or more

than one channel can be taken. The graph in Figure 3 is plotted over Python using the following code; it shows the synchronous signal of all channels.

```
sampling_freq = raw.info['sfreq']
start_stop_seconds = np.array([11, 13])
start_sample, stop_sample = (start_stop_seconds *
sampling_freq).astype(int)
channel_index = 0
raw_selection = raw[channel_index, start_sample:stop_sample]
print(raw_selection)
x = raw_selection[1]
y = raw_selection[0].T
#plot all channels
plt.plot(x,y)
```

The signals of the EEG-A1 and EOG-2 channels are shown in Figure 4 as plotted over Python using the following code.

```
<RawEDF | a.edf, 29 x 64768 (506.0 s), ~14.4 MB, data loaded>
<Info | 7 non-empty values
 bads: []
 ch_names: EEG A1, EEG Fp1, EEG A2, EEG F7, EEG F8, EEG T3, EEG T4, EEG T5, ...
 chs: 29 EEG
 custom_ref_applied: False
 highpass: 0.5 Hz
 lowpass: 50.0 Hz
 meas_date: 2021-10-28 10:49:28 UTC
 nchan: 29
 projs: []
 sfreq: 128.0 Hz
```

Figure 2: Data set information

```
channel_names = ['EEG A1', 'EOG-2']
two_meg_chans = raw[channel_names, start_sample:stop_sample]
y_offset = np.array([5e-11, 0])  # just enough to separate the channel
traces
x = two_meg_chans[1]
y = two_meg_chans[0].T + y_offset
lines = plt.plot(x, y)
#plot two selection
plt.legend(lines, channel_names)
print(raw.info)
#plot one channels fmax=50
raw.plot_psd(fmax=50)
raw.plot(duration=5, n_channels=30)
```

Specific events on the data can be detected with the "events" subcommand. In some data, events are captured from STIM channels.

```
mne.find_events(raw, stim_channel='STI 014')
```

Some EEG/MEG systems create files in which events are stored in a separate data stream rather than pulses on one or more STIM channels. For example, the EEGLAB format stores events as a collection of arrays in the.set file. When reading these files, MNE-Python automatically converts the stored events into an annotations object and stores it as an annotation attribute of the raw object. The underlying data in an annotations object can be accessed through its three attributes: start, duration, and description.

In the EEG data used as an example, the events are stored in a separate data series. To read event information:

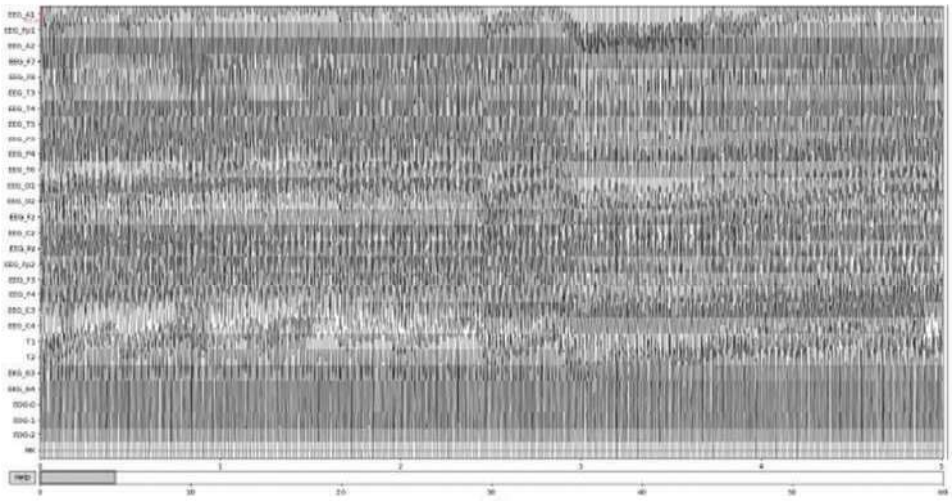

Figure 3: Signals of all channels

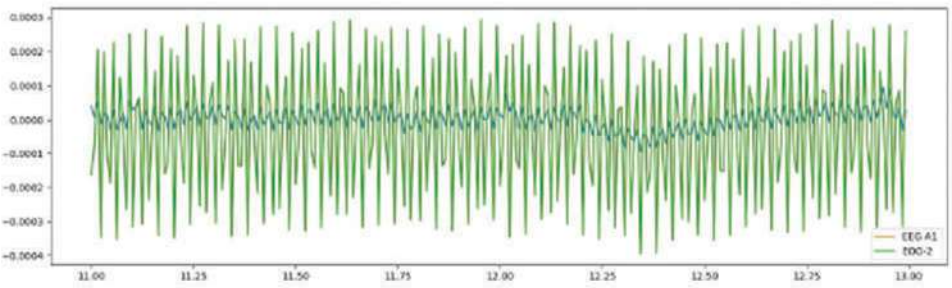

Figure 4: EEG A1 and EOG 2 signals. EEG: Electroencephalogram

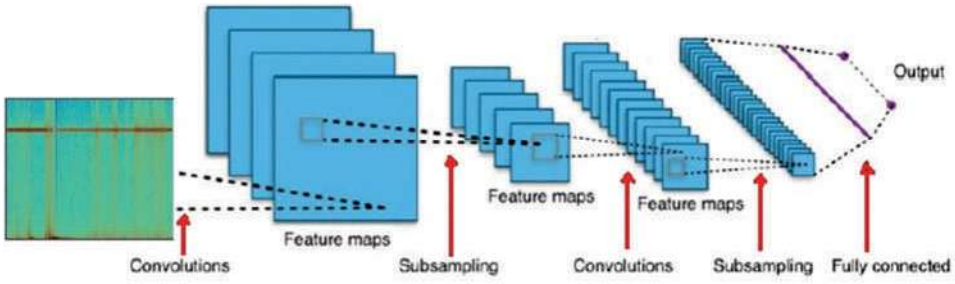

Figure 5: Convolutional neural network

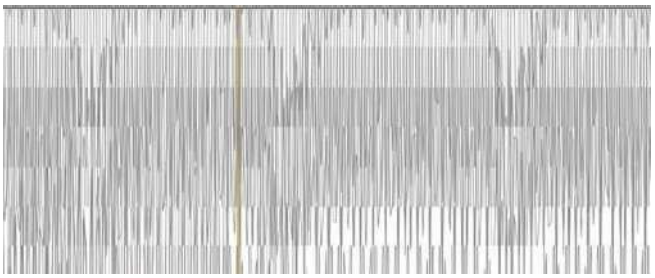

Figure 6: Image of an event on a signal

```
sample_data_raw_file = 'a.edf'
raw = mne.io.read_raw_edf(sample_data_raw_file)
#detect events from annotation
events_from_annot, event_dict = mne.events_from_annotations(raw)

events=mne.events_from_annotations(raw,event_id="auto")
print(events[0])
print(events[1])
```

Convolutional neural network

Recently, studies have been carried out on the use of CNN algorithms, which are a deep learning model on EEG signals[12-14] CNN is a feedforward neural network, which is a type of multilayer perceptron.[15] CNN consists of convolutional layer, pool layer, and fully connected layers. Figure 5 shows the representation of a CNN structure. When the image is given as input to a network, it feeds the system through interconnected polyconvolutions and finally produces an output.[16]

In the convolution layer, there are filters to extract features from the image given as input. Images are saved in matrix format and feature matrix called kernel is used to extract features. Convolution is a mathematical function that expresses how one shape changes by another:

$$|f * g| \equiv \int_0^\tau f(\tau)g(t-\tau)d\tau \quad (1)$$

Convolution operation is performed by shifting the convolution filter on the image matrix. Conflicting numbers are multiplied and transferred to the feature map matrix. With this operation, the size of our input image is reduced. However, this also causes image distortions. To prevent data loss, the same padding method, which is performed by adding a frame of zero values around the input image, is used. Since the feature map holds a certain part of the image, a large number of feature maps are needed. After the convolution process, the ReLu function is applied to reset the negative values.[17]

Results

Here in the "events" array received in the annotation object, events [0]→ event descriptions, events [1]→ event start time, and event identification number (ID) information are kept. To see the events on the signal, the signals received from all channels were plotted and colored during the recording and the event was detected on the signal. The image of the left leg raising and lowering event on the signal are shown in Figure 6. The ID-time graph of the events is shown in Figure 7.

```
sample_data_raw_file = 'a.edf'
raw = mne.io.read_raw_edf(sample_data_raw_file)
raw_temp=raw.copy()
#Dropped last 8 channels
raw_temp.drop_channels(['T1', 'T2', 'EKG 63', 'EKG 64', 'EOG-0', 'EOG-
1', 'EOG-2', 'MK'])
#selection time
raw_selection = raw_temp.copy().crop(tmin=262, tmax=263)
events=mne.events_from_annotations(raw_selection)
events_from_annot, event_dict =
mne.events_from_annotations(raw_selection)
raw_selection.plot(start=0,duration=4)
```

Since the first 21 channels contain information about the EEG data, the other nine channels' information has been removed. Annotation objects detected in the data were obtained with their ids as "events_dict."

Drawing channel spectrograms with YASA Library

Spectrogram images of the channels can be obtained With the Python YASA library. The code block below is for obtaining the spectrogram image of channel 20 [Figure 8].

```
npz = np.load('my_data.npy')
arr=np.array(npz)
arr_trans=arr.transpose()
print(arr_trans)
# fig = yasa.plot_spectrogram(data, sf)
print(len(arr_trans))
data = arr_trans[0,:]
print(data)
#Draw Figure For Example Channel
sf = 100
data = npz[20]
 #draw 1 channel
fig = yasa.plot_spectrogram(data, sf)
print("Figure"+str(20)+":"+str(raw.ch_names[19]))
```

In addition, spectrogram image can be obtained for each event with "plt.specgram" (spectrogram image for *ID = 1* event) [Figure 9].

Obtaining epoch objects with MNE library

The "epochs" object in the library was used to analyze the events in detail to extract the events in the data. After converting the events into epoch objects, operations such as reading the signal values of the events and saving them in the ".csv."

Application of CNN algorithm to electroencephalogram data

First, Python's deep learning libraries Keras and TensorFlow are included for data processing with CNN. In the ".csv" file, the values obtained from 21 channels are input and events are output. The events are numbered from 1 to 6, respectively, by converting them to category type. An example image with a sample signal for each category is presented in Figure 10.

First, 80% of the data set is reserved as training data and 20% as test data:

```
X_train, X_test, Y_train, Y_test = train_test_split(X, Y,
test_size=0.20, random_state=1)test_size=0.20, random_state=1
```

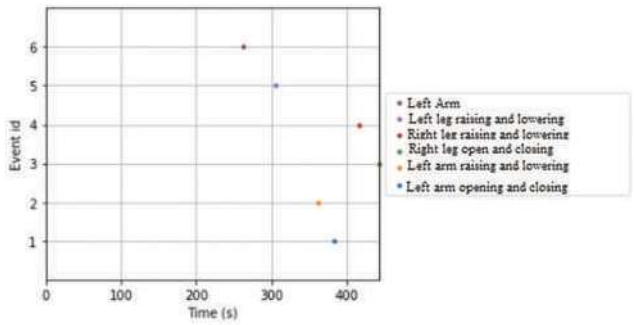

Figure 7: Occurrence time of events

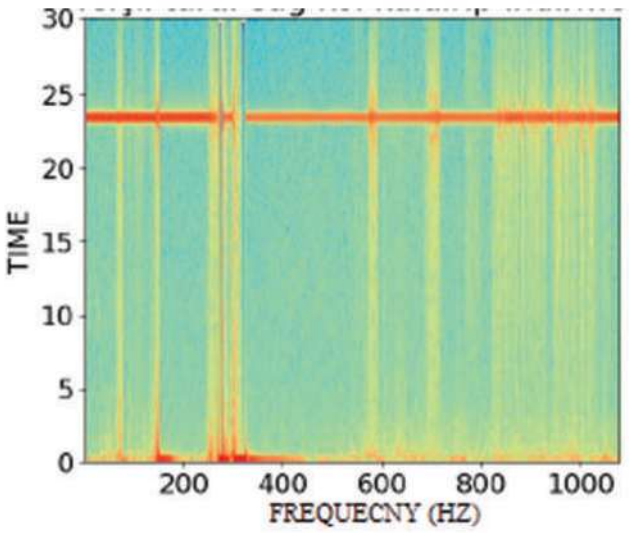

Figure 9: Spectrogram image for ID 1

The CNN structure generated using the following code is shown below.

```
model = Sequential()
model.add(Conv1D(filters=2048, kernel_size=3, activation='relu',
input_shape=(21,1)))
model.add(Dropout(0.5))
model.add(Conv1D(filters=1024, kernel_size=3, activation='relu'))
model.add(Dropout(0.5))
model.add(Conv1D(filters=512, kernel_size=3, activation='relu'))
model.add(Dropout(0.5))
model.add(Conv1D(filters=256, kernel_size=3, activation='relu'))
model.add(Dropout(0.3))
model.add(Conv1D(filters=128, kernel_size=3, activation='relu'))
model.add(Dropout(0.3))
model.add(Conv1D(filters=128, kernel_size=3, activation='relu'))
model.add(Dropout(0.3))
model.add(Conv1D(filters=64, kernel_size=3, activation='relu'))
model.add(Dropout(0.3))
model.add(Conv1D(filters=32, kernel_size=3, activation='relu'))
model.add(Dropout(0.3))
model.add(MaxPooling1D(pool_size=2))
model.add(Flatten())
model.add(Dense(6, activation='softmax'))
model.summary()
```

CNN was trained with the training data set and results were obtained with the test data set. The accuracy value was 89.58%. The data on the success of our model are shown in Figure 11.

Conclusion

In this article, it is aimed to examine EEG signals with Python. For this purpose, first of all, the EEG signal and then the spectrogram of this signal were visualized with Python. A CNN structure was designed to classify the obtained spectrogram image. As a result of the trials with the test data, it was determined which movement the data belonged to with an accuracy of 89.5%. This study can be

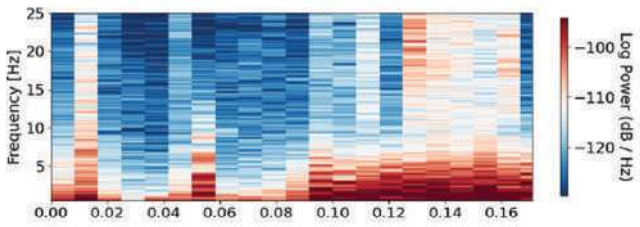

Figure 8: Spectrogram image

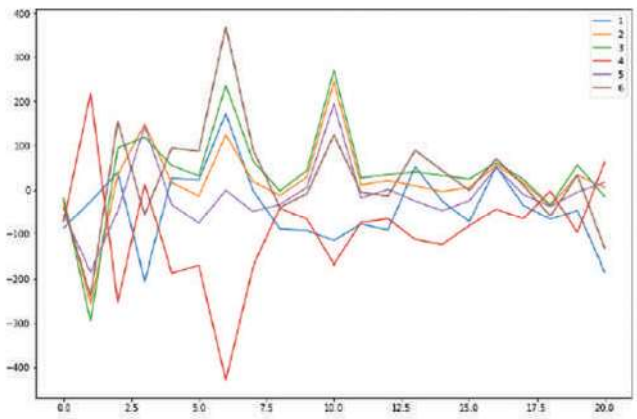

Figure 10: Signals

	precision	recall	f1-score	support
0	1.00	0.83	0.91	454
1	0.57	0.52	0.55	477
2	0.64	0.42	0.51	472
3	0.93	0.76	0.84	422
4	0.80	0.78	0.79	475
micro avg	0.79	0.66	0.72	2300
macro avg	0.79	0.66	0.72	2300
weighted avg	0.78	0.66	0.71	2300
samples avg	0.66	0.66	0.66	2300

accuracy: 89.58%

Figure 11: Accuracy

improved by studies such as device control with data from paralyzed individuals.

Patient informed consent

There is no need for patient informed consent.

Ethics committee approval

There is no need for ethics committee approval.

Conflicts of interest

There are no conflicts of interest to declare.

Financial support and sponsorship

No funding was received.

Author contribution subject and rate

- Vedat TOPUZ (35%): Design the research, Contributed with comments on manuscript organization and write-up.
- Ayça AK (30%): Contribute the research, Analyses and wrote the whole manuscript.
- Tülin BOYAR (35%): Realise the research. Implementing the software.

References

1. Cinar E, Sahin F. New classification techniques for electroencephalogram (EEG) signals and a real-time EEG control of a robot. Neural Comput Appl 2013;22:29-39.
2. Lotte F, Congedo M, Lécuyer A, Lamarche F, Arnaldi B. A review of classification algorithms for EEG-based brain-computer interfaces. J Neural Eng 2007;4:R1-13.
3. Al-Saegh A, Dawwd SA, Abdul-Jabbar JM. Deep learning for motor imagery EEG-based classification: A review. Biomed Signal Process Control 2021;63:102172.
4. Available from: https://www.ukbonn.de/epileptology/donate-to-the-verein-zur-foerderung-der-epilepsieforschung-ev/. [Last accessed on 2022 Nov 28].
5. Türk Ö, Özerdem MS. Epileptik EEG sinyallerinin sınıflandırılması için bir boyutlu medyan yerel ikili örüntü temelli öznitelik çıkarımı. Gazi Üniversitesi Fen Bilimleri Dergisi Part (c) 2017;5:97-107.
6. Çevik, Ç. Vehicle management with attention value obtained using brain signals. J Smart Syst Res (JOINSSR) 2020;1:30-8.
7. AkbenSB. İşaret Işleme Teknikleri Kullanarak EEG Işaretlerinden Migren Hastaliğinin Karakteristiklerinin Belirlenmesi, Thesis, Kahramanmaraş Sütçü İmam Üniversitei Fen Bilimleri Enstitüsü; 2012.
8. Coşkun M, İstanbullu A. Analysis of EEG Signals with FFT and Wavelet Transform, Akademik Bilişim'12- XIV. Academic Informatics Conference Proceedings, Uşak University; 2012.
9. Olivas-Padilla BE, Chacon-Murguia MI. Classification of multiple motor imagery using deep convolutional neural networks and spatial filters. Appl Soft Comput J. 2019;75:461-72.
10. Hernández-Del-Toro T, ReyesGarcia CA, Villasenor-Pineda L. Toward asynchronous EEG-based BCI: Detecting imagined words segments in continuous EEG signals. Biomed Signal Process Control 2021;65:102351.
11. Nacy S, Kbah S. Controlling a servo motor using EEG signals from the primary motor cortex. Am J Biomed Eng 2016;6:139-46.
12. Mao WL, Fathurrahman HI, Lee Y, Chang TW. EEG dataset classification using CNN method. J Phys 2019;1456. [doi: 10.1088/1742-6596/1456/1/012017].
13. Ma M, Cheng Y, Wei X, Chen Z, Zhou Y. Research on Epileptic EEG Recognition Based on Improved Residual Networks of 1-D CNN and indRNN. International Conference on Health Big Data and Artificial Intelligence; 2021.
14. Zhou M, Tian C, Cao R, Wang B, Niu Y, Hu T, *et al*. Epileptic seizure detection based on EEG signals and CNN. Neuroinformatics 2018;12:95.
15. Şeker, A., Diri, B. and Balik, H. H. A review of deep learning methods and applications. Gazi Univ J Eng Sci 2017;3:47-64. [doi: 10.3389/fninf.2018.00095].
16. Available from: https://www.datacamp.com/community/tutorials/convolutional-neural-networks-python. [Last accessed on 2022 Nov 28].
17. Available from: https://mgminsights.com/2021/09/19/convolutional-neural-network-cnn-nedir/. [Last accessed on 2022 Nov 28].