

# Lightweight EEGNet-Based Brain–Computer Interface for Energy-Efficient Edge Computing

Muskan Thukral

*Department of Computer Science and Engineering  
Chandigarh University  
Mohali, Punjab, India  
mthukral55@gmail.com*

Manni Kumar

*Department of Computer Science and Engineering  
Chandigarh University  
Mohali, Punjab, India  
mannikumar55@gmail.com*

Riya Sharma

*Department of Computer Science and Engineering  
Chandigarh University  
Mohali, Punjab, India  
sharmariya16@gmail.com*

Harsh Guleria

*Department of Computer Science and Engineering  
Chandigarh University  
Gharuan, India  
Harshguleria346@gmail.com*

**Abstract**—Brain–Computer Interfaces (BCIs) translate neural activity into control commands, providing an alternative to conventional physical input. However, deploying electroencephalography (EEG)-based BCIs on resource-constrained edge platforms remains challenging because of their limited computational, memory, and energy resources. This paper presents a lightweight BCI framework based on the EEGNet architecture for efficient EEG classification. EEGNet learns temporal and spatial features from multichannel EEG signals using depthwise and separable convolutions while maintaining a small number of trainable parameters. A low-complexity preprocessing pipeline consisting of band-pass filtering, normalization, and windowing is applied before classification. Experiments on a public EEG dataset using subject-wise evaluation show that EEGNet achieves higher mean accuracy and F1-score than the conventional CNN baseline while using substantially fewer parameters. The implemented EEGNet configuration contains 1,972 trainable parameters and requires 0.98 ms of mean CPU inference time per window, compared with 279,679 parameters and 1.65 ms for the baseline. These results demonstrate the potential of lightweight EEG classification for resource-constrained and edge-oriented BCI applications, while direct energy measurements and deployment on dedicated edge hardware remain important directions for future work.

**Index Terms**—Brain–Computer Interface (BCI), EEGNet, Electroencephalography (EEG), Edge Computing, Lightweight Deep Learning, Energy Efficiency, Real-Time EEG Classification, Neural Signal Processing, Resource-Constrained Devices, Low-Latency Computing.

## I. INTRODUCTION

### A. Problem Statement and Motivation

Brain–Computer Interfaces (BCIs) provide a direct communication pathway between the brain and an external device [1]. Among the available sensing modalities, electroencephalography (EEG) is widely adopted because it is non-invasive, low-cost, portable, and permits continuous monitoring of brain activity. EEG-based BCIs are applied in communication assistance, motor rehabilitation, cognitive-state assessment, and human–computer interaction.

Deploying EEG-based BCIs on edge devices is, however, challenging. Conventional deep learning models demand substantial computation, memory, and energy, which exceed the budgets of wearable hardware. Streaming raw EEG to a remote server is likewise undesirable, because the communication overhead increases both latency and energy consumption [2].

Lightweight deep learning models offer a promising solution. EEGNet [3] is a compact convolutional neural network designed for EEG classification, which extracts temporal and spatial features through depthwise and separable convolutions, and its small size suits resource-constrained environments [4]. Nevertheless, selecting a compact model does not by itself guarantee energy-efficient operation, since preprocessing complexity, inference time, memory usage, model size, and computational cost jointly determine the energy budget. This study therefore addresses the design and analysis of a lightweight EEGNet-based BCI optimized for energy-efficient edge computing.

### B. Related Work

Early BCI research focused on decoding brain signals into commands for external devices [1]. Lotte et al. [5] reviewed classification algorithms for EEG-based BCIs and showed that the choice of features and classifier strongly affects system performance. Müller et al. [6] demonstrated the effectiveness of machine learning for single-trial EEG analysis and mental-state monitoring. Signal-processing methods for electrical brain signals were surveyed by Bashashati et al. [7], and Pfurtscheller and Lopes da Silva [8] described event-related synchronization and desynchronization, which underlies the recognition of motor imagery. Lebedev and Nicolelis [9] reviewed the development of brain–machine interfaces and the neural control of external devices.

Classical approaches remain important baselines for new methods. Ang et al. [10] applied the filter bank common spatial pattern algorithm to motor-imagery EEG and showed

that combining spatial and spectral features improves class discrimination. Blankertz et al. [11] showed that a non-invasive BCI can achieve effective performance in untrained subjects.

Deep learning removes the need for manual feature extraction and selection. Schirmer et al. [12] showed that convolutional neural networks learn meaningful features from raw or minimally processed EEG. Lawhern et al. [3] introduced EEGNet, which combines temporal, depthwise, and separable convolutions to extract relevant EEG features with a small network, and which was designed around the characteristics of EEG signals and BCI datasets. Systematic reviews by Roy et al. [13], Craik et al. [14], and Hossain et al. [15] analyze convolutional and other neural architectures for EEG-based tasks, and the survey of Gu et al. [16] covers the pipeline from signal sensing to computational-intelligence methods. Craik et al. [14] also identified inter-subject variability, small datasets, and variations in EEG acquisition as practical difficulties. The theoretical foundations of representation learning are described in [17] and [18].

Reducing the computational cost of neural networks has been widely studied. Chollet [19] showed that depthwise separable convolutions require fewer computations than standard convolutions, and MobileNets [20] and SqueezeNet [21] demonstrated that compact networks can retain useful accuracy. For EEG, Li et al. [4] reviewed lightweight decoding models and stressed the trade-off between model complexity and classification performance, which is critical for wearable and embedded devices with limited processing power, memory, and battery capacity.

Edge computing processes data near its source, which reduces communication requirements and latency, although it introduces constraints on resources and their management [22]. These constraints are particularly relevant to EEG-based BCIs, since EEG is generated continuously. Nguyen et al. [2] surveyed Edge AI-enabled BCIs and reported that local processing reduces continuous data transmission, shortens response time, and improves data privacy, while computational capacity, energy efficiency, and latency remain open problems.

Despite extensive work on EEG classification, lightweight networks, and edge computing, a gap remains between designing a lightweight EEG model and deploying it effectively on an edge device. Most studies emphasize classification accuracy, whereas inference time, memory usage, computational cost, and energy expenditure receive comparatively little attention.

### C. Research Question and Contributions

On the basis of these limitations, this study investigates the following research question: can a lightweight EEGNet-based BCI reduce model complexity and inference latency while maintaining competitive EEG classification performance? To address this question, the study makes the following contributions.

- 1) A lightweight EEGNet-based BCI framework is presented for edge-oriented inference, with preprocessing, feature extraction, and classification organized as a compact local pipeline.

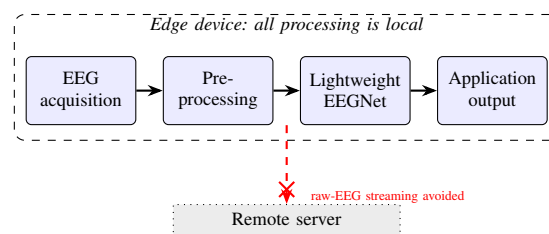


Fig. 1. Overview of the proposed edge-based EEGNet BCI system.

- 2) A computationally efficient pipeline is described that comprises filtering, normalization, segmentation, feature learning, and classification, with EEGNet learning the relevant EEG features automatically while remaining small enough for resource-limited devices.
- 3) The framework is assessed using model size, parameter count, classification accuracy, F1-score, and CPU inference time, providing a resource-aware evaluation beyond classification accuracy alone.
- 4) The feasibility of resource-constrained BCI inference is examined through measured computational complexity and inference latency, while an analytical energy model is used to discuss the conditions under which local inference can be preferable to raw-data transmission.
- 5) The proposed architecture supports applications such as assistive communication, motor-imagery detection, neurorehabilitation, and brain-state monitoring in wearable BCI systems that require low latency, privacy, and energy efficiency.

The remainder of this paper is organized as follows. Section II describes the proposed methodology, Section III defines the evaluation methodology, Section IV presents the results and performance analysis, Section V discusses clinical implications and limitations, and Section VI concludes the paper.

## II. PROPOSED METHODOLOGY

### A. System Architecture Overview

The proposed system is designed for local signal processing and classification on an edge device, as shown in Fig. 1. It consists of four stages: EEG acquisition, preprocessing, lightweight EEGNet classification, and application output. The design objective is to reduce the computational effort of EEG processing without compromising classification accuracy.

EEG signals are first recorded through wearable scalp electrodes and transferred to the edge device, where all subsequent processing is performed locally. This avoids the continuous transmission of EEG data to a server. The resulting brain-state or command label is passed to the target application, such as assistive communication or rehabilitation.

### B. EEG Signal Acquisition and Preprocessing

EEG electrodes measure small voltage fluctuations produced by brain activity. The recorded signal is contaminated by noise from eye movements, muscle activity, electrode displacement,

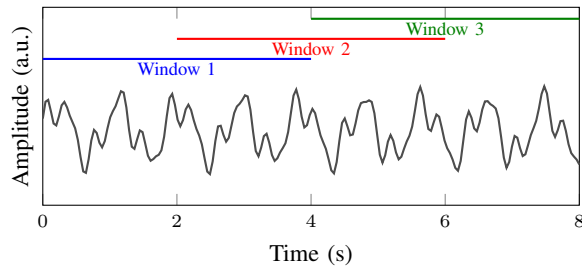


Fig. 2. Overlapping window segmentation of the EEG stream.

and electrical interference, so preprocessing is required before the signal is passed to the neural network.

The proposed workflow applies a band-pass filter to retain the relevant frequency components and suppress out-of-band noise. Each channel is then normalized to stabilize the input amplitude, and the continuous stream is divided into overlapping windows that are processed independently on the device, as illustrated in Fig. 2. Denoting the raw signal of channel  $c$  by  $x_c[n]$ , the three operations are

$$\tilde{x}_c[n] = \sum_{k=0}^{L-1} h[k] x_c[n-k], \quad (1)$$

$$\hat{x}_c[n] = \frac{\tilde{x}_c[n] - \mu_c}{\sigma_c + \epsilon}, \quad (2)$$

$$N_w = \left\lceil \frac{N_{\text{tot}} - T_w}{S} \right\rceil + 1, \quad (3)$$

where  $h[k]$  is a band-pass FIR filter of length  $L$ ,  $\mu_c$  and  $\sigma_c$  are the mean and standard deviation of channel  $c$ ,  $\epsilon$  is a small constant for numerical stability,  $N_{\text{tot}}$  is the total number of samples,  $T_w$  is the window length,  $S$  is the stride, and  $N_w$  is the number of windows. These operations are computationally light, which is essential because they run continuously on the edge device and directly affect its energy consumption.

Fig. 2 shows an illustrative synthetic signal segmented with a window length  $T_w = 4$  s and a stride  $S = 2$  s, which yields a 50% overlap between consecutive windows.

### C. Lightweight EEGNet Architecture

EEGNet [3] is selected as the classifier because its parameter count is considerably smaller than that of general-purpose deep networks. It is designed specifically for EEG and uses temporal and depthwise convolutions to extract features from multichannel input. The architecture is shown in Fig. 3, and its layer configuration is listed in Table I.

The first convolutional layer extracts temporal features from each EEG channel. The depthwise convolution then learns spatial filters across channels, and the separable convolution combines the resulting feature maps using far fewer parameters than a standard convolution. The remaining layers perform batch normalization, nonlinear activation, pooling, and dropout.

Table I uses the default hyper-parameters  $F_1 = 8$ ,  $D = 2$ , and  $F_2 = F_1 D = 16$ , where  $C$  is the number of channels,

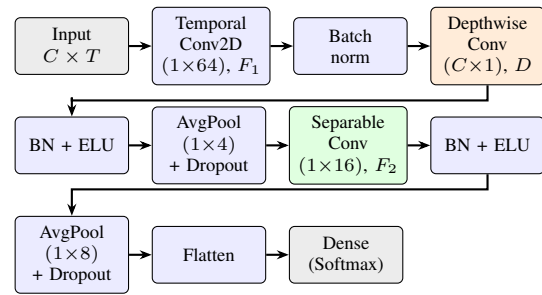


Fig. 3. Block diagram of the lightweight EEGNet model.

 TABLE I  
EEGNET LAYER CONFIGURATION AND PARAMETER COUNTS.

| Layer             | Kernel        | Output                     | Parameters           |
|-------------------|---------------|----------------------------|----------------------|
| Temporal Conv2D   | $1 \times 64$ | $F_1 \times C \times T$    | $64F_1$              |
| Batch norm        | –             | $F_1 \times C \times T$    | $2F_1$               |
| Depthwise Conv2D  | $C \times 1$  | $DF_1 \times 1 \times T$   | $CDF_1$              |
| Batch norm + ELU  | –             | $DF_1 \times 1 \times T$   | $2DF_1$              |
| AvgPool + Dropout | $1 \times 4$  | $DF_1 \times 1 \times T/4$ | 0                    |
| Separable Conv2D  | $1 \times 16$ | $F_2 \times 1 \times T/4$  | $16DF_1 + DF_1 F_2$  |
| Batch norm + ELU  | –             | $F_2 \times 1 \times T/4$  | $2F_2$               |
| AvgPool + Dropout | $1 \times 8$  | $F_2 \times 1 \times T/32$ | 0                    |
| Dense (Softmax)   | –             | $N_c$                      | $N_c F_2 T/32 + N_c$ |

$T$  the number of time samples, and  $N_c$  the number of classes. The temporal, depthwise, and classification stages are described by

$$y_f[n] = \sum_{k=0}^{K_t-1} w_f[k] \hat{x}[n-k], \quad (4)$$

$$z_d[n] = \sum_{c=1}^C v_{d,c} y_d[c, n], \quad (5)$$

$$\hat{p}_j = \frac{e^{s_j}}{\sum_{i=1}^{N_c} e^{s_i}}, \quad (6)$$

where  $w_f$  are the temporal kernels of length  $K_t$ ,  $v_{d,c}$  are the spatial depthwise weights,  $s_j$  are the class scores, and  $\hat{p}_j$  is the predicted probability of class  $j$ . The entire model resides on the edge device, so no EEG data need to be transferred to an external server for classification.

The compactness of EEGNet follows from its use of depthwise separable convolutions. For a convolution with a  $K \times K$  kernel,  $M$  input channels, and  $N$  output channels, the parameter counts of a standard layer and of a depthwise separable layer are

$$P_{\text{std}} = K^2 MN, \quad (7)$$

$$P_{\text{sep}} = K^2 M + MN, \quad (8)$$

$$\frac{P_{\text{sep}}}{P_{\text{std}}} = \frac{1}{N} + \frac{1}{K^2}. \quad (9)$$

The implications of (7)–(9) for the proposed model are quantified in Section IV-A.

#### D. Energy-Efficient Edge Inference

Energy efficiency is a primary design criterion because EEG monitoring may last for extended periods, and repeated network inference consumes processor and battery resources. The small parameter count and computational load of EEGNet reduce this consumption. In addition, the device performs classification only once per window, so no unnecessary high-rate processing is required.

The energy per window and the energy over  $N_w$  windows are modeled as

$$E_{\text{inf}} = P_{\text{avg}} t_{\text{inf}}, \quad (10)$$

$$E_{\text{edge}} = N_w (E_{\text{pre}} + E_{\text{inf}}), \quad (11)$$

$$E_{\text{cloud}} = N_w (E_{\text{pre}} + E_{\text{tx}}), \quad E_{\text{tx}} = e_b B_w, \quad (12)$$

where  $P_{\text{avg}}$  is the average device power,  $t_{\text{inf}}$  is the inference time,  $E_{\text{pre}}$  is the preprocessing energy,  $e_b$  is the radio energy per bit, and  $B_w$  is the number of bits per window. Equation (12) contrasts local inference with transmitting the raw window for remote classification, and it is evaluated in Section IV-B.

#### E. Real-Time BCI Output and Application Integration

After a window is processed by EEGNet, the predicted class is mapped to a BCI command. Depending on the dataset, the output may represent a motor-imagery state, a cognitive condition, or a user intention. Because processing is local, the decoded output can be delivered to an assistive communication device, a rehabilitation system, or a smart device with low latency. The overall design seeks a favorable trade-off between classification performance and computational efficiency by using a compact network instead of a computationally intensive one.

### III. EVALUATION METHODOLOGY

The proposed EEGNet-based BCI is evaluated on the public EEG dataset used in the supplied experimental implementation. The evaluation follows a subject-wise protocol in which the first session is used for training and the second session is used for testing. The implementation evaluates nine subjects and uses three random seeds, producing 27 subject-seed runs. EEG windows are generated from the preprocessed recordings and supplied to EEGNet and the conventional CNN baseline under the same evaluation protocol. The reported computational benchmark is a CPU inference measurement; a named physical edge device and direct power measurement are not part of the supplied experiment.

Classification performance is quantified by accuracy, precision, recall, and F1-score. With  $TP$ ,  $TN$ ,  $FP$ , and  $FN$  denoting true positives, true negatives, false positives, and false negatives, these metrics are

$$\text{Acc} = \frac{TP + TN}{TP + TN + FP + FN}, \quad \text{Prec} = \frac{TP}{TP + FP}, \quad (13)$$

$$\text{Rec} = \frac{TP}{TP + FN}, \quad F_1 = \frac{2 \text{Prec} \cdot \text{Rec}}{\text{Prec} + \text{Rec}}. \quad (14)$$

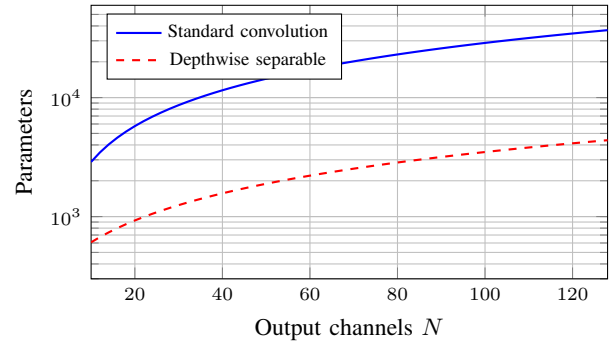


Fig. 4. Parameter growth of standard and separable convolutions.

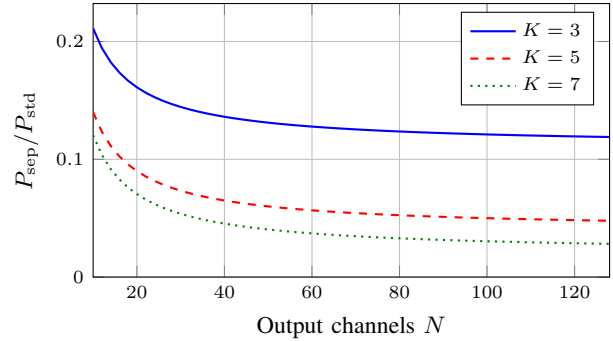


Fig. 5. Parameter ratio versus output channels for three kernel sizes.

Resource efficiency is quantified primarily through parameter count, model storage, and inference time. Classification performance is compared with that of the conventional CNN baseline. Energy is treated analytically rather than experimentally because no direct power measurement is supplied in the current experiment. This distinction is maintained in Section IV to avoid interpreting theoretical energy estimates as hardware measurements.

### IV. RESULTS AND PERFORMANCE ANALYSIS

#### A. Parameter and Memory Analysis

Fig. 4 compares the parameter growth of a standard and a depthwise separable convolution computed from (7) and (8) for  $K = 3$  and  $M = 32$ . The standard layer grows as  $K^2 MN$ , whereas the separable layer grows with a much smaller slope because the spatial and channel-mixing operations are factorized. Fig. 5 shows the ratio in (9), which converges to  $1/K^2$  as  $N$  increases. For  $K = 3$  and  $N = 64$  the ratio is approximately 0.127, corresponding to a reduction of about  $7.9\times$  in the parameter count.

For the implemented experimental network, the parameter count is obtained directly from the trained model. EEGNet contains 1972 trainable parameters, corresponding to approximately 7.7 kB when stored as 32-bit floating-point values. The conventional CNN baseline contains 279 679 trainable parameters, approximately 1.07 MB at the same precision. Thus, EEGNet uses about  $141.8\times$  fewer parameters, corresponding to approximately 99.3% parameter reduction. An

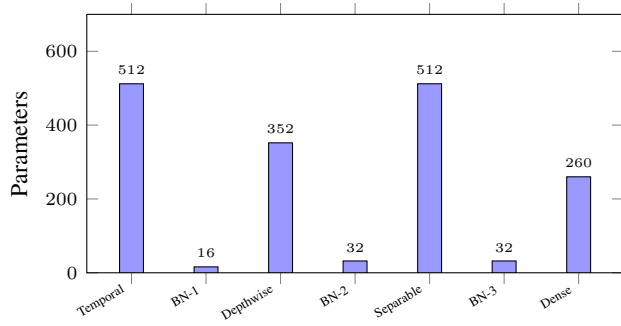


Fig. 6. Per-layer parameter count of the EEGNet model.

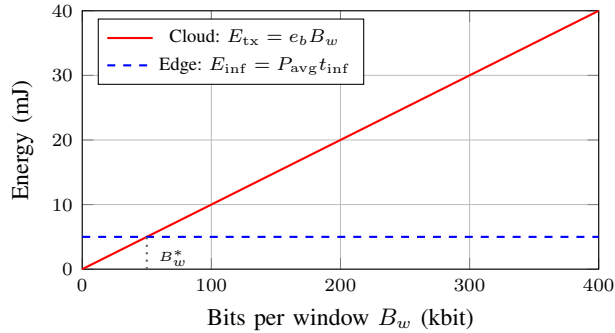


Fig. 7. Per-window energy of edge inference and raw-data transmission.

8-bit storage estimate would reduce the parameter-storage requirement by approximately four times, but no quantized model was evaluated experimentally in the current study.

### B. Energy Analysis

Fig. 7 compares the per-window energy of local inference, from (10), with that of transmitting the raw window, from (12). The preprocessing energy  $E_{pre}$  is common to both strategies and is omitted. The curves use the illustrative values  $P_{avg} = 0.5$  W,  $t_{inf} = 10$  ms, and  $e_b = 100$  nJ/bit; these values are analytical assumptions rather than measurements from the experimental platform. Local inference therefore has an illustrative cost of 5 mJ per window, whereas transmission cost grows linearly with payload. The break-even payload is  $B_w^* = P_{avg} t_{inf} / e_b = 50$  kbit under these assumptions. For example, a 4 s window of 22 channels sampled at 250 Hz with 16-bit resolution contains 352 kbit, for which the illustrative transmission cost is about 35 mJ. This analysis demonstrates the dependency of the edge-versus-transmission trade-off on payload size, device power, inference time, and radio characteristics; it should not be interpreted as a direct energy measurement.

### C. Comparative Analysis with Conventional Architectures

Table II summarizes the qualitative differences between a cloud-based deep CNN pipeline and the proposed edge EEGNet pipeline. Local processing removes the need for continuous raw-EEG transmission, which reduces communication cost and latency, and the smaller network lowers the

TABLE II  
QUALITATIVE COMPARISON OF PROCESSING STRATEGIES.

| Aspect               | Cloud-based deep CNN   | Proposed edge EEGNet |
|----------------------|------------------------|----------------------|
| Processing site      | Remote server          | On-device            |
| Raw-EEG transmission | Continuous             | Not required         |
| Latency              | Network + inference    | Inference only       |
| Data privacy         | Raw data leaves device | Data stays local     |

TABLE III  
EXPERIMENTAL COMPARISON OF EEGNET AND THE CONVENTIONAL CNN BASELINE.

| Metric                  | EEGNet          | CNN baseline    |
|-------------------------|-----------------|-----------------|
| Parameters              | 1 972           | 279 679         |
| Accuracy (%)            | $58.7 \pm 17.9$ | $53.6 \pm 18.9$ |
| F1-score (%)            | 57.8            | 52.4            |
| CPU inference time (ms) | 0.98            | 1.65            |

inference latency and power consumption incurred by repeated classification. Keeping the data on the device also provides better protection of sensitive neural information.

The experimental results indicate that the proposed EEGNet achieves a favorable accuracy–complexity trade-off: it improves mean accuracy and F1-score over the baseline while using substantially fewer parameters and lower measured CPU inference latency. The energy analysis provides a theoretical rationale for local processing, but direct hardware power measurements are required before an energy advantage can be claimed experimentally.

## V. DISCUSSION

### A. Clinical Implications

The proposed lightweight edge-oriented BCI is relevant wherever continuous brain-signal classification is required, including assistive communication, motor rehabilitation, cognitive monitoring, and control interfaces for people with motor impairments. Local analysis avoids the continuous transmission of raw neural signals to a remote server, which reduces communication delay and improves data security. Clinical use nevertheless requires reliable performance across users and conditions, so validation on diverse EEG datasets and with real users is essential.

### B. Limitations and Open Challenges

Several limitations should be considered. First, EEGNet performance varies between users because of inter-subject differences in EEG. Second, performance may degrade in the presence of strong motion artifacts and environmental noise. Third, the present computational benchmark is a CPU measurement and does not by itself establish performance on a particular edge device. Fourth, direct power, energy-per-inference, and runtime-memory measurements were not supplied, so the energy discussion remains analytical. Finally, the evaluation uses a limited number of subjects and sessions, and broader cross-subject and cross-dataset validation is required. Future work should evaluate the framework on

several named edge devices and measure inference time, memory, average power, and energy per window under realistic conditions. Quantization and pruning may further reduce the computational complexity of EEGNet.

## VI. CONCLUSION AND FUTURE WORK

This paper presented a lightweight EEGNet-based BCI for resource-constrained and edge-oriented inference. In the supplied experiment, EEGNet achieved 58.7% mean accuracy and 57.8% F1-score, compared with 53.6% accuracy and 52.4% F1-score for the conventional CNN baseline. EEGNet used 1972 parameters versus 279 679 for the baseline, representing approximately 99.3% fewer parameters, and its measured CPU inference time was 0.98 ms versus 1.65 ms. These results demonstrate a favorable accuracy–complexity trade-off. The analytical energy model further indicates that local inference can become preferable to raw-data transmission above a payload-dependent break-even point, although this conclusion requires direct hardware power measurements for experimental confirmation.

The central objective is a practical compromise between classification accuracy and resource utilization. Future work will evaluate the model on various edge devices and compare it with conventional CNN-based BCI models in terms of classification accuracy, inference time, runtime memory, average power, and energy per inference. Further improvements may include quantization, pruning, and hardware-specific optimization.

## REFERENCES

- [1] J. R. Wolpaw, N. Birbaumer, D. J. McFarland, G. Pfurtscheller, and T. M. Vaughan, "Brain-computer interfaces for communication and control," *Clinical Neurophysiology*, vol. 113, no. 6, pp. 767–791, 2002.
- [2] M.-D. Nguyen, T. Do, X.-T. Tran, Q.-T. Nguyen, and C.-T. Lin, "Edge AI-brain-computer interfaces system: A survey," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 33, pp. 4051–4066, 2025, doi: 10.1109/TNSRE.2025.3618688.
- [3] V. J. Lawhern, A. J. Solon, N. R. Waytowich, S. M. Gordon, C. P. Hung, and B. J. Lance, "EEGNet: A compact convolutional neural network for EEG-based brain-computer interfaces," *J. Neural Eng.*, vol. 15, no. 5, Art. no. 056013, 2018.
- [4] Y. Li, E. Chen, X. Xiao, M. Xu, and D. Ming, "Lightweight deep learning models for EEG decoding: A review," *J. Neural Eng.*, vol. 22, no. 6, Art. no. 061004, 2025, doi: 10.1088/1741-2552/ae2717.
- [5] F. Lotte, M. Congedo, A. Lécuyer, F. Lamarche, and B. Arnaldi, "A review of classification algorithms for EEG-based brain-computer interfaces," *J. Neural Eng.*, vol. 4, no. 2, pp. R1–R13, 2007.
- [6] K.-R. Müller, M. Tangermann, G. Dornhege, M. Krauledat, G. Curio, and B. Blankertz, "Machine learning for real-time single-trial EEG-analysis: From brain-computer interfacing to mental state monitoring," *J. Neurosci. Methods*, vol. 167, no. 1, pp. 82–90, 2008.
- [7] A. Bashashati, M. Fatourehchi, R. K. Ward, and G. E. Birch, "A survey of signal processing algorithms in brain-computer interfaces based on electrical brain signals," *J. Neural Eng.*, vol. 4, no. 2, pp. R32–R57, 2007.
- [8] G. Pfurtscheller and F. H. Lopes da Silva, "Event-related EEG/MEG synchronization and desynchronization: Basic principles," *Clinical Neurophysiology*, vol. 110, no. 11, pp. 1842–1857, 1999.
- [9] M. A. Lebedev and M. A. L. Nicolelis, "Brain-machine interfaces: Past, present and future," *Trends Neurosci.*, vol. 29, no. 9, pp. 536–546, 2006.
- [10] K. K. Ang, Z. Y. Chin, C. Wang, C. Guan, and H. Zhang, "Filter bank common spatial pattern algorithm on BCI competition IV datasets 2a and 2b," *Front. Neurosci.*, vol. 6, Art. no. 39, 2012, doi: 10.3389/fnins.2012.00039.
- [11] B. Blankertz, G. Dornhege, M. Krauledat, K.-R. Müller, and G. Curio, "The non-invasive Berlin brain-computer interface: Fast acquisition of effective performance in untrained subjects," *NeuroImage*, vol. 37, no. 2, pp. 539–550, 2007.
- [12] R. T. Schirrmester, J. T. Springenberg, L. D. J. Fiederer, M. Glasstetter, K. Eggensperger, M. Tangermann, F. Hutter, W. Burgard, and T. Ball, "Deep learning with convolutional neural networks for EEG decoding and visualization," *Hum. Brain Mapp.*, vol. 38, no. 11, pp. 5391–5420, 2017.
- [13] Y. Roy, H. Banville, I. Albuquerque, A. Gramfort, T. H. Falk, and J. Faubert, "Deep learning-based electroencephalography analysis: A systematic review," *J. Neural Eng.*, vol. 16, no. 5, Art. no. 051001, 2019.
- [14] A. Craik, Y. He, and J. L. Contreras-Vidal, "Deep learning for electroencephalogram (EEG) classification tasks: A review," *J. Neural Eng.*, vol. 16, no. 3, Art. no. 031001, 2019.
- [15] K. M. Hossain, M. A. Islam, S. Hossain, A. Nijholt, and M. A. R. Ahad, "Status of deep learning for EEG-based brain-computer interface applications," *Front. Comput. Neurosci.*, vol. 16, Art. no. 1006763, 2023, doi: 10.3389/fncom.2022.1006763.
- [16] X. Gu, Z. Cao, A. Jolfaei, P. Xu, D. Wu, T.-P. Jung, and C.-T. Lin, "EEG-based brain-computer interfaces (BCIs): A survey of recent studies on signal sensing technologies and computational intelligence approaches and their applications," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 18, no. 5, pp. 1645–1666, 2021, doi: 10.1109/TCBB.2021.3052811.
- [17] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.
- [18] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
- [19] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, 2017, pp. 1251–1258.
- [20] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient convolutional neural networks for mobile vision applications," arXiv:1704.04861, 2017.
- [21] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5 MB model size," arXiv:1602.07360, 2016.
- [22] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE Internet Things J.*, vol. 3, no. 5, pp. 637–646, Oct. 2016, doi: 10.1109/IIOT.2016.2579198.