COSINE CONVOLUTIONAL NEURAL NETWORK AND ITS APPLICATION FOR SEIZURE DETECTION

GUOYANG LIU, LAN TIAN, YIMING WEN, WEIZE YU, WEIDONG ZHOU[1]

School of Integrated Circuits, Shandong University,
Jinan 250100, P.R.China
E-mail: wdzhou@sdu.edu.cn

[1]Corresponding author.

Abstract

Traditional convolutional neural networks (CNNs) often suffer from high memory consumption and redundancy in their kernel representations, leading to overfitting problems and limiting their application in real-time, low-power scenarios such as seizure detection systems. In this work, a novel cosine convolutional neural network (CosCNN), which replaces traditional kernels with the robust cosine kernel modulated by only two learnable factors, is presented, and its effectiveness is validated on the tasks of seizure detection. Meanwhile, based on the cosine lookup table and KL-divergence, an effective post-training quantization algorithm is proposed for CosCNN hardware implementation. With quantization, CosCNN can achieve a nearly 75% reduction in the memory cost with almost no accuracy loss. Moreover, we design a configurable cosine convolution accelerator on Field Programmable Gate Array (FPGA) and deploy the quantized CosCNN on Zedboard, proving the proposed seizure detection system can operate in real-time and low-power scenarios. Extensive experiments and comparisons were conducted using two publicly available epileptic EEG databases, the Bonn database and the CHB-MIT database. The results highlight the performance superiority of the CosCNN over traditional CNNs as well as other seizure detection methods.

Keywords: Cosine convolutional neural network (CosCNN); cosine kernel; network quantization; seizure detection; Field programmable gate array (FPGA); Electroencephalogram (EEG).

1. Introduction

Convolutional Neural Networks (CNNs) are the most widely used deep learning approach developed in the past decade, and have become a mainstream method in the field of computer vision (Khan et al., 2018; Krizhevsky et al., 2012) and natural language processing (Li, 2017). Traditional CNNs contain multiple convolutional layers, where a set of filters is learned in each convolutional layer to obtain discriminative deep-learned features (Goodfellow et al., 2016; LeCun et al., 2015). Unlike the traditional hand-crafted feature engineering approaches that model the natural data without the learning process, CNN is a typical data-driven model that can directly train feature extractors from large data sets using the backpropagation algorithm, which demonstrates superior performance and can extract more inherent features than conventional feature engineering methods. In view of the strengths of CNNs in feature extraction, they have also been extensively investigated and applied to various EEG classification tasks, such as seizure detection (Shoeibi et al., 2021; Thuwajit et al., 2021). However, the performance of traditional CNNs is highly dependent on complex model parameters and expensive training, causing model redundancy and overfitting problems, especially when the training dataset is limited in size (Everitt & Skrondal, 2002). Furthermore, the large number of parameters in deep CNNs will result in significant computational complexity, and the inherent "black-box" nature of learned convolutional kernels inevitably leads to low interpretability (LeCun et al., 2015).

Epilepsy is a common neurological disorder characterized by repetitive, unpredictable, and short-lasting seizure attacks (Fisher et al., 2005). About 1% population around the world is affected by epilepsy (Thijs et al., 2019; World-Health-Organization, 2019), and the life quality of those patients is seriously threatened by various clinical phenomenology of epileptic seizures such as impaired motor control and awareness loss (Elger & Hoppe, 2018). Electroencephalogram (EEG) has been widely used in

epilepsy diagnosis (Acharya et al., 2013) and other Brain-Computer Interface (BCI) tasks (Faraji & Khodabakhshi, 2023; Zhang et al., 2023; Zhang et al., 2021). In recent years, an increasing number of studies have employed deep learning methods for EEG-based automatic seizure detection, where the CNN-based methods, including 1-D CNNs and 2-D CNNs, are the most popular (Shoeibi et al., 2021). However, as ictal EEG cannot be easily acquired in clinical settings, the available training data for optimizing CNNs are usually insufficient. Due to the inherent parameter redundancy in CNNs, CNN-based seizure detection models often result in overfitting issues. Moreover, the kernels in traditional CNNs lack clear physical meanings, leading to poor model robustness, which hinder the clinical application of CNNs in seizure detection.

Typical characteristic waves of epileptic EEG include spike waves, sharp waves, etc., with specific amplitude and periodic components (Latka et al., 2003). To detect these epileptic waveforms, previous studies achieved considerable success in seizure detection by integrating spectral features with CNNs to address the overfitting issues, indicating the significance of frequency and amplitude features in detecting epileptic EEG signals (Shoeibi et al., 2021). For instance, Liu et al. (2020) extracted spectral features from intracranial EEG by S-transform and designed a CNN for seizure detection. Ozdemir et al. (2021) integrated the Fourier-based synchrosqueezing transform (SST) with a CNN for both seizure detection and prediction tasks. Nevertheless, these studies separated the process of spectral feature extraction from the classification process of CNNs, which necessitate laborious feature engineering, and made it challenging to obtain optimal EEG features and realize efficient seizure detection. Inspired by these works, we attempt to explore if it is feasible to embed trigonometric functions with clear amplitude and frequency information into a traditional CNN. More specifically, can we replace the traditional kernels in a CNN with trigonometric functions that require fewer parameters, thereby constructing a novel end-to-end compact CNN model? Such a model would enable direct extraction of spectral features from raw signals for classification. If feasible, the scale of parameters in CNNs can be significantly reduced to make the model more compact. It is also expected to alleviate the overfitting problem of CNNs with the new trigonometric kernels and improve the interpretability and robustness of the deep model.

In this work, we present a novel cosine convolutional neural network (CosCNN) along with its hardware implementation, aiming to promote the model performance and interpretability, to compress the model scale, and enhance the model generalization abilities. In contrast to the traditional CNNs, CosCNN learns a set of cosinusoidal kernels with different amplitudes and frequencies in each convolutional layer. The main contributions of the present work are summarized as follows：

● In order to address limitations of traditional CNNs such as high memory costs, poor interpretability and overfitting, we propose an innovative CNN model, CosCNN, where only two parameters representing amplitude and frequency are needed to learn in each cosine filter, significantly reducing the model size and enhancing the model generalization ability. To the best of our knowledge, this is the first work that proposes to replace traditional convolution kernels with cosine filters.

● The CosCNN has demonstrated its higher classification ability than the traditional 1-D CNN and achieved state-of-the-art performance in EEG-based seizure detection tasks on CHB-MIT epileptic EEG database, indicating its great potential in future applications. The cosine filter can be readily integrated into any existing 1-D CNN architecture to obtain performance improvement.

● A quantization algorithm is presented for the CosCNN to enable the model hardware-friendly. By applying the cosine lookup table and a calibration algorithm based on KL divergence, all weights can be quantized to 8-bit integers with seldom accuracy loss, alleviating the memory access burden. The

proposed quantization algorithm can be used to quantize two parameters of the cosine kernel, rather than the full-length kernel, thus achieving fewer parameters than traditional quantization algorithms.

● A configurable cosine convolution hardware accelerator is designed, and the whole quantized CosCNN is deployed on an energy-efficient Field Programmable Gated Array (FPGA) to realize real-time seizure detection.

The remainder of this paper is organized as follows. Section 2 presents the related works. Section 3 explicitly describes details of the CosCNN and its corresponding quantization method and hardware implementation. Experimental databases and results are elucidated in Section 4. Section 5 discusses experimental results and depicts the result comparison. Finally, conclusions are drawn in Section 6.

2. Related Work

2.1. *Specially-Designed Filters in 1-D CNNs*

In recent years, attempts have been made to improve the interpretability and accuracy of deep learning models. The adequate design of the 1-D convolutional filter is proven to be effective for CNNs to process raw 1-D signals in an explainable way. Here, some mainstream specially-designed filters in 1-D CNNs are summarized.

EEGNet (Lawhern et al., 2018) is a classic compact CNN architecture that is widely used in EEG signal classification. It contains a set of learnable temporal filters and spatial filters, with all the filters being implemented by the traditional 1-D convolutional filter. According to EEGNet, Thuwajit et al. (2021) presented an end-to-end model named EEGWaveNet for ictal EEG classification. Furthermore, Ravanelli and Bengio (2018) proposed the SincNet for speaker recognition, which replaced all filters of the first convolutional layer with filters modulated by the sinc function. Besides, some recent works combined the SincNet and EEGNet to enhance the model interpretability and performance in EEG classification tasks (Borra et al., 2020; Liu et al., 2022). Priyasad et al. (2021) adopted the customized SincNet model to encode each channel of the EEG signal and introduced the attention mechanism for multichannel ictal EEG detection. In addition to EEGNet and SincNet, some other types of specially-designed filters were also exploited and incorporated with CNNs to realize various time-series signal classifications. For example, the Gabor filter (Noé et al., 2020; Zeghidour et al., 2021) and Gammatone filter (Abdoli et al., 2019) were designed for raw audio signal processing, and the wavelet filter (T. Li et al., 2021) was applied to industrial intelligent diagnosis.

The specially-designed 1-D convolutional filters presented in existing studies are usually served as the learnable frontend of 1-D CNNs to process raw signals, which means only the first convolutional layer is replaced by the new filter module. Although filters in the first convolutional layer are explainable and have fewer learnable parameters, the backend of CNNs containing the majority of learnable parameters still uses traditional convolutional filters. Besides, the specially-designed filters usually have longer filter lengths to ensure the performance of filters, which improves the computational complexity of the model. Moreover, most of the filters are designed for audio signal classification, and few works concentrate on ictal EEG classification tasks. Unlike that existing specially designed filters usually only replace the kernels in the frontend of the traditional 1-D CNNs, the proposed cosine kernel can replace all kernels in a traditional 1-D CNN, making the CosCNN more compact and interpretable.

2.2. *CNN-based seizure detection algorithms*

To achieve end-to-end feature extraction, some studies took the time-frequency representation of the EEG signals as the input of 2-D CNNs. Cho and Jang (2020) transformed raw ictal EEG signals into 2-D images using Short Time Fourier Transform (STFT) and classified them by a CNN. Liu et al. (2020) sent the S-transform representation of multi-channel EEG signals into a CNN with four convolutional layers for seizure detection. Ozdemir et al. (2021) employed SST to obtain discriminative image-based features and proved that it was a better feature representation than STFT when combined with a CNN. However, the time-frequency transform is time-consuming, and the high-resolution time-frequency image will increase the computational complexity of CNNs. For more efficient EEG classification, 1-D CNNs that fed with raw EEG signals were developed. Acharya et al. (2018) constructed a 1-D CNN architecture with 13 convolutional layers for the three-class classification of single-channel ictal EEG. O'Shea et al. (2020) proposed a fully convolutional network with 1-D convolution kernels to detect seizures in neonates. Wang et al. (2021) presented a stacked 1-D CNN model consisting of two parallel CNN blocks with different kernel lengths, which yielded promising results on two long-term multi-channel ictal EEG databases. In addition, combining the 1-D CNN with recurrent neural networks such as Long-Short Term Memory (LSTM) is another widely explored deep learning architecture in seizure detection (Li et al., 2020; Liu et al., 2021). At present, most CNN models designed for seizure detection managed to improve the performance and generalization ability by modifying network architecture, loss function, input feature, etc. The optimization of convolutional filters according to the intrinsic characteristics of EEG signals was seldom considered. The presented study enhances the performance of the 1-D CNN by modifying the basic convolutional filter, which is fundamentally different from existing CNN-based seizure detection studies.

2.3. *CNN Quantization Methods*

In general, the deep-learning model is trained and inferred with the 32-bit floating-point (FP32) data to meet the needs of computing precision. However, significant memory and computational complexity are required for FP32 convolution, making the model hard to be deployed on hardware platforms such as FPGAs. Although some quantization-aware training methods (Jacob et al., 2018; Liang et al., 2021) were proposed to quantize the CNN parameter to low bit-width and achieve excellent performance, they required time-consuming retraining or fine-tuning. As a more practical quantization approach, the Post-Training Quantization (PTQ) method has recently attracted much attention (Nagel et al., 2020; Nahshan et al., 2021). It only needs some unlabeled data for calibration, rather than retraining the model. A representative PTQ method presented by Migacz (2017) leveraged the Kullback-Leibler (KL) divergence to determine the optimal activation threshold layer by layer. It could quantize all weights and intermediate activations of the popular CNN model to 8-bit integers with minimal accuracy loss. All the mentioned quantization methods are designed for CNNs with traditional filters, and the effective quantization algorithm for CNNs with specially-designed filters is seldom investigated. In this work, a KL divergence-based CosCNN quantization algorithm is developed to significantly reduce the memory occupation of the CosCNN model and make the model hardware-friendly and energy-efficient.

On the other hand, most existing seizure detection algorithms were only offline tested in laboratory environments, and their testing platforms were usually the Central Processing Unit (CPU) and Graphics Processing Unit (GPU), which were not energy-efficient. However, the online EEG monitoring system deployed in low-power portable devices is highly demanded for clinical practice and personal daily use

(Kuhlmann et al., 2018). Some traditional machine learning-based seizure detection algorithms have been implemented on low-power hardware platforms such as FPGAs (Feng et al., 2017; Page et al., 2014). At the same time, several studies also explored the feasibility of deploying deep learning-based seizure detection algorithms in FPGAs and other low-power embedded processors by utilizing quantized or fix-point CNNs. Truong et al. (2018) introduced a hardware-friendly CNN with integer inputs and weights for seizure detection. Nevertheless, they only simulated it on the computer rather than deploying it on an energy-efficient platform. Bahr et al. (2021) established a CNN-based seizure detector with four convolutional layers and implemented it on the low-power RISC-V processor. The common limitation of the current hardware-implemented epilepsy diagnosis systems is that the model performance and power consumption cannot be well balanced, especially for deep learning-based systems (Wei et al., 2020).

3. Proposed Method

In this section, we elaborate the details of the proposed CosCNN. The definition of the cosine filter is first introduced. Subsequently, the detailed theoretical derivations of forward and backward propagation are provided for cosine convolution. Following that, the parameter quantization and hardware implementation procedures of the CosCNN are presented.

3.1. *Cosine Filters*

Unlike convolutional filters (kernels) used in traditional CNNs, only two parameters need to be learned in proposed cosine filters, namely the frequency factor ω and the amplitude factor A. Suppose k represents the filter length, then the cosine filter $K_m(A,\omega)$ can be defined as:

$$K_m(A,\omega)=A\cos\left(\omega\left(m-\frac{k-1}{2}\right)\right), m\in\{0,1,...,k-1\}, \tag{1}$$

It can be seen that the $K_m(A,\omega)$ is a centrosymmetric filter, which means that for the proposed cosine filter, the convolution operation is equivalent to the cross-correlation operation. In practice, the factor A and ω in all cosine filters are sampled from Gaussian distribution with zero mean and unit variance at the model initialization phase.

3.2. *Cosine Convolution*

Each convolutional layer in CosCNN contains multiple input channels and output channels. For the ease of understanding, the simplest convolution case where only one channel is sampled from input to output can be considered. Let x_i^l be the i-th output value in l-th layer, A^l and ω^l be the learnable parameters in l-th layer, O^{l-1} be the feature vectors obtained from previous layer. Then the cosine convolution in "valid" mode, which is computed without the zero-padded edge, can be expressed as:

$$\begin{aligned} x_i^l &= \left(K_m\left(A^l,\omega^l\right)*O^{l-1}\right)_i \\ &= \sum_{m=0}^{k-1} A^l\cos\left(\omega^l\left(m-\frac{k-1}{2}\right)\right)O_{i+m}^{l-1}, \end{aligned} \tag{2}$$

where $*$ represents the convolution operator. This is a memory-efficient implementation of cosine convolution since only two parameters are required for generating a cosine convolutional kernel. However, compared to traditional convolution, it results in a slight increase in computational complexity due to calculations of the cosine function. Therefore, in scenarios with limited computational resources, it

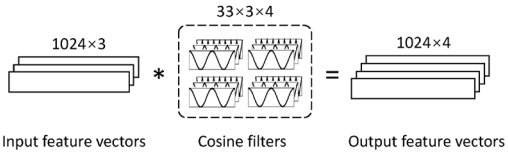


Fig. 1. The schematic of convoluting the three-channel input feature vectors with four 33-point cosine filters. The convolution is performed in "same" mode by padding the input feature vector with appropriate zero points to ensure the length of the input and output feature vector are the same.

is also feasible to calculate and store all parameters of the cosine convolution kernels after training, and then perform inference as with traditional convolution:

$$\begin{aligned} x_i^l &= \left(w_m * O^{l-1} \right)_i \\ &= \sum_{m=0}^{k-1} w_m O_{i+m}^{l-1} \end{aligned}, \tag{3}$$

where $w_m = K_m\left(A^l, \omega^l\right)$ are the pre-computed parameters of cosine kernel. For cosine filters with long length, the convolution operation can be accelerated by Fast Fourier Transform (FFT). Assume the convoluted feature vector $O^{l-1} \in \mathbb{R}^L$, then the output feature vector $x^l \in \mathbb{R}^{L-k+1}$ can be obtained by the following equation:

$$x^l = \left[\!\left[\mathrm{Re}\left(\mathcal{F}_L^{-1}\left\{ \mathcal{F}_L\left\{ K\left(A^l, \omega^l\right)\right\} \odot \mathcal{F}_L\left\{ O^{l-1}\right\}\right\}\right)\right]\!\right]_k^L, \tag{4}$$

where $\odot$ stands for the Hadamard product, $\mathcal{F}_L\{\cdot\}$ and $\mathcal{F}_L^{-1}\{\cdot\}$ are the *L*-point FFT and inverse FFT, respectively. $\mathrm{Re}(\cdot)$ returns the real part of the input data, and $[\![\cdot]\!]_k^L$ truncates the feature vector from *k* to *L*. Fig. 1 demonstrates an example of cosine convolution with multiple input and output feature vectors. After cosine convolutions, batch normalization (Ioffe & Szegedy, 2015) is performed on each output channel to adjust the distribution of the output feature vectors and accelerate the convergence speed. Further, the Rectified Linear unit (ReLu) (Glorot et al., 2011) is optionally added to the batch normalization outputs to enhance the non-linearity of the model. For deep models with multiple stacked cosine convolutional layers, the max-pooling layer (Ranzato et al., 2007) is inserted between two convolutional layers to achieve dimension reduction and alleviate overfitting by computing the maximum of each specific region with a certain stride.

3.3. *Updating CosCNN*

At the training stage, the cross-entropy loss with an L_2 regularization term on the amplitude factors is employed. Let θ be the learnable parameter of the CosCNN, *p* be the number of samples in one mini-batch, *q* be the number of classes, and λ be the regularization coefficient. Then the optimization of loss function *E* is denoted as:

$$\arg\min_{\theta}\left\{ -\sum_{i=1}^{p}\sum_{j=1}^{q} s_{ij} \ln\left(\frac{\exp\left(z_j^{(i)}\right)}{\sum_{u=1}^{q}\exp\left(z_u^{(i)}\right)}\right) + \frac{\lambda}{2}\sum_{\text{except }\omega}\theta^2 \right\}, \tag{5}$$

where $z_j^{(i)}$ with respect to θ is the total weighted sum of the last output feature vector to *j*-th categorical neuron for sample *i*, s_{ij} is the indicator that the *i*-th sample belongs to the *j*-th class. In this study, empirically λ is set to 0.001 in all experiments, and Adam optimizer (Kingma & Ba, 2015) is used to update the weights of the model.

In this study, the backpropagation process is deployed to train the CosCNN, and the gradient of the amplitude factor A^l in *l*-th layer can be given as:

$$\begin{aligned}\frac{\partial E}{\partial A^l} &= \sum_{i=0}^{L-k}\frac{\partial E}{\partial x_i^l}\frac{\partial x_i^l}{\partial A^l} \\ &= \sum_{i=0}^{L-k}\delta_i^l\frac{\partial x_i^l}{\partial A^l} \\ &= \sum_{i=0}^{L-k}\sum_{m=0}^{k-1}\delta_i^l\cos\left(\omega^l\left(m-\frac{k-1}{2}\right)\right)O_{i+m}^{l-1} \\ &= \frac{1}{A^l}\sum_{m=0}^{k-1}K_m\left(A^l,\omega^l\right)\left(\text{flip}\left(\delta^l\right)*O^{l-1}\right)_m\end{aligned}, \tag{6}$$

where E is the loss function, δ^l is the error vector propagated from the deeper layer, $\text{flip}(\cdot)$ denotes the flip function. Similarly, the gradient of the frequency factor ω^l can be written as:

$$\begin{aligned}\frac{\partial E}{\partial \omega^l} &= \sum_{i=0}^{L-k}\frac{\partial E}{\partial x_i^l}\frac{\partial x_i^l}{\partial \omega^l} \\ &= \sum_{i=0}^{L-k}\delta_i^l\frac{\partial x_i^l}{\partial \omega^l} \\ &= \sum_{i=0}^{L-k}\sum_{m=0}^{k-1}-\delta_i^l A^l\left(m-\frac{k}{2}\right)\sin\left(\omega^l\left(m-\frac{k}{2}\right)\right)O_{i+m}^{l-1} \\ &= \sum_{m=0}^{k-1}\tilde{K}_m\left(A^l,\omega^l\right)\left(\text{flip}\left(\delta^l\right)*O^{l-1}\right)_m\end{aligned}, \tag{7}$$

where $\tilde{K}_m\left(A,\omega\right)$ is the partial derivative of $K_m\left(A,\omega\right)$ with respect to ω, which is defined as:

$$\tilde{K}_m\left(A^l,\omega^l\right)=-A^l\left(m-\frac{k}{2}\right)\sin\left(\omega^l\left(m-\frac{k}{2}\right)\right). \tag{8}$$

Additionally, the error vector δ^l utilized in l-th layer is given by:

$$\delta_i^l=\frac{\partial E}{\partial x_i^l}=\frac{\partial f\left(x_i^l\right)}{\partial x_i^l}\left(\delta^{l+1}*K_m\left(A^{l+1},\omega^{l+1}\right)\right)_i, \tag{9}$$

where $f(\cdot)$ is the activation function such as ReLu, and the convolution operation in Eq. (9) is the full convolution. In the CosCNN, the gradient for updating cosine kernels involves calculations of two additional trigonometric functions $K_m\left(A^l,\omega^l\right)$ and $\tilde{K}_m\left(A^l,\omega^l\right)$. Unlike the traditional convolutional kernels which require to update every kernel parameter, the cosine convolutional kernels contain only two parameters to be tuned, substantially reducing memory consumption during the backpropagation process.

3.4. *Quantizing CosCNN*

To make the model more memory-saving and computationally efficient, a novel quantization algorithm is proposed that can quantize the weights of CosCNN to a low-bit integer format. Since the proposed cosine convolution has to compute the cosine function that cannot be directly quantized, a cosine lookup table for the cosinusoidal convolution quantization is motivated to be established. Meanwhile, inspired by the post-training quantization method realized in TensorRT (Migacz, 2017), a KL divergence-based method is employed to compute the activation quantization scale (Q-scale) and quantization factor (Q-factor). The detailed CosCNN quantization process is illustrated in Algorithm 1, where $[\cdot]$ is the round function, N_l denotes the number of the cosine convolutional module, and BW_A, BW_ω, BW_{act}, BW_Ω represent the bit width of A, ω, activation outputs, and parameters in the quantized cosine lookup table, respectively. Notice that the proposed cosine filter is centrosymmetric, therefore only half the length of the quantized cosine filter is required to be computed and stored in the cosine lookup table $\bar{Q}$. Actually, $\bar{Q}$ can be further squeezed by eliminating the repeating terms. For example, a

Algorithm 1 CosCNN quantization algorithm

Input: Floating CosCNN, calibration dataset X_c, and the bit width BW_{act}, BW_A, BW_ω, BW_Ω

Output: Quantized CosCNN

Find the absolute maximum of ω in all layers of CosCNN and compute the Q-scale: $S_\omega = (2^{BW_\omega} - 1) \mid \omega \mid_{max}^{-1}$
Compute the Q-scale and Q-factor of the input signal using the absolute maximum of X_c : $S_{act}^0 = M^0 = (2^{BW_{act}-1} - 1) \mid X_c \mid_{max}^{-1}$
Compute the Q-scale of the parameters in quantized cosine lookup table: $S_\Omega = 2^{BW_\Omega - 1} - 1$
Initialize the quantized cosine lookup table $\bar{Q} \in \mathbb{Z}^{2^{BW_\omega} \times \lceil k/2 \rceil}$
For $j = 1...2^{BW_\omega}$ **do**
  For $m = 1...\lceil k/2 \rceil$ **do**
    $\bar{Q}(j,m) = \lceil S_\Omega \cos(j(m-1)/S_\omega) \rceil$
  End for
End for
For $l = 1...N_l$ **do**
  Merge the convolutional layer with its corresponding batch normalization layer
  Quantize ω: $\omega_Q^l = [S_\omega \mid \omega^l \mid]$
  Find the absolute maximum of A in *i-th* layer and compute the Q-scale: $S_A^l = (2^{BW_A - 1} - 1) \mid A^l \mid_{max}^{-1}$
  Quantize A: $A_Q^l = [S_A^l A^l]$
  Compute the activation Q-scale S_{act}^l using X_c and KL divergence-based calibration method
  Compute the Q-factor of the activation of the current layer: $M^l = S_{act}^l / (S_A^l S_\Omega S_{act}^{l-1})$
  Update the bias term: $B^l = S_{act}^l B^l$
End for
Compute the dequantization factor for the last convolutional layer: $M_{deQ} = 1 / S_{act}^{N_l}$
Assemble the network using quantized parameters
Return quantized network

$\bar{Q} \in \mathbb{Z}^{4096 \times 3}$ obtained from a quantized trained model (BW_ω and *k* are set to 12 and 5) can be squeezed to $\bar{Q} \in \mathbb{Z}^{721 \times 3}$, which reduces approximately 82.46% memory. Each cosine convolutional layer is merged with its following batch normalization layer before quantization to simplify the architecture and speed up the inference. The updated amplitude weight of the cosine filter and the bias term *B* can be computed as:

$$A = \frac{\gamma A}{\sqrt{\sigma_B^2 + \varepsilon}}, \tag{10}$$

$$B = \beta - \frac{\gamma \mu_B}{\sqrt{\sigma_B^2 + \varepsilon}}, \tag{11}$$

where μ_B and σ_B^2 are the mean and the variance gathered in the training stage by calculating over time and observation dimension, γ and β are the rescale factor and offset factor learned in batch normalization layer, $\varepsilon = 10^{-5}$ is a constant that improves numerical stability when the variance is very small. Since the range of feature vectors (activations) in each convolution layer changes with the input data of the model, it is important to calibrate the threshold for activations so that the quantized model can maintain its accuracy. In this study, the optimal activation threshold is determined by minimizing the KL divergence value between the normalized floating activation distribution and the normalized quantized activation distribution obtained from the calibration dataset. The detailed calibration procedure is illustrated in Algorithm 2. The KL divergence D_{KL} between H^* and $H_{QExpand}^*$ is given by:

$$D_{KL} = \sum_{n=1}^{N_{bin}} H^*(n) \ln\left(\frac{H^*(n)}{H_{QExpand}^*(n)}\right), \tag{12}$$

Algorithm 2 KL divergence-based calibration

Input: Calibration dataset X_c and the floating model

Output: Activation Q-scale S_{act}

Obtain the floating activations using X_c and the floating model
Compute the FP32 histogram *H* with 2048 bins using absolutized floating activations
For $i = 2^{BW_{act}-1}...2048$ **do**
    Take first *i* bins from *H* to H^*, and perform normalization: $H^* = H^* / sum(H^*)$
    Quantize the H^* into $2^{BW_{act}}$ levels and obtain H_Q^*
    Expand the H_Q^* to *i* bins and obtain $H_{QExpand}^*$
    Normalize the $H_{QExpand}^*$: $H_{QExpand}^* = H_{QExpand}^* / sum(H_{QExpand}^*)$
    Compute the KL divergence between H^* and $H_{QExpand}^*$
End for
Find the minimum KL divergence and record its corresponding index n_H
n_H multiply by the width of a bin is the optimal threshold $T_{optimal}$
Compute the activation Q-scale: $S_{act} = (2^{BW_A-1}-1)T_{optimal}^{-1}$
Return S_{act}

where N_{bin} is the number of bins in H^* or $H_{QExpand}^*$.

The cosine convolution with quantized weights can be written as:

$$x_{Q,i}^l = \left[B^l + M^l \sum_{m=0}^{k-1} A_Q^l \overline{Q}\left(\omega_Q^l, |\, m+1-\lfloor k/2 \rfloor |\right) O_{Q,i+m}^{l-1} \right], \tag{13}$$

where $x_{Q,i}^l$ and $O_{Q,i}^{l-1}$ represent the *i*-th value of the quantized output feature vector in the *l*-th layer and the *i*-th value of the quantized feature vector obtained from the previous layer. The inference phase of the quantized CosCNNs is demonstrated in Algorithm 3. The FP32 Q-factor M^l and the FP32 bias term B^l in the *l*-th layer can also be converted to integer format by multiplying with 2^{n_B} and then having it rounded, where $n_B \in \mathbb{N}_+$. After calculating convolution, the result has to multiply 2^{-n_B}, which can be implemented with an efficient bit shift operation. In this work, $n_B = 23$ is empirically set in all experiments.

3.5. *Hardware Implementation*

To verify the effectiveness and feasibility of the proposed CosCNN, we implement the quantized CosCNN on the Xilinx Zynq Zedboard, which has the Zynq-7000 SoC's tightly coupled ARM Cortex-A9 Processing System (PS) and 7 series Programmable Logic (PL). The whole hardware implementation

Algorithm 3 Inference phase of quantized CosCNN

Input: Quantized CosCNN and testing sample X_S

Output: Predicted label

Quantize the X_S : $O_Q^0 = M^0 X_S$
For $l = 1...N_l$ **do**
    Compute the convolution using Eq. (13) and obtain x_Q^l
    Compute the activation function (optional)
    Dequantize and quantize the output: $x_Q^l = M^l x_Q^l + B^l$
    Compute the max-pooling and obtain O_Q^l
End for
Dequantize the output: $x_Q^{N_l} = M_{deQ} x_Q^{N_l}$
Compute the fully connected layer
Compute the softmax mapping
Find the maximum value and its corresponding label
Return predicted label

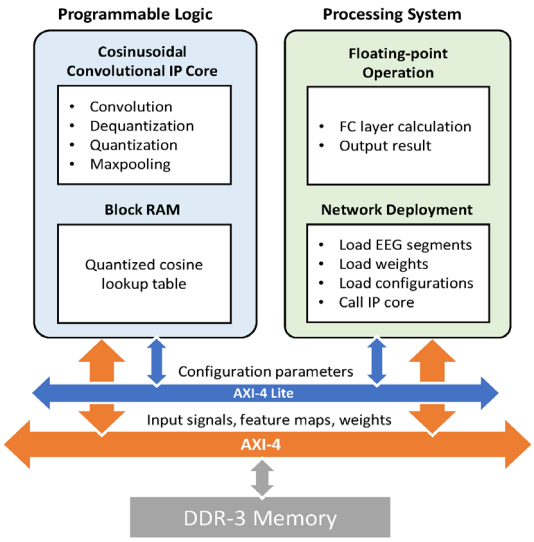


Fig. 2. The hardware implementation scheme of the quantized CosCNN.

scheme is depicted in Fig. 2. In the PL part, a configurable cosine convolution accelerator is designed using Vivado High Level Synthesis (HLS) tools (2019.1 version). The cosine convolution, dequantization and quantization of the activations, and max-pooling operation with C++ code are realized firstly. Then compiling them to Verilog codes and generating Intellectual Property (IP) cores as hardware accelerator by HLS. All loops are optimized by adding "PIPELINE progma", an HLS command that can automatically pipeline the loop. Since all convolution layers share one quantized cosine lookup table, it is stored in Random Access Memory (RAM) block of PL. The PS is leveraged for data preparation and preprocessing, such as receiving EEG data, loading weights, and configuring the accelerator. Additionally, the FC layer with FP32 weights and softmax mapping is also computed in PS. For a CosCNN with multiple layers, configuration parameters of the first layer, including the number of input and output channels, the length of the feature vector and cosine filter, n_B, and the quantized Q-factor, are first sent to the accelerator in PL by AXI-Lite bus. Then the input signal or feature vector, quantized bias terms, and weights are sent by AXI-Full bus. When the accelerator finishes the computation, the output data is transferred to the PS using the same AXI-Full bus. Subsequently, the accelerator is reconfigured using the configuration parameters of the next layer with AXI-Lite bus, and the data previously inputted into the PS are fed into the accelerator for a new round of calculation by AXI-Full bus. Repeat the aforementioned procedures until reaching the last convolutional layer. All data defined in PS are stored in DDR-3 memory. The designed configurable cosine convolution accelerator is synthesized by Vivado 2019.1, and the routines in PS are programmed using Xilinx SDK 2019.1.

4. Experiments and Results

Experiments are conducted on two publicly available epileptic EEG databases, namely the Bonn database and the CHB-MIT database. In this section, the two EEG databases are introduced, and the comprehensive experimental setups and the corresponding results on these databases are presented.

4.1. *Experimental EEG databases*

Two EEG databases are employed in this work to evaluate the effectiveness of the proposed method. The first database was collected by Bonn university (Andrzejak et al., 2001). Three subsets of the Bonn database, namely set B, set D, and set E, are utilized in the present study, in which each subset has 100 single-channel EEG segments that correspond to a certain type of EEG signal. Each segment is with a length of 23.6 seconds in a sampling rate of 173.6Hz. In detail, set B was collected from five healthy volunteers with eyes closed, and set D and set E originated from five epileptic patients, where set D was recorded from the epileptogenic zone when the patients were in the interictal stage and set E only contained the EEG segments that were inspected to be seizure activity. All the EEG segments are band-pass filtered to frequency range of 0.53-40Hz. In this study, 10-fold cross-validation is adopted to evaluate the performance of the CosCNN on the Bonn database. The whole database is randomly split into 10 folds, and each class of the EEG data is averagely distributed in each fold. In the training phase, each fold of data is used as the test set in turn, and the remaining 90% of the database serve to train and validate the model. Within this 90%, 70% of the data are randomly selected to train the model, while the remaining 30% are used as validation set to monitor the generalization ability of the model and serve as the calibration dataset for the post-training quantization algorithm. The trained models were evaluated only on testing sets, with the average accuracies of 10-fold cross-validation being reported. This configuration aligns with the setup used in a classic deep learning-based epileptic EEG classification study on Bonn database (Acharya et al., 2018).

Another EEG database named CHB-MIT database which is used in this study is recorded from 24 children subjects at the Children's Hospital Boston (Shoeb & Guttag, 2010). This database consists of approximately 980-hour scalp EEG recordings with a sampling rate of 256Hz. Most EEG recordings have 18-23 bipolar EEG electrodes placed in the international 10-20 system. In this work, the EEG data from 18 common bipolar electrodes (FP1-F7, F7-T7, T7-P7, P7-O1, FP1-F3, F3-C3, C3-P3, P3-O1, FP2-F4, F4-C4, C4-P4, P4-O2, FP2-F8, F8-T8, T8-P8, P8-O2, FZ-CZ, CZ-PZ) are employed. A total of 184 seizure events are labeled by experts in the CHB-MIT database, of which 40 seizure events are for training, and others are for testing. For most patients, the first seizure event is adopted as training data. And for patients 6, 12, 13, and 16, four to eight seizure events are utilized due to the short seizure duration and higher frequency of seizure attacks. Because the training data is limited, the non-seizure data five times the length of the seizure data is randomly selected. Correspondingly, the seizure data are overlapped upsampled five times to balance the training and testing sets. In order to quantize the model correctly and consider the imbalanced data, for each patient, 20 minutes of EEG signals without seizure activities are selected as the calibration dataset. In summary, a total of 3.04h recordings are used as training while the rest of 976.89h recordings serve as the testing set. The detailed information of this database is summarized in Table A.1 of Appendix A (Supplementary Material).

The CHB-MIT database comprises continuous long-term scalp EEG recordings collected from clinical environments, inherently containing various types of noise and artifacts (Shoeb & Guttag, 2010). Before feeding them into the deep learning models for seizure detection, EEG preprocessing steps, including EEG segmentation and filtering, are commonly adopted. Following previous studies of seizure detection based on the CHB-MIT database (Evangelidis & Kugiumtzis, 2023; Yuan et al., 2018), the EEG recordings in the CHB-MIT database are segmented into 4-s non-overlapping segments for training and testing. To eliminate noises and artifacts in the CHB-MIT database, we filter each EEG segment by

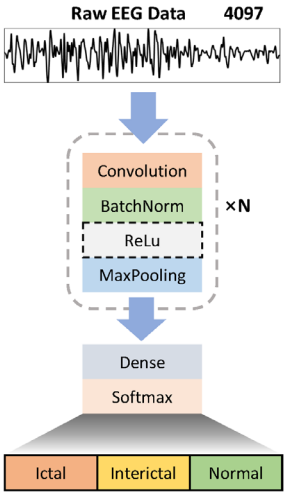


Fig. 3. The experimental CosCNN architecture with N cosine convolutional blocks. The n-th cosine convolutional block contains a convolutional layer with the output channel number of 2^{n+1}, a batch normalization layer, an optional ReLu layer, and a max-pooling layer with a nonoverlapping pooling length of 2.

Discrete Wavelet Transform (DWT) with Daubechies-4 (Db4) wavelet. Previous studies indicated that the DWT with Db4 wavelet had been successfully employed for epileptic EEG classification (Faust et al., 2015; Ficici et al., 2022). DWT decomposes the EEG signals into five scales that correspond to 64-128Hz (d1), 32-64Hz (d2), 16-32Hz (d3), 8-16Hz (d4), and 4-8Hz (d5). Besides, an approximation term corresponding to 0-4Hz (a5) is also yielded. d3, d4, and d5 scales are reconstructed to acquire the EEG data with the frequency band of 4-32Hz for subsequent analysis. In this study, all experiments are carried out in Matlab R2021a, executing on a workstation with an Intel i9-13900K CPU, an Nvidia RTX3090 GPU, and 64 GB memory.

4.2. *Experiments and results on Bonn database*

4.2.1. Experimental setup

To comprehensively verify the accurateness and effectiveness of the proposed CosCNN, numerous comparison experiments are conducted on Bonn database. As is demonstrated in Fig. 3, experimental CosCNN architectures with N cosine convolutional blocks are constructed, where $N \in \{1,2,...,8\}$. The last max-pooling layer is followed by a dense layer with three output neurons, which are mapped into probabilities of three classes by the softmax layer. Eight different filter lengths, namely 5, 9, 13, 17, 21, 25, 29, 33, are considered in each CosCNN architecture. Besides, six different experimental settings shown in Table 1 are applied. Each experiment is conducted using a 10-fold cross-validation scheme, and the mean accuracy is reported as the performance metric. A total of 8×8×6×10=3840 models are trained,

Table 1

The experimental settings on Bonn dataset.

Exp.	Type	Update ω	Update A	Update W	ReLu
1	CNN	-	-	√	√
2	CNN	-	-	√	×
3	CNN	-	-	×	×
4	CosCNN	√	√	-	√
5	CosCNN	√	√	-	×
6	CosCNN	×	×	-	×

Note: Exp. is the abbreviation of Experiment. W is the learnable weight in traditional CNNs. A and ω are the learnable weights in CosCNNs.

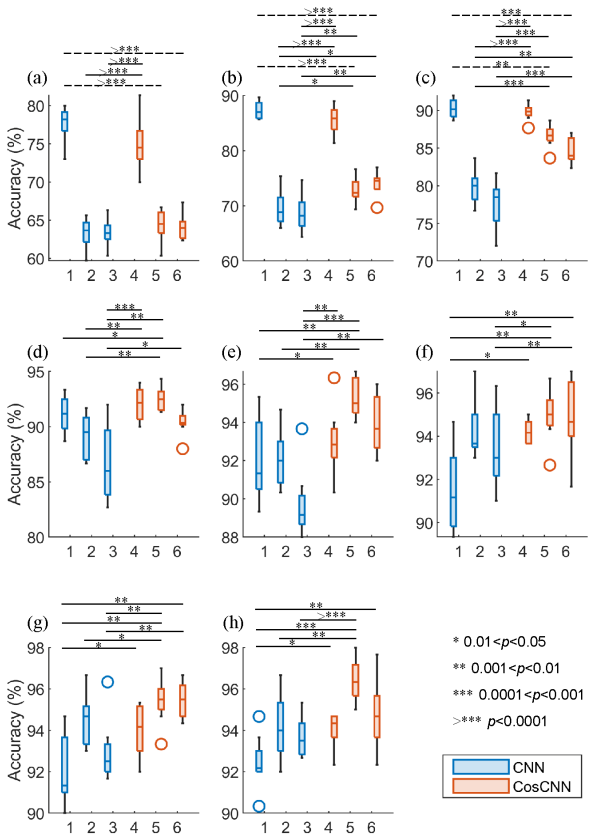


Fig. 4. The experimental results on the Bonn database. (a)-(h) correspond to the results of the models with 1-8 convolutional blocks. The X-axis represents the experimental setting index. Each box presents the summary statistics of cross-validation mean accuracies over the models with all possible filter lengths. The *t*-test is conducted between all CNN and CosCNN experiment pairs, and the pairs with a significant difference are marked, where the solid line indicates the result of CosCNN is significantly better than CNN while the dashed line is the opposite.

and all the models are updated by 240 epochs in mini-batch size of 90. The learning rate was initially set to 2×10^{-4} and exponentially decreased to 2×10^{-5} within 240 epochs. The average training time for each cosine convolutional network model and traditional convolutional network model is 16.9 seconds and 12.2 seconds, respectively. In addition, the quantized CosCNNs with 8 cosine convolutional blocks and various bit width and kernel length settings are deployed on the hardware platform to verify the proposed CosCNN quantization algorithm.

4.2.2. Experimental results

A comprehensive statistical analysis of the results is presented in Fig. 4. It can be observed from Fig. 4 that for the traditional CNNs and CosCNNs, the performance of the model improves with the increase

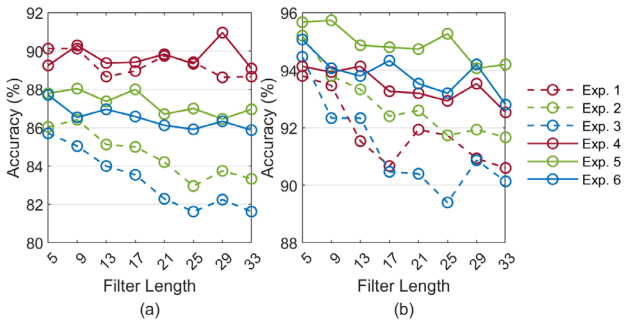


Fig. 5. The effect of different filter lengths and experimental settings on the model performance. (a) shows the mean accuracies over the models with 1-8 convolutional blocks. (b) shows the mean accuracies over the deep models with 4-8 convolutional blocks.

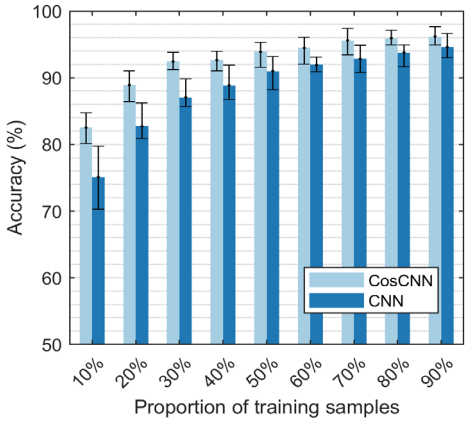


Fig. 6. The experimental results on the models trained with various proportions of the original training set. The CosCNN and traditional CNN correspond to the experiment setting of Exp. 5 and Exp. 2. All models consist of 5 convolutional blocks, and each bar represents the mean accuracy over the model with all possible filter length. The error bar denotes the standard deviation.

of the model depth. Particularly, the model with 8 cosine convolutional blocks and 5-point convolutional filters in Exp. 5 reaches the highest mean accuracy of 98.00%. Meanwhile, observing that the shallow model with ReLu has a better performance compared to the model without ReLu. However, as the model deepens, ReLu becomes the obstacle to learning effective features of the network. This may be because the shallow network urgently needs more nonlinear functions to adapt to the complicated distribution in training set while enough nonlinear properties are provided by max-pooling layers in the deep network. Another interesting phenomenon is that even if the CosCNNs do not update the cosine filters, it can still achieve considerably high performance, proving that the proposed cosine filter is an effective weight initialization method for EEG classification tasks. The *t*-test results indicate that CosCNNs are significantly better than traditional CNNs in all cases except the cases of the shallow model with ReLu. Fig. 5 demonstrates the effect of filter length on model performance. It is obvious that a longer filter length may lead to worse model performance. Nevertheless, CosCNNs are less affected by the filter length than traditional CNNs. This is mainly because the cosine filter adds prior knowledge to its frequency response, and the few learnable parameters enable the model to have better generalization ability. For the traditional kernel, the longer the kernel, the more the learnable weights, which makes the

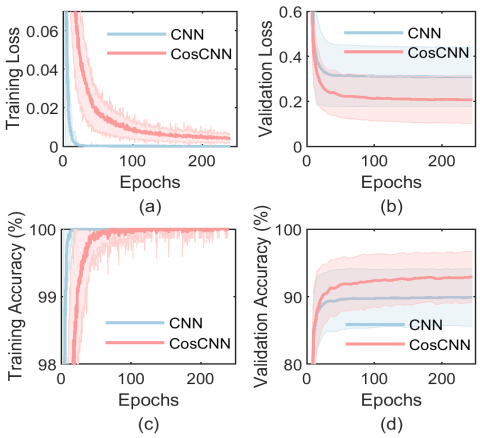


Fig. 7. Average training process curves for CosCNN (Exp. 5) and traditional CNN (Exp. 1) with eight convolutional blocks across all possible filter length settings. (a) and (b) describes the changing process of the mean training loss and validation loss, while (c) and (d) demonstrates the changing process of the training accuracy and validation accuracy. The shaded region indicates the standard deviation region.

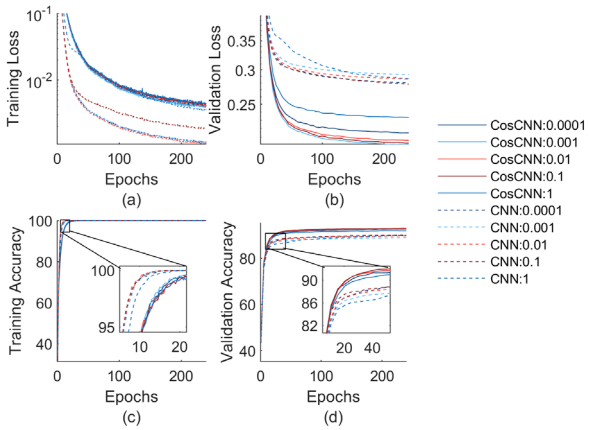


Fig. 8. Average training process curves for CosCNN and traditional CNN with five convolutional blocks across all possible filter length settings. (a) and (b) describes the changing process of the mean training loss and validation loss, while (c) and (d) demonstrates the changing process of the training accuracy and validation accuracy. The solid and dashed lines indicate CosCNNs and traditional CNNs with different regularization coefficients, respectively.

model tends to over-fit. From Fig. 5 (b), it indicates that deep CosCNNs without ReLu (Exp. 5) achieves superior results over all the other experimental settings.

Furthermore, as illustrated in Fig. 6, models are trained by using different numbers of training samples to explore the few-shot learning ability of CosCNNs. It can be found that the fewer the samples in the training set, the more obvious the performance advantage of CosCNNs, which shows that CosCNNs have better generalization ability and are more suitable for few-shot learning. According to the training progress plotted in Fig. 7, the traditional CNNs perform well on the training set and poorly on the validation set. In contrast, CosCNNs excel on the validation set while performing poorly on the training set. This demonstrates the effectiveness of incorporating cosine filters into convolutional neural networks, and also suggesting that cosine kernels effectively enhance the model generalization ability and prevent overfitting. Theoretically, traditional CNNs can also prevent overfitting by increasing the L_2 regularization coefficient. Therefore, Fig. 8 investigates the impact of different regularization coefficients on the performance of traditional CNNs and CosCNNs. It is observed that during training, CosCNNs consistently exhibit lower validation loss and higher validation accuracy compared to traditional CNNs. For traditional CNNs with a larger regularization coefficient (equal to 1), although their training loss curves are comparable to those of CosCNNs, their training accuracy and validation accuracy remain significantly lower than those of CosCNNs, and their validation loss is also significantly higher. This

Table 2

The experimental results with quantized CosCNN. The mean accuracy of 10-fold cross-validation is reported. The highest mean accuracy is marked in bold face.

No.	Bit width BW_ω - BW_Ω - BW_A - BW_{act}	Mean accuracy (%) k=5	k=33
1	/-32-32-32	98.00	95.00
2	8-8-8-8	86.00	72.00
3	10-8-8-8	**98.33**	90.67
4	12-8-8-8	97.00	95.00
5	16-8-8-8	96.67	**95.67**
6	12-4-4-4	52.33	51.00
7	12-5-5-5	59.67	69.67
8	12-6-6-6	80.00	85.67
9	12-7-7-7	89.33	93.00
10	12-4-8-8	71.00	83.67
11	12-8-4-8	50.33	63.67
12	12-8-8-4	66.67	83.33

further underscores the effectiveness of cosine convolutional kernels. According to Fig. 8 (b), when the regularization coefficient in the loss function of CosCNNs is set to 0.001, it achieves the lowest loss value on the validation set. Hence, in this study, the regularization coefficient λ in the loss function of all CosCNNs is set to 0.001.

The results obtained above are with FP32 models, and the results on quantized models with different bit width settings are depicted in Table 2. Two models with 8 convolutional blocks and the kernel lengths that are set as 5 and 33 are evaluated under different bit widths of different parameters. The result with bit width setting of "/-32-32-32" serves as the baseline is from the FP32 model. Experiment 2-5 manifest that the quantized model with a kernel length of 5 can achieve the highest mean accuracy of 98.33% under "10-8-8-8" bit width setting, and the quantized model with a kernel length of 33 realize a mean accuracy of 95.67% under "16-8-8-8" bit width setting. However, the performance of the model with a kernel length of 33 drops 4.33% accuracy under "10-8-8-8" bit width setting. Meanwhile, the quantized cosine lookup table will have 65536 rows when the BW_{ω} is set to 16, which can consume numerous resources for the hardware platform. Therefore, the bit width setting of "12-8-8-8" is an optimum selection for quantizing the model. The results of experiments 6-12 prove that CosCNNs with a long convolutional kernel length is more robust to BW_{Ω}, BW_{A}, and BW_{act} than CosCNNs with a short convolutional kernel length. Simultaneously, experiments 10-12 indicate that the quantized CosCNNs are most sensitive to BW_{A}, followed by BW_{act}, and least sensitive to BW_{Ω}. Overall, the comprehensive results on quantized CosCNNs reveal the effectiveness and advances of the proposed CosCNN quantization method based on the quantized cosine lookup table and KL divergence. It is noteworthy that all these results are obtained from the aforementioned Zedboard hardware platform.

4.3. *Experiments and results on CHB-MIT database*

4.3.1. Experimental setup

Since the model with 8 convolutional blocks and a filter length of 5 achieves the highest performance on Bonn database, a similar CosCNN architecture is established as presented in Fig. 9 to assess the performance of the CosCNN on CHB-MIT database. The input of the CosCNN is the preprocessed 4-s EEG segment with 18 channels. And numbers of output channels are 32, 32, 64, 64, 128, 128, 256, 256 for 8 cosine convolutional layers. Each convolutional layer is followed by a batch normalization layer and a max-pooling layer with a non-overlapping pooling size of 2. No activation function is added. The feature vectors obtained from the last max-pooling layer are flattened and fed into the dense layer with

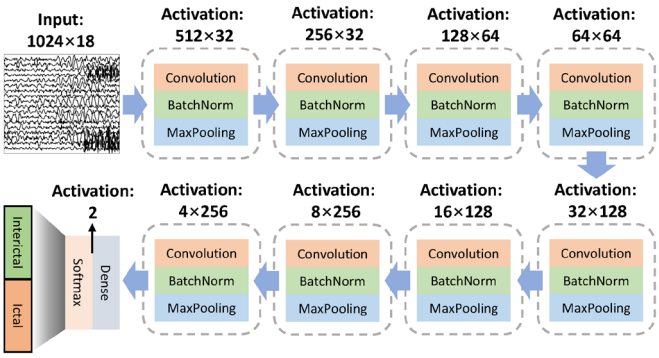


Fig. 9. The CosCNN architecture used in CHB-MIT database.

two output neurons. Then the softmax mapping is performed on these two neurons to acquire the predicted score representing the seizure probability. After that, the postprocessing operations (Liu et al., 2020), including smoothing, thresholding, and collar, are employed to remove the isolated false detection and improve the sensitivity and specificity of the detection results. The model was trained for 500 epochs with a mini-batch size of 128, and the learning rate was initially set to 2×10^{-4} and exponentially decreased to 2×10^{-5} within 500 epochs.

As for the performance evaluation method, the segment-based, the event-based, and the Area Under the Receiver Operating Characteristic curve (AUROC)-based approaches are employed for assessing the performance of CosCNN on the CHB-MIT database. Three performance metrics, namely sensitivity, specificity, and accuracy, are computed for the segment-based approach. The sensitivity is defined as the percentage of the number of seizure segments correctly predicted to the number of the seizure segments that experts labeled. The specificity is the percentage of the number of interictal segments correctly predicted to the number of interictal segments marked by the experts. The accuracy is defined as the percentage of the number of all the EEG segments correctly labeled to the total number of the EEG segments. For the event-based evaluation approach, the event-based sensitivity is calculated, which is defined as the ratio of the number of seizure events correctly detected to the number of seizure events marked by experts. Besides, the False Detection Rate (FDR) is another event-based indicator that reflects the false alarm per hour. Moreover, the AUROC is utilized to evaluate the robustness of the model and compare the performance of different models intuitively. When computing the AUROC, uniformly setting the length of the smooth window and the collar to 24-s. Since CHB-MIT database had 24 patients, 24 patient-specific CosCNN models were trained and the average performance metrics were reported.

4.3.2. Experimental results

For the segment-based evaluation scenario, a mean sensitivity of 98.12% and a mean accuracy of 98.18% are fulfilled by the quantized CosCNN under a mean specificity of 98.19%. A total of 16 patients' detection results achieve 100% sensitivity and over 99% specificity. And for the event-based evaluation scenario, 143 of 144 seizure events are successfully detected under an FDR of 0.69, achieving an event-based mean sensitivity of 99.31%. All patients except patient 18 reach an event-based sensitivity of 100%, where half can achieve an FDR below 0.1/h. The detailed segment-based and event-based results can be found in Table A.2 and A.3 of Appendix A (Supplementary Material). Note that all the results reported in this section are with the quantized model, and the reported performance is exactly the same as the results obtained from the Zedboard hardware platform. Another two comparative experiments are conducted to highlight the advances of the quantized CosCNN. As shown in Table 3, the traditional FP32 CNN model, FP32 CosCNN model, and quantized CosCNN model are compared. Except for the convolution type and quantization process, all the other experimental hyperparameters are exactly the same. Both the FP32 CosCNN and quantized CosCNN outperform the FP32 CNN model ($p<0.05$).

Table 3

Results comparison on the CHB-MIT database.

Model	AUROC	Parameters	Model Size	MAC
CNN	96.04%±7.53%	658.1k	2.51MB	3.09MB
CosCNN	98.26%±3.43%	265.2k	1.01MB	1.59MB
CosCNN*	98.31%±3.34%	266.4k	0.26MB	0.40MB

Note: * represents the quantized model. MB stands for Million Bytes.

Table 4

Comparison results on Bonn database

Author (Year)	Model	Params†	Params°	Accuracy (%)	*p*-Value
Acharya et al. (2018)	CNN	96.2k	1.31k	90.80±6.04	0.0188
	CosCNN	94.8k	0.62k	93.40±4.74	
Ullah et al. (2018)	CNN	83.8k	1.70k	96.93±3.49	0.0098
	CosCNN	83.1k	1.10k	98.07±2.62	
W. Zhao et al. (2020)	CNN	101.5k	53.00k	97.53±2.41	0.0398
	CosCNN	56.6k	8.04k	98.27±2.54	
This work*	CNN	902.9k	874.80k	94.20±4.75	$<10^{-6}$
	CosCNN	377.6k	349.44k	98.40±2.54	

Note: † represents all the parameters of the model. ° represents the convolutional parameters of the model. * The model with 8 convolutional layers and 5-point filter length is compared.

Particularly, the quantized CosCNN model obtains a higher AUROC with a 2.27% improvement compared to the FP32 CNN model, while the model parameter is 59.51% reduced, the model size is 89.64% reduced, the Memory Access Cost (MAC) is 87.06% reduced. The detailed AUROC results of each patient are provided in Table A.4 of Appendix A (Supplementary Material).

5. Discussion

5.1. *Performance Comparisons*

In this study, the CosCNN is proposed and evaluated on the widely used Bonn database and CHB-MIT database. There also exist some other seizure detection methods evaluated on these two databases. As is shown in Table 4, three representative state-of-the-art 1-D CNN models designed for the Bonn database are reproduced and compared. The 5×10-fold validation results with significance test are reported. All experimental settings, including training/testing schemes and model architectures, are reproduced based on the original literature. According to the alternative approaches illustrated in Fig. 10, the performance of their corresponding CosCNN model using the same hyperparameters is fully explored. Notably, as shown in Fig. 10 (c) and (d), the traditional convolutional module without a maxpooling layer is replaced with the cosine convolutional module with a single-stride maxpooling layer that has a pooling size $r>1$ and a stride $s=1$. Empirically, the result is not sensitive to the pooling size of the single-stride maxpooling layer, and it is set to 8 in this study. As the results show in Table 4, the alternative CosCNNs can realize significantly better accuracies than traditional CNNs ($p<0.05$). Meanwhile, the model parameters, especially the convolutional parameters, are reduced. Since the 1-D CNN model proposed by Wei Zhao et al. (2020) has a longer filter length, convolutional parameters are 84.83% reduced when the model is converted to CosCNN model.

On the other hand, the comparison of different state-of-the-art seizure detection methods evaluated on

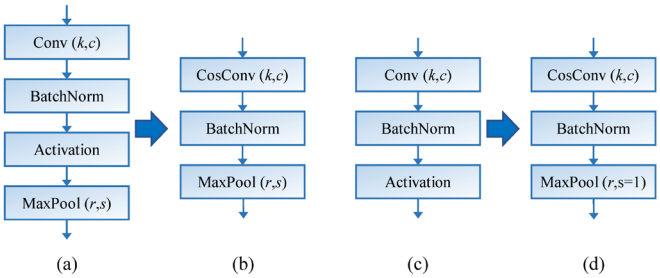
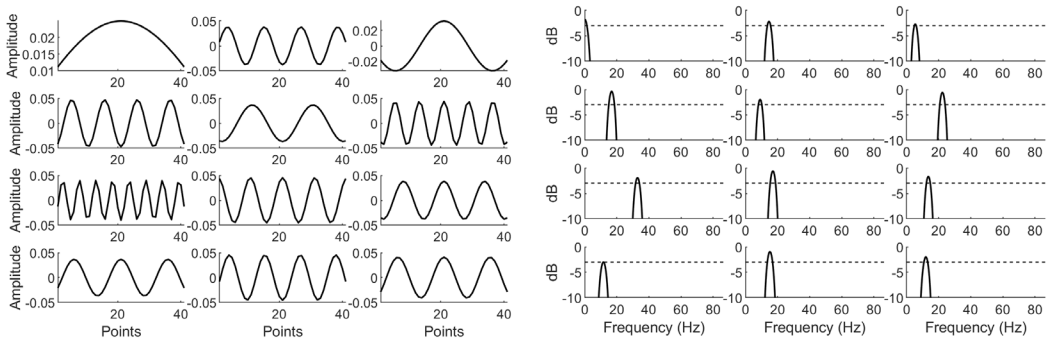


Fig. 10. The convolutional block of traditional CNN and CosCNN. (a) and (b) are the alternative scheme for convolutional module with maxpooling layer. (c) and (d) are the alternative scheme for convolutional module without maxpooling layer. k and c are the filter length and filter number of the convolutional layer. r and s are the pooling size and the stride of the maxpooling layer.

Table 5
Comparison of the performance for different methods on CHB-MIT dataset

No.	Author (Year)	EEG data used (h)	Number of used cases	Number of training / used seizures	Segment-based sensitivity (%)	Specificity (%)	AUC (%)	Event-based sensitivity (%)	FDR (/h)
1	Bhattacharyya and Pachori (2017)	178	23	-/157	97.91	99.57	99.9	-	-
2	Yuan et al. (2018)	958.2	24	42/187	95.65	95.75	-	94.48	0.68
3	Selvakumari et al. (2019)	-	24	-	97.50	94.50	-	-	-
4	Li et al. (2020)	846.23	24	63/198	95.42	95.29	-	94.07	0.66
5	Zabihi et al. (2020)	172	23	-/-	91.15	95.16	93.16	-	-
6	Wang et al. (2021)	518	24	121/145	88.14	99.62	-	99.31	0.2
7	C. Li et al. (2021)	976.9	24	54/185	97.34	97.50	-	98.47	0.63
8	Zhang et al. (2022)	870.44	24	73/198	93.89	98.49	-	95.49	0.31
9	Sopic et al. (2022)	996	24	-/198	96.00	-	-	90.40	0
10	This work	979.93	24	40/184	98.12	98.19	98.31	99.31	0.69

the CHB-MIT database is demonstrated in Table 5. Li et al. (2020) combined the 1-D CNN and Long Short-Term Memory (LSTM) for seizure detection, with results of 95.42% segment-based sensitivity and 94.07% event-based sensitivity on 846.23-hour EEG recordings. And Wang et al. (2021) designed a 1-D CNN architecture composed of two branches to detect seizure onsets, achieving higher specificity and lower FDR than the results we reported. However, 121 of 145 seizure events in 518-hour EEG data were utilized for training the models. Therefore, though some of the performance metrics are higher than ours, they are not fully comparable. Bhattacharyya and Pachori (2017) extracted the EEG features using empirical wavelet transform (EWT), obtaining an AUROC and specificity over 99%. Similarly, considering a selected 178-hour EEG data were leveraged, the model performance cannot be simply compared according to the metrics reported. Sopic et al. (2022) achieved zero FDR by using a template matching-based algorithm. However, the personalized seizure signature used in their method is manually selected, which leads to less automation of seizure detection. Overall, the proposed quantized CosCNN yields the state-of-the-art segment-based sensitivity and event-based sensitivity of 98.12% and 99.31%, with the least number of seizure events for training. These advanced results obtained indicate that the proposed CosCNN has strong feasibility in clinical utilization. It should be noted that the CosCNN architecture evaluated on the CHB-MIT database is derived from the architecture optimized on the single-channel Bonn database, and no specific architectural optimizations are made for the CHB-MIT database. Hence, better performance can be expected when the architecture is further optimized for the multi-channel EEG database.



(a) (b)

Fig. 11. Twelve cosine kernels randomly collected from the first layer of the CosCNN model (filters with a maximum frequency response higher than -3dB were considered). The format is the same as in Fig. 1.

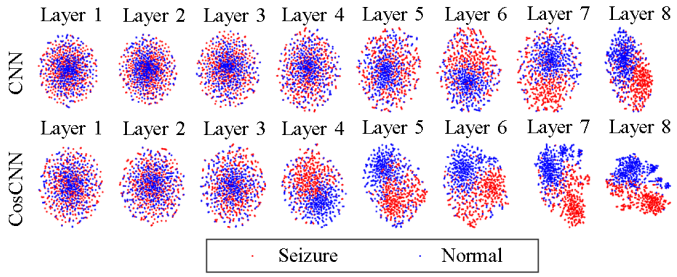


Fig. 12. The t-SNE clustering visualization of the output feature vector for each convolutional layer. The first row corresponds to the traditional CNN, and the second row corresponds to the CosCNN. The input seizure and normal data are 378 seizure and normal segments collected from the testing set of patient 12 in the CHB-MIT database.

5.2. *Feature Visualization*

Fig. 11 provides 12 cosine filters randomly collected from the first convolutional layer of a trained CosCNN model that has the same architecture as the model proposed by Wei Zhao et al. (2020). It is evident that filters with frequency below 30Hz are learned, which is consistent with the previous findings that the epileptic waveforms were mainly concentrated in the frequency band between 0.5-30Hz (Gotman, 1982; Shoeb & Guttag, 2010). This also suggests that the CosCNN can effectively extract discriminative ictal EEG features. Meanwhile, to figure out how the deep network learns features layer by layer, Fig. 12 offers the t-SNE maps of the intermediate feature vectors for each layer. It can be seen that as the number of layers increases, more separable features will be learned. For the traditional CNN, the features from the first five layers are inseparable, while layers 6-8 gradually learn the useful features. By contrast, the CosCNN learns separable features from the 4-th layer, and the clustering output of the last layer in CosCNN is lower-coupling and higher-cohesion. This proves the superior feature extraction ability of the CosCNN model.

5.3. *Complexity Analysis*

Compared to the traditional CNN, the proposed CosCNN with fewer learnable parameters has a lower memory occupation. Table 6 presents a comparison of the complexity between different types of convolutional modules, where u represents the size of the selected data type in Bytes, k is the kernel length, L denotes the length of the input and output signals (assuming that the input and output lengths are the same), and C_{in}, C_{out} are the number of input and output channels of the convolutional module. For the FP32 models, u=4, whereas for the quantized INT8 models, u=1. Notably, for the cosine convolution implemented with Eq. (2), the parameter scal is only $2/k$ of that in traditional convolution, significantly reducing the MAC. Due to the significantly reduced number of learnable parameters in the CosCNN

Table 6

Complexity comparison of different convolution methods.

Module	Parameters	MAC (Bytes)	Time Complexity	FLOPs
Traditional convolution	$kC_{in}C_{out}$	$uL(C_{in}+C_{out})+ukC_{in}C_{out}$	$O(kLC_{in}C_{out})$	$kLC_{in}C_{out}$
Cosine convolution[1]	$2C_{in}C_{out}$	$uL(C_{in}+C_{out})+2uC_{in}C_{out}$	$O(kLC_{in}C_{out})$	$\geqslant 3kLC_{in}C_{out}$ *
Cosine convolution[2]	$kC_{in}C_{out}$	$uL(C_{in}+C_{out})+ukC_{in}C_{out}$	$O(kLC_{in}C_{out})$	$kLC_{in}C_{out}$
Cosine convolution[3]	$kC_{in}C_{out}$	$uL(C_{in}+C_{out})+ukC_{in}C_{out}$	$O(L\log_2(L)C_{in}C_{out})$	$6L\log_2(L)C_{in}C_{out}$

* The FLOPs of computing cosine function are not considered. [1] Implemented with Eq. (2).
[2] Implemented with Eq. (3). [3] Implemented with Eq. (4).

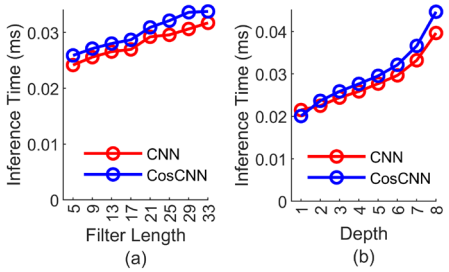


Fig. 13. Average inference time (per sample) for CNN and CosCNN models. (a) Average inference time for models with different filter lengths. (b) Average inference time for models with different depths.

compared to the traditional CNN, along with the integration of effective periodicity prior information, CosCNN exhibits stronger generalization capabilities and is less prone to overfitting (see Fig. 7 and Fig. 8). While significantly reducing the space complexity, the computation of the cosine function in cosine convolution modules also results in a slight increase in floating-point operations (FLOPs). Fig. 13 compares the GPU inference time for a single sample from the Bonn database between CosCNN models implemented using Eq. (2) and traditional CNNs under various configurations. All values represent the average inference time over ten repetitions using the FP32 model. Results indicate that due to the higher FLOPs, the inference time of the CosCNN is slightly higher than that of the traditional model (approximately 1.07 times). Nonetheless, as shown in Fig. 13 (b), in shallower models, the inference time of CosCNN models might even be lower than the traditional CNNs due to significantly reduced MAC. Given that the CosCNN achieves higher accuracy with the parameter scale only about $2/k$ of that in the traditional CNN, it has considerable advantages over traditional CNN models. Meanwhile, since the proposed CosCNN quantization algorithm is based on Eq. (2), these complexity measures are also applicable to quantized cosine convolution module. Moreover, for devices with extremely limited computational resources, inference process of the cosine convolution module can be implemented using Eq. (3), which is entirely consistent with traditional convolution, thus maintaining its complexity. For long convolution kernels, especially when the length of the convolution kernel is similar to the length of the input signal, the FFT-based cosine convolution module as formulated in Eq. (4) can be employed to accelerate the CosCNN model.

5.4. *Evaluation of Hardware Acceleration*

As aforementioned, a hardware accelerator is executed on FPGA for computing cosine convolution, and all the quantized models are deployed and tested on Zedboard. To verify the advances of the hardware accelerator and the effectiveness of the proposed quantization algorithm, a cosine convolution module is also realized in PS for comparison, which can simulate the hardware execution efficiency of the unquantized CosCNN running on ARM processors. Table 7 demonstrates the comparison results on

Table 7

The time consumption and energy efficiency of the CosCNN deployed in Xilinx Zynq Zedboard

Database	Segment Length (s)	Accelerator Platform†	Time (s)	Energy (J/segment)
Bonn	23.6	PS	94.57	159.82
Bonn	23.6	PL	1.11	1.94
CHB-MIT	4	PS	48.01	81.14
CHB-MIT	4	PL	1.04	1.82

Notes: † The PS runs at 667MHz while the PL runs at 100MHz.

Table 8

The resource utilization in PL of Zedboard for the CosCNN model used in CHB-MIT database

Resource	Total	Used	Utilization
LUT as Logic	53200	5069	9.53%
LUT as Memory	17400	352	2.02%
Flip Flop (FF)	106400	6625	6.23%
Block RAM	140	10.5	7.5%
DSP (DSP48E1)	220	12	5.45%

CosCNN models with 8 convolutional blocks. According to Table 7, the FPGA-based accelerator implementation can achieve approximately 85× and 46× faster than the ARM-based accelerator implementation on models used in the Bonn database and CHB-MIT database, which is considerably efficient. In clinical practice, developing a real-time seizure detection system that can provide detection results for EEG data rapidly is crucial for providing timely intervention. The real-time seizure detection system can complete the data processing and output results within the short interval of data collection. With the FPGA-based accelerator, our system can predict a 4-second multi-channel EEG segment from the CHB-MIT database in only 1.04 seconds, effectively meeting real-time requirements. In contrast, executing the same model on an ARM processor takes 81.14 seconds to predict a single 4-second multi-channel EEG segment, unable to meet the necessary real-time criteria. Moreover, the FPGA-based accelerator requires only 1.94J to predict a 23.6-second single-channel EEG segment from the Bonn database, demonstrating its significant energy efficiency. Similarly, for an 18-channel 4-second EEG segment from the CHB-MIT database, the FPGA-based inference consumes 1.82J, significantly lower than the 81.14J required for inference using an ARM processor. It can be estimated that a battery with a capacity of approximately 2200mAh and a voltage of 5V is sufficient to process the 24-hour EEG data from the CHB-MIT database. These satisfactory results indicate the great potential and feasibility of the proposed CosCNN and its corresponding quantization method applying to low-power real-time portable devices for seizure detection.

Table 8 illustrates the resource utilization of the synthesized circuit in PL for the CosCNN model used in CHB-MIT database. Besides, the total on-chip power of the hardware system is 1.762W, where the static power is 0.144W (8.2%), and the dynamic power is 1.617W (91.8%). For the dynamic power, PS occupies 1.546W (95.6%), and PL occupies 0.071W (4.4%). The above information indicates that the designed circuit is highly efficient in terms of power and resource utilization.

6. Conclusions

In this work, a novel cosine convolutional neural network (CosCNN) is presented to address the limitations of traditional CNNs, particularly in overfitting problem, space complexity, and feature extraction ability, and its effectiveness is demonstrated in seizure detection tasks. By utilizing unique cosine kernels modulated by only two learnable parameters, CosCNN not only significantly reduces memory cost but also improves model performance and interpretability. Its detailed theoretical derivations of forward and backward propagation, and loss function are also provided. In addition, a new quantization method for the CosCNN model is proposed, which promotes the co-design of the algorithm and hardware while reducing the memory access cost of the model without causing accuracy loss. Moreover, a cosine convolution accelerator with the quantized CosCNN is implemented on the low-power Xilinx Zynq Zedboard, achieving a real-time seizure detection system. To evaluate the effectiveness of the CosCNN, two widely used epileptic EEG databases are employed. Results show that the generalization and feature extraction ability of the proposed CosCNN is superior to the traditional CNN, and competitive

performance is yielded on the Bonn database and CHB-MIT database with fewer model parameters and memory costs. Due to the calculation of cosine functions, the training and inference time of a CosCNN is a bit higher than that of a traditional CNN with the same kernel length. In our future work, we will further lower the computational complexity of cosine convolution while reducing memory costs, for example, by utilizing a more efficient implementation of cosine functions and developing an FFT-based fast cosine convolution algorithm. Besides, we will further optimize the hardware implementation of CosCNNs by introducing efficient FFT processors for better meeting the needs for practical use and portable monitoring devices, and apply CosCNNs to EEG-based brain-computer interface (BCI) tasks (Bi & Chu, 2023; Mammone et al., 2023) in low-power scenarios.

Acknowledgements

The support in part by the National Natural Science Foundation of China (No. 62271291), the Key Program of Natural Science Foundation of Shandong Province (No. ZR2020LZH009), the Shenzhen Science and Technology Program (No. GJHZ20220913142607013) and the Natural Science Foundation of Shandong Province (No. ZR2021ZD40, No. ZR2021MF065) was gratefully acknowledged.

References

Abdoli, S., Cardinal, P., & Koerich, A. L. (2019). End-to-end environmental sound classification using a 1D convolutional neural network. *Expert Systems with Applications*, *136*, 252-263.

Acharya, U. R., Oh, S. L., Hagiwara, Y., Tan, J. H., & Adeli, H. (2018). Deep convolutional neural network for the automated detection and diagnosis of seizure using EEG signals. *Computers in biology and medicine*, *100*, 270-278.

Acharya, U. R., Sree, S. V., Swapna, G., Martis, R. J., & Suri, J. S. (2013). Automated EEG analysis of epilepsy: a review. *Knowledge-Based Systems*, *45*, 147-165.

Andrzejak, R. G., Lehnertz, K., Mormann, F., Rieke, C., David, P., & Elger, C. E. (2001). Indications of nonlinear deterministic and finite-dimensional structures in time series of brain electrical activity: Dependence on recording region and brain state. *Physical Review E*, *64*(6), 061907.

Bahr, A., Schneider, M., Francis, M. A., Lehmann, H. M., Barg, I., Buschhoff, A.-S., Wulff, P., Strunskus, T., & Faupel, F. (2021). Epileptic Seizure Detection on an Ultra-Low-Power Embedded RISC-V Processor Using a Convolutional Neural Network. *Biosensors*, *11*(7), 203.

Bhattacharyya, A., & Pachori, R. B. (2017). A Multivariate Approach for Patient-Specific EEG Seizure Detection Using Empirical Wavelet Transform. *Ieee Transactions on Biomedical Engineering*, *64*(9), 2003-2015. https://doi.org/10.1109/tbme.2017.2650259

Bi, J. F., & Chu, M. (2023). TDLNet: Transfer Data Learning Network for Cross-Subject Classification Based on Multiclass Upper Limb Motor Imagery EEG. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, *31*, 3958-3967. https://doi.org/10.1109/Tnsre.2023.3323509

Borra, D., Fantozzi, S., & Magosso, E. (2020). Interpretable and lightweight convolutional neural network for EEG decoding: Application to movement execution and imagination. *Neural Networks*, *129*, 55-74.

Cho, K.-O., & Jang, H.-J. (2020). Comparison of different input modalities and network structures for deep learning-based seizure detection. *Scientific reports*, *10*(1), 1-11.

Elger, C. E., & Hoppe, C. (2018). Diagnostic challenges in epilepsy: seizure under-reporting and seizure detection. *The Lancet Neurology*, *17*(3), 279-288.

Evangelidis, A., & Kugiumtzis, D. (2023). Adaptive Decomposition of Multicomponent Signals and Estimation of Phase Synchronization. *IEEE Transactions on Signal Processing*.

Everitt, B., & Skrondal, A. (2002). *The Cambridge dictionary of statistics* (Vol. 106). Cambridge university press Cambridge.

Faraji, P., & Khodabakhshi, M. B. (2023). CollectiveNet-AltSpec: A collective concurrent CNN architecture of alternate specifications for EEG media perception and emotion tracing aided by multi-domain feature-augmentation. *Neural Networks*.

Faust, O., Acharya, U. R., Adeli, H., & Adeli, A. (2015). Wavelet-based EEG processing for computer-aided seizure detection and epilepsy diagnosis. *Seizure*, *26*, 56-64.

Feng, L., Li, Z., & Wang, Y. (2017). VLSI design of SVM-based seizure detection system with on-chip learning capability. *IEEE transactions on biomedical circuits and systems*, *12*(1), 171-181.

Ficici, C., Telatar, Z., & Erogul, O. (2022). Automated temporal lobe epilepsy and psychogenic nonepileptic seizure patient discrimination from multichannel EEG recordings using DWT based analysis. *Biomedical Signal Processing and Control*, *77*. https://doi.org/ARTN 103755
10.1016/j.bspc.2022.103755

Fisher, R. S., Boas, W. V. E., Blume, W., Elger, C., Genton, P., Lee, P., & Engel Jr, J. (2005). Epileptic seizures and epilepsy: definitions proposed by the International League Against Epilepsy (ILAE) and the International Bureau for Epilepsy (IBE). *Epilepsia*, *46*(4), 470-472.

Glorot, X., Bordes, A., & Bengio, Y. (2011). Deep sparse rectifier neural networks. Proceedings of the fourteenth international conference on artificial intelligence and statistics,

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT press.

Gotman, J. (1982). Automatic recognition of epileptic seizures in the EEG. *Electroencephalography and clinical Neurophysiology*, *54*(5), 530-540.

Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. International conference on machine learning,

Jacob, B., Kligys, S., Chen, B., Zhu, M., Tang, M., Howard, A., Adam, H., & Kalenichenko, D. (2018). Quantization and training of neural networks for efficient integer-arithmetic-only inference. Proceedings of the IEEE conference on computer vision and pattern recognition,

Khan, S., Rahmani, H., Shah, S. A. A., & Bennamoun, M. (2018). A guide to convolutional neural networks for computer vision. *Synthesis Lectures on Computer Vision*, *8*(1), 1-207.

Kingma, D. P., & Ba, J. (2015). Adam: A Method for Stochastic Optimization. ICLR (Poster),

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, *25*, 1097-1105.

Kuhlmann, L., Lehnertz, K., Richardson, M. P., Schelter, B., & Zaveri, H. P. (2018). Seizure prediction—ready for a new era. *Nature Reviews Neurology*, *14*(10), 618-630.

Latka, M., Was, Z., Kozik, A., & West, B. J. (2003). Wavelet analysis of epileptic spikes. *Physical Review E*, *67*(5), 052902.

Lawhern, V. J., Solon, A. J., Waytowich, N. R., Gordon, S. M., Hung, C. P., & Lance, B. J. (2018). EEGNet: a compact convolutional neural network for EEG-based brain–computer interfaces. *Journal of neural engineering*, *15*(5), 056013.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *nature*, *521*(7553), 436-444.

Li, C., Zhou, W., Liu, G., Zhang, Y., Geng, M., Liu, Z., Wang, S., & Shang, W. (2021). Seizure Onset Detection Using Empirical Mode Decomposition and Common Spatial Pattern. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, *29*, 458-467.

Li, H. (2017). Deep learning for natural language processing: advantages and challenges. *National Science Review*.

Li, T., Zhao, Z., Sun, C., Cheng, L., Chen, X., Yan, R., & Gao, R. X. (2021). WaveletKernelNet: An interpretable deep neural network for industrial intelligent diagnosis. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*.

Li, Y., Yu, Z., Chen, Y., Yang, C., Li, Y., Allen Li, X., & Li, B. (2020). Automatic seizure detection using fully convolutional nested lstm. *International journal of neural systems*, *30*(04), 2050019.

Liang, T., Glossner, J., Wang, L., Shi, S., & Zhang, X. (2021). Pruning and quantization for deep neural network acceleration: A survey. *Neurocomputing*, *461*, 370-403.

Liu, C., Jin, J., Daly, I., Li, S., Sun, H., Huang, Y., Wang, X., & Cichocki, A. (2022). SincNet-based Hybrid Neural Network for Motor Imagery EEG Decoding. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*.

Liu, G., Tian, L., & Zhou, W. (2021). Patient-Independent Seizure Detection Based on Channel-Perturbation Convolutional Neural Network and Bidirectional Long Short-Term Memory. *International journal of neural systems*, 2150051.

Liu, G., Zhou, W., & Geng, M. (2020). Automatic seizure detection based on S-transform and deep convolutional neural network. *International Journal of Neural Systems*, *30*(04), 1950024.

Mammone, N., Ieracitano, C., Adeli, H., & Morabito, F. C. (2023). AutoEncoder Filter Bank Common Spatial Patterns to Decode Motor Imagery From EEG. *Ieee Journal of Biomedical and Health Informatics*, *27*(5), 2365-2376. https://doi.org/10.1109/Jbhi.2023.3243698

Migacz, S. (2017). 8-bit inference with tensorrt. GPU technology conference,

Nagel, M., Amjad, R. A., Van Baalen, M., Louizos, C., & Blankevoort, T. (2020). Up or down? adaptive rounding for post-training quantization. International Conference on Machine Learning,

Nahshan, Y., Chmiel, B., Baskin, C., Zheltonozhskii, E., Banner, R., Bronstein, A. M., & Mendelson, A. (2021). Loss aware post-training quantization. *Machine Learning*, *110*(11), 3245-3262.

Noé, P.-G., Parcollet, T., & Morchid, M. (2020). Cgcnn: Complex gabor convolutional neural network on raw speech. ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP),

O'Shea, A., Lightbody, G., Boylan, G., & Temko, A. (2020). Neonatal seizure detection from raw multi-channel EEG using a fully convolutional architecture. *Neural Networks*, *123*, 12-25.

Ozdemir, M. A., Cura, O. K., & Akan, A. (2021). Epileptic eeg classification by using time-frequency images for deep learning. *International journal of neural systems*, 2150026.

Page, A., Sagedy, C., Smith, E., Attaran, N., Oates, T., & Mohsenin, T. (2014). A flexible multichannel EEG feature extractor and classifier for seizure detection. *IEEE Transactions on Circuits and Systems II: Express Briefs*, *62*(2), 109-113.

Priyasad, D., Fernando, T., Denman, S., Sridharan, S., & Fookes, C. (2021). Interpretable Seizure Classification Using Unprocessed EEG With Multi-Channel Attentive Feature Fusion. *IEEE Sensors Journal*, *21*(17), 19186-19197.

Ranzato, M. A., Huang, F. J., Boureau, Y.-L., & LeCun, Y. (2007). Unsupervised learning of invariant feature hierarchies with applications to object recognition. 2007 IEEE conference on computer vision and pattern recognition,

Ravanelli, M., & Bengio, Y. (2018). Speaker recognition from raw waveform with sincnet. 2018 IEEE Spoken Language Technology Workshop (SLT),

Selvakumari, R. S., Mahalakshmi, M., & Prashalee, P. (2019). Patient-Specific Seizure Detection Method using Hybrid Classifier with Optimized Electrodes. *Journal of Medical Systems*, *43*(5), Article 121. https://doi.org/10.1007/s10916-019-1234-4

Shoeb, A. H., & Guttag, J. V. (2010). Application of machine learning to epileptic seizure detection. ICML,

Shoeibi, A., Khodatars, M., Ghassemi, N., Jafari, M., Moridian, P., Alizadehsani, R., Panahiazar, M., Khozeimeh, F., Zare, A., & Hosseini-Nejad, H. (2021). Epileptic seizures detection using deep learning techniques: A review. *International Journal of Environmental Research and Public Health*, *18*(11), 5780.

Sopic, D., Teijeiro, T., Atienza, D., Aminifar, A., & Ryvlin, P. (2022). Personalized seizure signature: An interpretable approach to false alarm reduction for long-term epileptic seizure detection. *Epilepsia*.

Thijs, R. D., Surges, R., O'Brien, T. J., & Sander, J. W. (2019). Epilepsy in adults. *The Lancet*, *393*(10172), 689-701.

Thuwajit, P., Rangpong, P., Sawangjai, P., Autthasan, P., Chaisaen, R., Banluesombatkul, N., Boonchit, P., Tatsaringkansakul, N., Sudhawiyangkul, T., & Wilaiprasitporn, T. (2021). EEGWaveNet: Multi-Scale CNN-Based Spatiotemporal Feature Extraction for EEG Seizure Detection. *IEEE Transactions on Industrial Informatics*.

Truong, N. D., Nguyen, A. D., Kuhlmann, L., Bonyadi, M. R., Yang, J., Ippolito, S., & Kavehei, O. (2018). Integer convolutional neural network for seizure detection. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, *8*(4), 849-857.

Ullah, I., Hussain, M., & Aboalsamh, H. (2018). An automated system for epilepsy detection using EEG brain signals based on deep learning approach. *Expert Systems with Applications*, *107*, 61-71.

Wang, X., Wang, X., Liu, W., Chang, Z., Kärkkäinen, T., & Cong, F. (2021). One dimensional convolutional neural networks for seizure onset detection using long-term scalp and intracranial EEG. *Neurocomputing*, *459*, 212-222.

Wei, Y., Zhou, J., Wang, Y., Liu, Y., Liu, Q., Luo, J., Wang, C., Ren, F., & Huang, L. (2020). A review of algorithm & hardware design for AI-based biomedical applications. *IEEE transactions on biomedical circuits and systems*, *14*(2), 145-163.

World-Health-Organization. (2019). *Epilepsy: a public health imperative* (9241515937).

Yuan, S., Liu, J., Shang, J., Kong, X., Yuan, Q., & Ma, Z. (2018). The earth mover's distance and Bayesian linear discriminant analysis for epileptic seizure detection in scalp EEG. *Biomedical Engineering Letters*, *8*, 373-382.

Zabihi, M., Kiranyaz, S., Jantti, V., Lipping, T., & Gabbouj, M. (2020). Patient-Specific Seizure Detection Using Nonlinear Dynamics and Nullclines. *Ieee Journal of Biomedical and Health Informatics*, *24*(2), 543-555. https://doi.org/10.1109/jbhi.2019.2906400

Zeghidour, N., Teboul, O., Quitry, F. d. C., & Tagliasacchi, M. (2021). LEAF: A Learnable Frontend for Audio Classification. International Conference on Learning Representations,

Zhang, D., Li, H., Xie, J., & Li, D. (2023). MI-DAGSC: A domain adaptation approach incorporating comprehensive information from MI-EEG signals. *Neural Networks*, *167*, 183-198.

Zhang, K., Robinson, N., Lee, S.-W., & Guan, C. (2021). Adaptive transfer learning for EEG motor imagery classification with deep convolutional neural network. *Neural Networks*, *136*, 1-10.

Zhang, Y., Yao, S., Yang, R., Liu, X., Qiu, W., Han, L., Zhou, W., & Shang, W. (2022). Epileptic Seizure Detection Based on Bidirectional Gated Recurrent Unit Network. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, *30*, 135-145.

Zhao, W., Zhao, W., Wang, W., Jiang, X., Zhang, X., Peng, Y., Zhang, B., & Zhang, G. (2020). A novel deep neural network for robust detection of seizures using EEG signals. *Computational and Mathematical Methods in Medicine*, *2020*.

Zhao, W., Zhao, W. B., Wang, W. F., Jiang, X. L., Zhang, X. D., Peng, Y. H., Zhang, B. C., & Zhang, G. K. (2020). A Novel Deep Neural Network for Robust Detection of Seizures Using EEG Signals. *Computational and Mathematical Methods in Medicine*, *2020*, Article 9689821. https://doi.org/10.1155/2020/9689821