A Flexible Software-Hardware Framework for Brain EEG Multiple Artifact Identification

Mohit Khatwani, Hasib-Al Rashid [*], Hirenkumar Paneliya, Mark Horton, Houman Homayoun, Nicholas Waytowich, W. David Hairston, and Tinoosh Mohsenin

Abstract This chapter presents an energy efficient and flexible multichannel Electroencephalogram (EEG) artifact identification network and its hardware using depthwise and separable convolutional neural networks (DS-CNN). EEG signals are recordings of the brain activities. The EEG recordings that are not originated from cerebral activities are termed as artifacts. Our proposed model does not need expert knowledge for feature extraction or pre-processing of EEG data and has a very efficient architecture implementable on mobile devices. The proposed network can be reconfigured for any number of EEG channel and artifact classes. Experiments were done with the proposed model with the goal of maximizing the identification accuracy while minimizing the weight parameters and required number of operations.

Mohit Khatwani
University of Maryland Baltimore County, USA, e-mail: `khatwan1@umbc.edu`

Hasib-Al Rashid
University of Maryland Baltimore County, USA, e-mail: `hrashid1@umbc.edu`

Hirenkumar Paneliya
University of Maryland Baltimore County, USA, e-mail: `hpaneli1@umbc.edu`

Mark Horton
University of Maryland Baltimore County, USA, e-mail: `hmark2@umbc.edu`

Houman Homayoun
University of California, Davis, USA, e-mail: `hhomayoun@ucdavis.edu`

Nicholas Waytowich
Human Research and Engineering Directorate, US Army Research Lab, USA, e-mail: `nicholas.r.waytowich.civ@mail.mil`

W. David Hairston
Human Research and Engineering Directorate, US Army Research Lab, USA, e-mail: `william.d.hairston4.civ@mail.mil`

Tinoosh Mohsenin
University of Maryland Baltimore County, USA, e-mail: `tinoosh@umbc.edu`

[*] corresponding author

Our proposed network achieves 93.14% classification accuracy using EEG dataset collected by a 64 channel BioSemi ActiveTwo headsets, averaged across 17 patients and 10 artifact classes. Our hardware architecture is fully parameterized with number of input channels, filters, depth and data bit-width. The number of processing engines (PE) in the proposed hardware can vary between 1 to 16 providing different latency, throughput, power and energy efficiency measurements. We implement our custom hardware architecture on Xilinx FPGA (Artix-7) which on average consumes 1.4 mJ to 4.7 mJ dynamic energy with different PE configurations. Energy consumption is further reduced by 16.7× implementing on application-specified integrated circuit at the post layout level in 65-nm CMOS technology. Our FPGA implementation is 1.7× to 5.15× higher energy efficient than some previous works. Moreover, our ASIC implementation is also 8.47× to 25.79× higher energy efficient compared to previous works. We also demonstrated that the proposed network is reconfigurable to detect artifacts from another EEG dataset collected in our lab by a 14 channel Emotiv EPOC+ headset and achieved 93.5% accuracy for eye blink artifact detection.

Keywords

EEG, Artifact, Depthwise Separable CNN, FPGA, ASIC, Flexible Reconfigurable Hardware.

Introduction

Electroencephalography is a method of recording non-invasive electrical signals of brain through electrodes. EEG signals can be easily contaminated through noise originating from line electrical noise, muscle movement or ocular movements. These distortions in the EEG signals can be referred to as artifacts. These artifacts can lead to difficulties in extracting underlying neuro information (Iriarte et al 2003; Nuwer 1988).

Artifacts can overlap the EEG signal in spectral as well as temporal domain which turns out to be difficult for simple signal processing to identify artifacts (Islam et al 2016). A method involving regression which subtracts the portion of signal from reference signal was widely used. Problem with this method is that it needs one or more reference channels. As of now, the independent component analysis technique (ICA) is one of the most frequently used method for EEG artifact detection (Jafari et al 2017). The ICA is a denoising technique that involves the whitening of data and separation of linearly mixed sources (Winkler et al 2011; Jung et al 1998). A major drawback of this method is that it is not fully automated and still requires an expert person to label and tag the EEG artifacts. ICA is computationally intensive (Jafari et al 2017) which makes it unsuitable for use in embedded hardware applications.

Convolution neural networks (CNNs) have been successfully used in computer vision tasks such as image and audio classification (Paneliya et al 2020; H.Ren et al 2020, in press; M.Hosseini et al 2020; Hosseini and Mohsenin 2020). Recently, it is also used in reinforcement learning applications (Shiri et al 2020; Prakash et al 2020; Islam et al 2019; Islam and Razi 2019). The advantage of using CNNs in these tasks is that it doesn't need hand crafted features from experts, it learns them automatically using raw data. In (Jafari et al 2019; Zheng et al 2014) authors have shown that time series signals from multimodal sensors can be combined in a 2D images and passed to the convolution layers to learn the features and then passed to Multi-Layer Perceptron (MLP) to perform final classification. One major disadvantage of using CNNs is its high memory and computation requirements.

In this chapter, we use depthwise and separable convolution layers to create memory and computationally efficient CNNs which are used for multiple artifact identification from continuous multi-channel EEG signal. A scalable low power hardware is designed for the optimized model and is implemented both on FPGA and with ASIC post-layout flow.

This chapter makes the following major contributions:

- Propose a scalable depthwise separable CNN based network that can be programmed for any number of EEG channels and artifacts for identification.
- Evaluate and compare proposed model with various other architectures in terms of identification accuracy (multi-class), number of parameters, and total number of computations.
- Perform extensive hyperparameter optimization in terms of number of filters, shape of the filters and data bit-width quantization to reduce the power consumption and memory requirements without affecting the classification accuracy.
- Propose a custom low power hardware architecture which can be configured with different number of processing engines in terms of 2^n where n is ranging from 0 to 3.
- The flexible hardware architecture is parameterized with number of input channels, filters, depth and data-width. Also, it can be of different layers and types.
- Implement proposed hardware using Verilog HDL and synthesized, placed, and routed on low power Xilinx Artix-7-200T FPGA and post layout ASIC in 65-nm CMOS technology, and provide results, analysis and comparison in terms of power consumption, latency and resource utilization.
- Experimentally demonstrated the proposed model with the EEG data collected in our lab by a 14 channel Emotiv headsets

The rest of the chapter is organized as follows: Section presents some related works. Section provides description of EEG data , artifacts and the information on experiments performed to collect it. Moreover, it shows visualization of EEG artifacts as well. Background on different types of convolution layers is given in Section . Section provides details about the proposed artifact identification architecture and classification results. Model optimization and quantization techniques are given in Section . Hardware architecture design is presented in Section . Section provides detailed analysis and results for hardware implementation. Section provides comparison with existing works. Section presents our experiments with the Emotiv EPOC+ headset to collect our own dataset and implement our model for binary classification of eye blink artifact detection. Section concludes the chapter.

Related Work

This section contains a brief description of artifacts that can influence the analysis and the interpretation of EEG recording. It further deals with existing ways for artifact identification. EEG monitors the electrical activity of the brain, signals generated can be used in many applications including seizure detection, and brain-computer interfaces (BCI) (Chander et al 2011)(Page et al 2015a). Some of the electrical activity has rhythmic features while other can be characterized as transient. The bands of frequency for the rhythmic activity are usually alpha (8 - 12 Hz), beta (12 - 30 Hz), delta (1 - 4 Hz) and theta (4 - 7 Hz) waves. EEG signal have very low signal to noise ratio. These signals are usually interfered with artifacts generating from muscle, ocular movements, and power lines.

Artifact is electrical activity with noise which occurs outside and inside of the brain yet is still recorded by the EEG. Essentially an artifact is not of cerebral origin. It can be physiological: originating from the patient's body or extra physiological. The latter can include activity from some of the equipment in the room, electrode pop up and cable movement. Some can be read in global channels, while others can only be found in single channels. Some are recorded as periodic regular events, while others in contrast are extremely irregular.

In order to detect artifact in EEG signal the use of a straightforward signal processing technique is not always the best method for artifact detection. This is mainly due to the fact that artifacts can coincide with EEG signals in both spectral and temporal domains. The main challenge is that both the existence and the actual type of artifact will command the selected process of removal. A traditional way of determining the former and latter is to follow an ICA based procedure as a primary step (Islam et al 2017b). The type of artifact at hand will then determine whether time or frequency domain (or a combination of both) should be used for identification. In our earlier work in (Jafari et al 2017), we proposed an artifact detection technique based on ICA and multi-instance learning classifier. Because of high memory requirements and complex computation, the execution time takes 8.8 seconds on ARM Cortex-A57 processor which is not appropriate for real-time application. In (Page et al 2015a) and (Page et al 2015b), authors compared different machine learning algorithms for designing a reliable and low-power, multi-channel EEG feature extractor and classifier such as K-nearest neighbor (KNN), support vector machine (SVM), naive Bayes, and logistic regression. Among all classifiers, logistic regression has the best average F1 measures of 91%, the smallest area and power footprint, and lowest latency of 0.0018 ms. In (Chou et al 2016), authors proposed a hardware design of automatic muscle artifacts removal system for multi-channel EEG data using blind source separation (BSS) from canonical correlation analysis (CCA) known as BSS-CCA. They show that the BSS-CCA result for eye-blinking and biting is better than automatic artifact removal (AAR) tools. The work in (Bhardwaj et al 2015) proposes a real-time low-complexity and reliable system design methodology to remove blink and muscle artifacts and noise without the need of any extra electrode. The average value of correlation and regression lie above 80% and 67%, respectively. Authors in (Sundaram et al 2016) implemented a moving average and median filter to remove physiological noise from EEG data signal as part of preprocessing. The results show that the median filter consumes slightly less power, and occupies 60% less area, while the moving average filter is 1.2× faster. In (Dutande et al 2018), author proposed a low pass butterworth filter for removing out-band components and adaptive LMS noise canceller for removing in-band components from EEG data signal. In (Mahabub 2018), author proposed a complete filter which is a combination of integrator filter and differentiate filter which support detection of both low and high noises. The total FPGA utilization for complete filter is less than 1%.

In (Islam et al 2017a), authors have used feature extraction and traditional machine learning classifiers such as KNN and SVM to build a fully automated EEG artifact classifier. This method outperforms the ICA based methods, exhibiting lower computation and memory requirements. Proposed architecture is also implemented on embedded ARM Cortex CPU. On average, it consumes 1.5 W power at 1.2 GHz frequency.

In (Barachant et al 2010), Riemannian geometry is used to propose a framework for classification in BCI applications. In this approach, the classification task is performed by calculating the covariance matrices of the given input epoch without performing any pre-processing. This algorithm relies on mean computation of covariance matrices which is obtained by mapping the dataset into tangential space, making computation in small embedded systems for real time applications. Deep neural networks require a lot of data by performing data augmentation. The use of deep neural networks have grown due to their success in image classification problems (Krizhevsky et al 2012). In (Schetinin and Schult 2004), authors have used feed forward neural network combined with decision tree to detect ocular artifacts in EEG signal. Authors in (Majidov and Whangbo 2019) overcome the problem mentioned in (Barachant et al 2010) by using CNN for their classification task. Convolution neural networks have been also used in (Khatwani et al 2018) for detecting ocular and muscular related artifacts. One disadvantage of using CNN is its high memory and computation requirements. Another method which accomplishes desired results is that of, recurrent neural networks (RNNs). The long short-term memory (LSTM) approach was proposed in (Wang et al 2018) as a variance of RNN-based EEG classifier in motor imaginary task. In (Abdelhameed et al 2018), authors proposed a deep convolutional bidirectional LSTM based classifier for epileptic seizure detection. However, both RNN and LSTM require memory bandwidth-bound computation which is not hardware friendly and therefore restricts the applicability of these neural network solutions. LSTM structure comprises four separate linear layers per unit to run at and for every time-step sequence. Those linear layers require multiple memory units which might not be efficient for hardware design. Therefore, hardware implementation of RNN/LSTM and its variances are not good contender to be implemented as energy efficient hardware.

Depthwise and separable convolution layers can be used to reduce the weight parameters. This can lead to increase in efficiency without decreasing performance. Use of depthwise separable convolution was also demonstrated in the first layer of Inception-v4 (Szegedy et al 2017). The use of Xception model on ImageNet dataset led to a small change in classification performance with large improvement in computational requirements (Chollet 2017).

Our proposed model presents an energy efficient architecture with lower number of weight parameters and computations which enables both detection and identification of multiple artifact. Use of depthwise and separable convolution layers decouples the mapping of cross-channel and spatial correlations, leading to reduction in number of required parameters and computation.

EEG Artifacts and Visualization

In order to assess and evaluate the accuracy of our model, we used a previously recorded EEG dataset. The data was collected based on the experiments in which participants manually performed a series of different ocular or muscular artifacts (i.e. jaw clenching, eye blinking, etc.). The EEG data was recorded using a 64 channel BioSemi ActiveTwo system with a sampling rate of 512Hz and compared to the two mastoids average. Four different channels were used to monitor eye motions by EOG. EOG behavior was documented in order to validate the EEG instances of eye blinks and saccades but was not included in the subsequent experiments. The usage of EEG channels alone allows simple implementation of the model. The data were down-sampled to 256 Hz using a discrete wavelet transform to reduce the computing strain and also to extend the analytical frequency spectrum. The data then high-pass filtered at 1 Hz using a 8 order IIR Butterworth filter. EEGLAB was used to process the data and ERPLAB was used to filter the data. Participants were required to perform a series of noise-inducing body, facial, head or eye movements, which were gathered as part of a larger study (Lawhern et al 2012). The list of movements were reviewed before starting the experiment so that every patient is familiar with it.

It was up to the participants to determine the precise choreography of each movement and to perform movements which felt more natural to them. Each movement was performed as a separate set of 20 repetitions. A screen was put in place in order to remind the participants of the movement they should make. A male voice initially counted down from 3 at a rate of every 2 seconds followed by a tone every 2 seconds. This procedure was done for each set. The participants would make the movements in time with the vocal commands. They were advised to perform the tasks in the first second of the 2 seconds period and to relax in the remaining 1 seconds. Additionally, each participant performed a baseline recording session where they were instructed to keep still and look straight at a blank computer screen for around 8 seconds at the start of every run. EEG data from this baseline session was used as ”clean” (or artifact-free) data. Artifacts considered are clenching jaw (CJ), move jaw (MJ), blink eyes (BE), move eyes leftwards (EL), move eyes rightwards (ER), raise eyebrows (RE), rotate head (RH), shrugging shoulders (SS) and rotate torso (RT). Table 1 gives a brief description of nine artifacts which were performed by every patient. Since participants were instructed to conduct the action in a normal manner, heterogeneity was observed among subjects in the movement performance latencies. For examples, some participants waited for the audio tone to execute the operation, resulting in a response time interval of 300–400 ms while other participants sought to anticipate the audio tone, resulting in certain time periods that did not include an artifact attributable to conducting the operation too early. As a consequence, the particular timestamp timing details for each individual was changed such that the time-course of the artifact was present in the epoch. Around 100 samples are generated for each artifact class as well as clean signal. EEG timestamps of size 64×512 is used as input both from artifact and artifact-free signal with different step size.

Table 1 Description on nine artifacts performed by every patient.

Artifact Code	Artifact Description
101	Clench Jaw
102	Move Jaw Vertically
103	Blink Eyes
104	Move Eyes Leftward
105	Move Eyes Rightward
106	Raise Eyebrows
107	Rotate Head
108	Shrug Shoulders
109	Rotate Torso

Figure 1 shows plot for first 20 of 64 electrodes placed on the scalp to capture the EEG signals. This can be useful to inspect which electrodes are significant in capturing the specified artifacts. This plot shows nine artifacts for single patient. Every artifact generates a different pattern which helps in identifying the specified artifact. Vertical lines indicate the instant at which event has occurred. There may be differences in signals before vertical line which may occur due to noise or external sources. The location of vertical lines are adjusted so that it can correctly capture data which relates to particular artifact event.

Figure 2(a) shows the position of 64 electrodes used for capturing the EEG data. Figure 2(b) and Figure 2(c) show the topographical plot for, respectively the artifact 101 (clenching jaw) and artifact 103 (Eye blink). Figure 2(c) clearly shows that the electrodes placed in the front part of EEG are more significant in identifying ocular related artifacts.

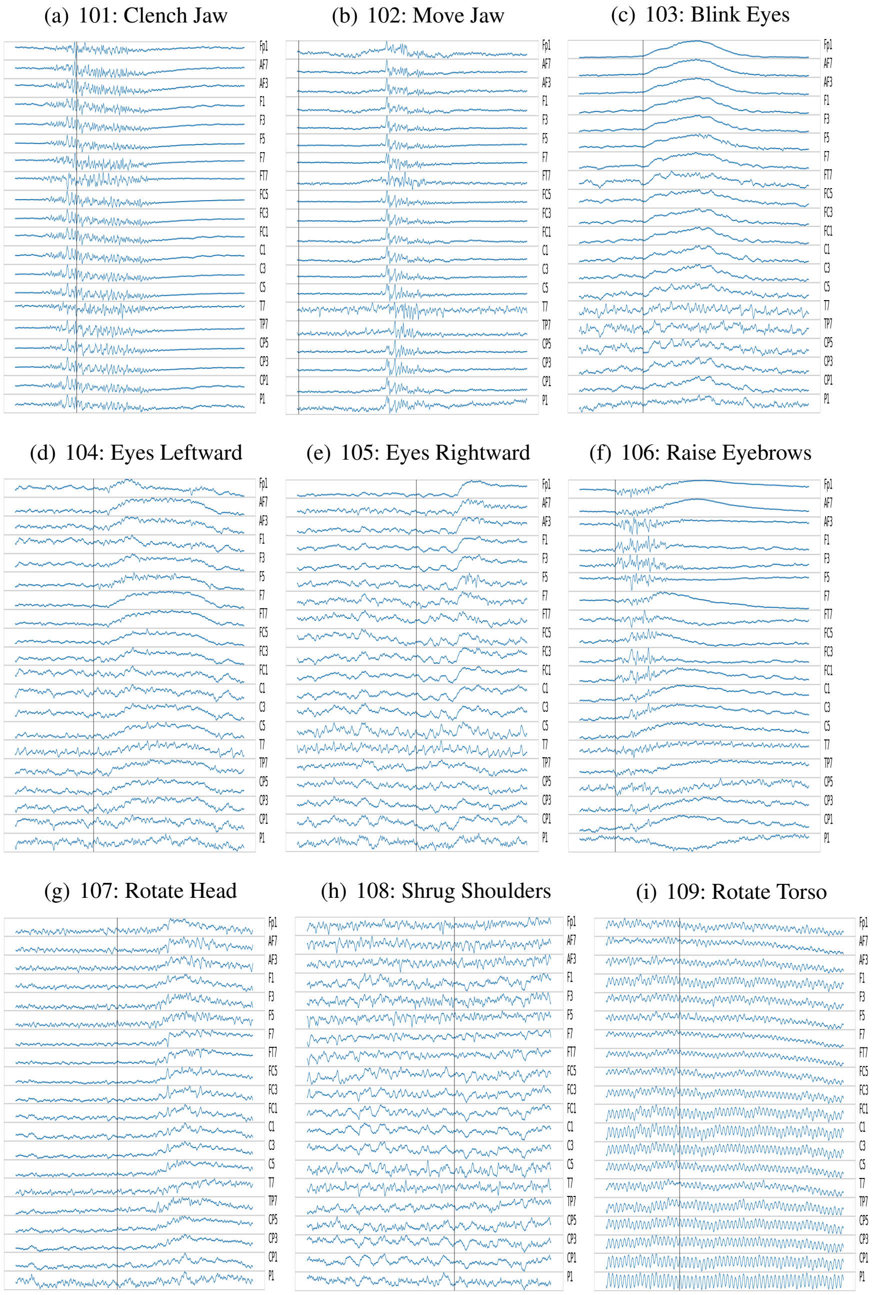
(a) 101: Clench Jaw
(b) 102: Move Jaw
(c) 103: Blink Eyes
(d) 104: Eyes Leftward
(e) 105: Eyes Rightward
(f) 106: Raise Eyebrows
(g) 107: Rotate Head
(h) 108: Shrug Shoulders
(i) 109: Rotate Torso

Fig. 1 Visualization of nine artifacts performed by patients. Instructions were given to patients every two seconds and it was advisable to perform the task in the first second. Vertical line indicates the start of experiment.

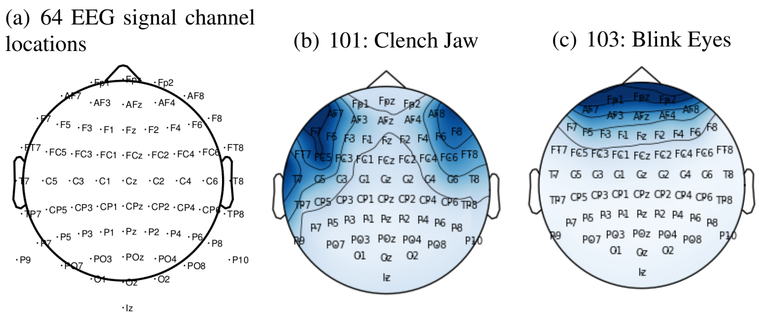
(a) 64 EEG signal channel locations
(b) 101: Clench Jaw
(c) 103: Blink Eyes

Fig. 2 (a) shows locations of 64 EEG electrodes, (b) and (c) show topographical plot for the artifact 101 (Clenching jaw) and 103 (Blinking eyes), respectively.

Theoretical Background

Traditional Convolution

Figure 3 presents the conventions for the traditional convolution. In traditional convolution layer if the input is of size $D_f \times D_f \times M$ and N is the number of filters applied to this input of size $D_k \times D_k \times M$ then output of this layer without zero padding applied is of size $D_p \times D_p \times M$. If the stride for the convolution is S then D_p is determined by the following equation:

$$D_p = \frac{D_f - D_k}{S} + 1 \tag{1}$$

In this layer, the filter convolves over the input by performing element wise multiplication and summing all the values. A very important note is that depth of the filter is always same as depth of the input given to this layer. The computational cost for traditional convolution layer is $M \times D_k^2 \times D_p^2 \times N$ (Howard et al 2017).

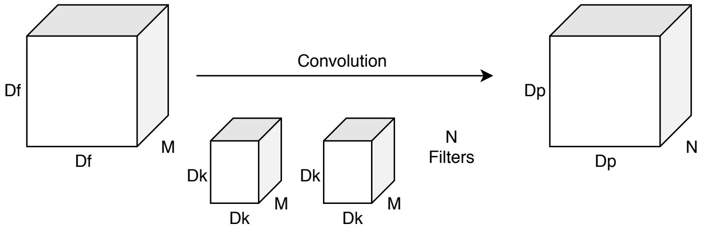

Fig. 3 Traditional convolution layer with the input shape of $D_f \times D_f \times M$ and output shape of $D_p \times D_p \times N$.

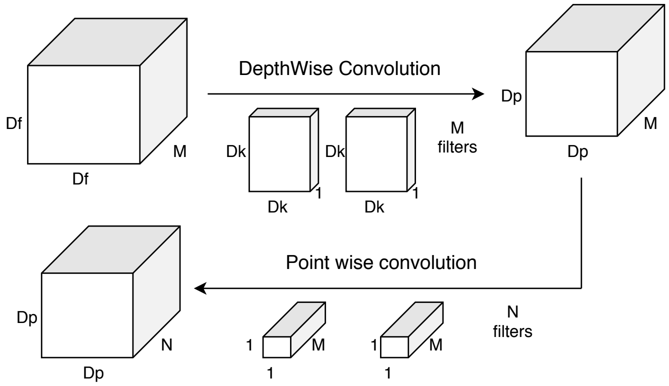

Fig. 4 Depthwise separable convolution layer which is a combination of depthwise convolution and pointwise convolution.

DepthWise Convolution

Figure 4 presents the conventions for the depthwise convolution. For every input of size $D_f \times D_f \times M$ we have M filters of shape $D_k \times D_k$ and depth 1. $D \times M$ filters are used in depthwise convolution where D is the depth multiplier. As every input channel in depthwise convolution has a separate filter, the overall computational cost is $M \times D_k^2 \times D_p^2$ which is $M\times$ less than with traditional convolution (Howard et al 2017)(Chollet 2017).

Table 2 Number of parameter and required computations equations for different types of convolution layers.

Convolution Layers	Parameters	No. of Computations
Traditional	$M \times D_k^2 \times N$	$M \times D_k^2 \times D_p^2 \times N$
Depthwise	$M \times D_k^2$	$M \times D_k^2 \times D_p^2$
Depthwise Separable	$M \times D_k^2 + M \times N$	$M \times D_p^2 \times D_k^2 + M \times D_p^2 \times N$

Depthwise Separable Convolution

Depthwise Separable convolution is a combination of depthwise and pointwise convolution (Kaiser et al 2017). In depthwise operation, convolution is applied to a single channel at a time unlike standard CNN's in which it is done for all the M channels. So here the filters/kernels will be of size $D_k \times D_k \times 1$. Given there are M channels in the input data, then M such filters are required. Output will be of size $D_p \times D_p \times M$. A single convolution operation require $D_k \times D_k$ multiplications. Since the filter are slided by $D_p \times D_p$ times across all the M channels. The total number of computation for one depthwise convolution comes to be $M \times D_p^2 \times D_k^2$.

In point-wise operation, a 1×1 convolution is applied on the M channels. So the filter size for this operation will be $1 \times 1 \times M$. If we use N such filters, the output size becomes $D_p \times D_p \times N$. A single convolution operation in this requires $1 \times M$ multiplications. The total number of operations for one pointwise convolution operation is $M \times D_p^2 \times N$. Therefore, total computational cost of one depthwise separable convolution is $M \times D_p^2 \times D_k^2 + M \times D_p^2 \times N$ (Howard et al 2017).

Table 2 summarizes the equations for parameters and number of computations for different convolution layers. Here $D_k \times D_k$ is the size of the filter, $D_p \times D_p$ is the size of the output, M is number the of input channels and N is the number of output channels.

Proposed Network Architecture and Results

EEG Artifact Identification Model Architecture

Figure 5 shows the architecture of the proposed model. It consist of one traditional convolution layer, one depthwise convolution layer, one depthwise separable convolution layer and one softmax layer that is equivalent in size to the number of class labels. Average pooling is applied twice, once after depthwise convolution, another one after depthwise separable convolution. The complete model architecture including number of filters, filter shapes, data bit precision level are chosen based on an extensive hyperparameter optimization process which is discussed in Section .

The first CNN layer operates on the raw EEG data so that it can learn to extract the necessary features for artifact identification. However, DC offset was removed such that the EEG signals are centered around zero. The EEG epochs of size 64×512 is passed to the first 2-d convolution layer consisting 16 filters of size 64×4. This ensures that adequate spatial filter is learned in the first layer. Zero padding is avoided to avoid large computations. After traditional convolution, a depthwise convolution is used with filter size of 1×32 and depth multiplier of 1 which means there will be 1 filters associated with each depth. This is followed by an average pooling layer with pool size of 1×32. A separable convolution is further used with 1×32 filter size which is again followed by an average pooling layer with pool size of 1×8. All layers are followed by a rectified linear unit (ReLU) activation function. Once these convolution operations have been performed, the output from the last average pooling layer is flattened into a single vector so that a fully connected layer can be added. Only one fully connected layer is employed in the model which has ten nodes with Softmax activation for 10-class classification application. A Fully connected layer before the Softmax layer is avoided to reduce the number of parameters.

The weights for each of the layers of the network are initialized from a normal distribution. The network is trained using the Adam optimization method and a learning rate of 0.001. Categorical cross-entropy is used as the loss function. In total, the network utilizes 5,546 parameters and 4.69 million operations for processing the input frame.

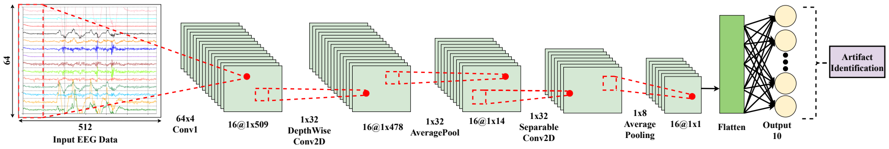

Fig. 5 Proposed architecture which uses combination of depth-wise and separable convolution layers. A total of 5,546 parameters is required for this architecture.

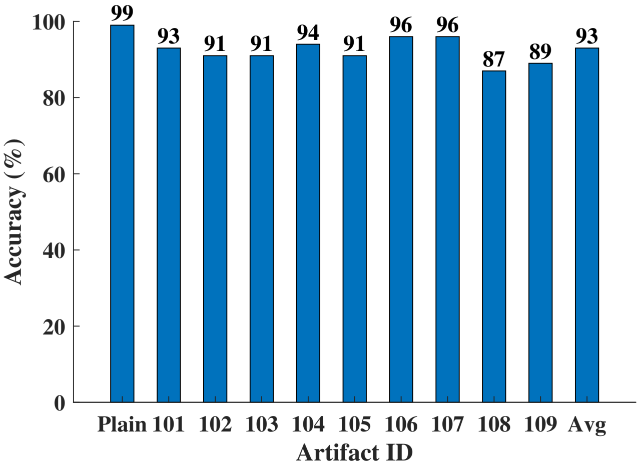

Fig. 6 Class-wise and average accuracy for proposed model

Classification Analysis and Results

Our model architecture is evaluated for seventeen patients for nine different artifacts. Our model is trained and tested using intra-patient setting. The model is trained using 70% of the data, 10% is used for validation, and the remaining 20% is used for testing. All the ten classes are balanced for classification task.

Figure 6 shows class-wise accuracy of the proposed model. It can be seen that our proposed model identifies all nine artifacts with average accuracy of 93%. The accuracy ranges between 87% and 99%. It can be concluded that muscle related artifacts such as the shrugging shoulders (108) and rotating torso (109) are more difficult to identify as compared to other artifacts. From figure 1 it is clearly seen that the EEG signals for shrugging shoulders has similarities between the both side of the vertical line so that it's identification accuracy is the lowest one among all other artifacts. The artifact with best in-class accuracy for all the models is the raising eyebrows (106) and rotating head (107) which exhibit 96% accuracy.

Table 3 Comparison of parameters, computations and average accuracy (17 patients and 10 classes which includes 9 artifacts and 1 plain signal) of different model configurations. All the models classify 9 different artifacts with test data and training data for the same patient

Model	Accuracy (%)	Parameter	# Computations (Millions)
CNN (Khatwani et al 2018)	80.37	24,842	35.4
EEGNet (Lawhern et al 2018)	95.30	4,394	135.3
This work	**93.14**	**5,546**	**4.7**

Comparison of Classification Accuracy with Existing Work

We compare our model with the previous works (Lawhern et al 2012; Islam et al 2017a; Khatwani et al 2018; Lawhern et al 2018) in terms of accuracy, number of parameters, and computation cost. In (Lawhern et al 2012), the auto-regressive (AR) model for artifact detection can be considered as a baseline model but it achieves 68.42% classification accuracy. Whereas, comparing the results reported in (Islam et al 2017a), it can be said that the simple linear machine learning approaches have less accuracy compared to deep learning methods. The authors in (Islam et al 2017a) reported that KNN has average classification accuracy of 78.8%, Logistic Regression (LR) has average classification accuracy 52.6% and Support Vector Machine (SVM) has 53.3% average accuracy for detecting the artifacts from EEG signals. The extensive comparative results among the models mentioned in (Khatwani et al 2018; Lawhern et al 2018) and ours to identify EEG artifacts are presented in Table 3. In (Khatwani et al 2018), two convolution layers are followed by two maxpooling layers to detect the EEG artifacts using the same dataset. We run the same model to identify multiple EEG artifacts. This model results in the overall average accuracy 80.37% with 24,842 parameters and 35.4 million computations. In (Lawhern et al 2018), one convolution layer is used with one depthwise and one separable convolution layer. We run the model in (Lawhern et al 2018) to identify multiple EEG artifacts. This model results in the average accuracy of 95.30% with 4,394 parameters and 135.31 millions of computation. The main differences between figure 7 and figure 5 is the shape of the filters in first layer. After changing shape from horizontal to vertical i.e. 1×64 to 64×4, the computation in the first layer is decreased significantly. Our proposed model achieves the overall average accuracy of 93.13% with 5,546 parameters and 4.69 million computations. Although EEGNet (Lawhern et al 2018) outperforms our proposed model in terms of accuracy and model parameters, our proposed model shows a significant reduction in number of computations, yielding a more hardware friendly solution. Since we use the EEGNet like model to identify multiple artifacts, we also presented a layer-wise comparison in Table 4 to show the improvements that our model yields in terms of number of computations, justifying model design. Based on the results shown in Table 4, the number of computations in each layer is significantly reduce with our proposed model. In conv1 layer, EEGNet has $134.21 millions$ computation while our proposed model has $4.1 millions$ computations. Thus, the computation in that layer is reduced with the proposed method by $32.18\times$ as compared to EEGNet. In total, our proposed model reduces the number of computations by a factor of 28.81 as compared to EEGNet.

Table 4 Comparison of computation in each layer for EEGNet (Lawhern et al 2018) and the architecture of this proposed model.

Layers	Computation in EEGNet (Lawhern et al 2018)	Computation in this work	Reduction of computation
Conv1	134,217,728	4,169,728	32.18 ×
DepthwiseConv2D	1,048,576	489,472	2.14 ×
AveragePooling2D	16,384	14,336	1.14 ×
SeparableConv2D	32,768	21,504	1.52 ×
AveragePooling2D	1,024	256	4.00 ×
Output	1,280	320	4.00 ×
Total	135,317,760	4,695,616	**28.81** ×

Model Optimization for Embedded Low-Power Hardware

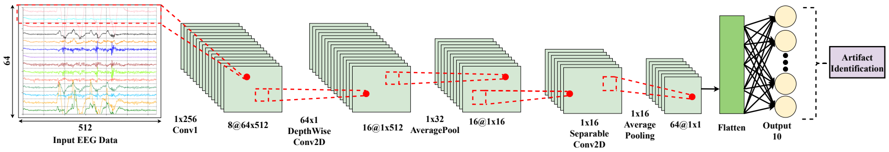

Fig. 7 Original EEGNet architecture which uses combination of depthwise and separable convolution layers. Total parameters required for this architecture is 4,394.

From the discussion of Section , it can be seen that the network architecture consists of one traditional convolution layer, one depth-wise convolution layer, one separable convolution layer, and two average pooling layers. In this section, we explain the reason for choosing the network architecture and parameters. To deploy our network at low powered and small IoT and wearable devices, we have done multiple experiments to optimize the model. Extensive hyperparameter optimization has been executed to reduce the memory requirements, hardware complexity, and power consumption while maintaining high detection accuracy. In this section, we will specifically explore the impact of changing network parameters and quantization on our model accuracy.

Network Parameters Optimization

The number of the filters, the shape of the filters, and the size of the pooling layers are important hyperparameter which affect the memory requirements and number of the computations required to finish a classification task. The number of the computations directly influences on the energy consumption. We experimented with different configurations of our model including different number of filters (F_1) for the first convolution layer and the number of the spatial filters for each temporal filter(i.e. the depth multiplier (D) of the depth-wise convolution layer). We set the number of the filters (F_2) for the separable layer as $F_1 \times D$. Table 5 shows six different configurations with 8, 16 and 32 filters for the first convolution layer and the multiplier depth of 1 and 2. Considering optimum number of parameters and number of computations without compromising the accuracy value much, we got 93.13% of average accuracy with 16 filters for the first convolution layer and 1 as depth multiplier. Figure 8 shows six different sets of configurations where filter height for the first layer of convolution is kept constant at 64 and values for filter width changes to 4, 8, 16. We experimented with two different sizes of the filters for depthwise and depthwise separable convolution layers, (1×16) and (1×32). We kept the number of the filters for the first convolution layer and depth multiplier fixed as previous selection. Experimenting with these different sets, we selected Set 2 for our network configuration as it gives the optimum parameters and number of calculations without compromising the identification accuracy. Table 6 shows the experimental results for the different configurations mentioned earlier.

Table 5 Impact of number of filters in first convolution layer and depth multiplier on the classification accuracy, model parameters and number of computations.

(F_1, D)	Accuracy (%)	Parameters	# Computations (Millions)
(8,1)	90.30	2,714	2.34
(8,2)	91.8	3,498	2.61
(16,1)	**93.13**	**5,546**	**4.69**
(16,2)	93.46	7,498	5.23
(32,1)	94.43	11,549	9.40
(32,2)	94.73	17,034	10.52

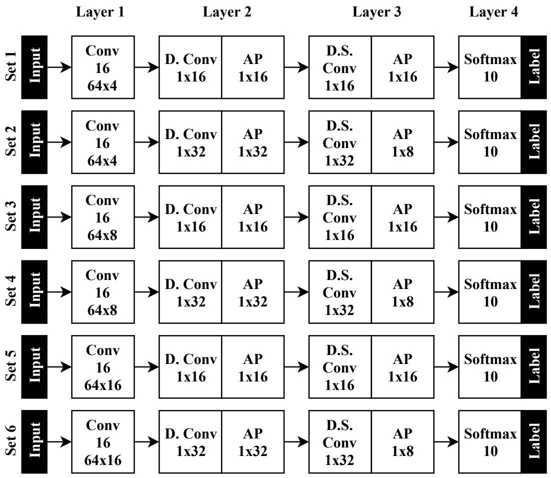

Fig. 8 Six different sets of configurations showing different filter shape shapes for different convolution layers and size for average pooling layers

Table 6 Impact of filter sizes for different convolution layers on the classification accuracy, model parameters and number of computations.

Sets	Accuracy (%)	Parameters	# Computation (Millions)
Set 1	92.08	5,034	4.46
Set 2	**93.13**	**5,546**	**4.69**
Set 3	92.59	9,130	8.57
Set 4	93.37	9,642	8.76
Set 5	92.86	17,322	16.57
Set 6	92.11	17,834	16.79

Model Weights Quantization

Quantizing model weights is a popular method to reduce the model size. Quantization reduces the complexity of the model by reducing the precision requirements for the weights. Cache is reused in a more efficient way with the lower precision weights. Quantization is also power efficient since the low precision data movement is more efficient than the higher precision data (Krishnamoorthi 2018). Therefore, weight quantization with 16 bits and 8 bits is performed on our model. Based on the results shown in Figure 9 8, the 16 bit precision model has nearly same average accuracy as the full precision model. However, 8 bit precision model has only 2% (91.53%) drop in average accuracy from the full precision model. To design our hardware architecture we have chosen the 8 bit quantized model.

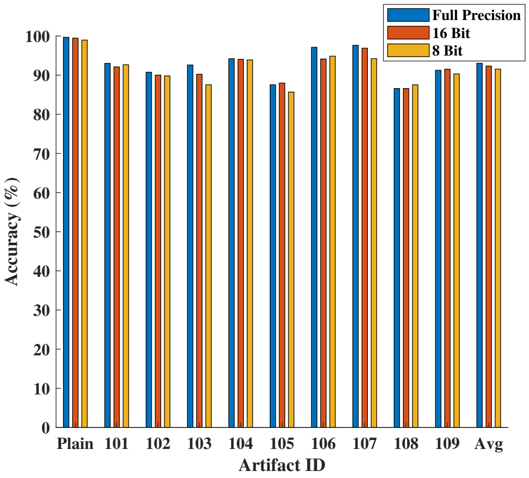

Fig. 9 Impact of model quantizations on the model accuracy. 16 bit quantized model gives same accurate results as the full precision model whereas 8 bit quantized model gets 2% average accuracy drop from full precision model

Hardware Architecture Design

Figure 10 shows the block diagram of the hardware design with implementation details for the proposed architecture. The primary objectives for the hardware architecture design are: consume minimal power, occupy small area, meet latency requirements, require low memory and need to be fully configurable. This design can be configured of doing all type of convolution layers mentioned in Section by changing in the state machine and parameters of the design such as input size, filter size, type of convolution, depth of input and filter, number of filters and size of softmax can be design according to prerequisites. According to Figure 10, the architecture design comprises of one shared filter memory, one shared feature map memory, convolution block, average pooling block and softmax block which are explained below.

1. **Convolution** performs a convolution operation with ReLU activation logic. It can configure up to 2^n processing engines (PEs).
2. **Average Pooling** block performs average operation in a window
3. **SoftMax** performs fully-connected layer operations that includes ReLU and softmax activation function.
4. **Filter Memory and Feature Memory** stores the weights and input data of the model architecture.

The convolution block presented in Figure 10 is using single entity of each adder, multiplier, small filter memory, input feature memory, output feature map, multiplexer and state machine block. We used the 8-bit data-path in our design and as per the requirements, we used the larger data-path after multiplication and addition

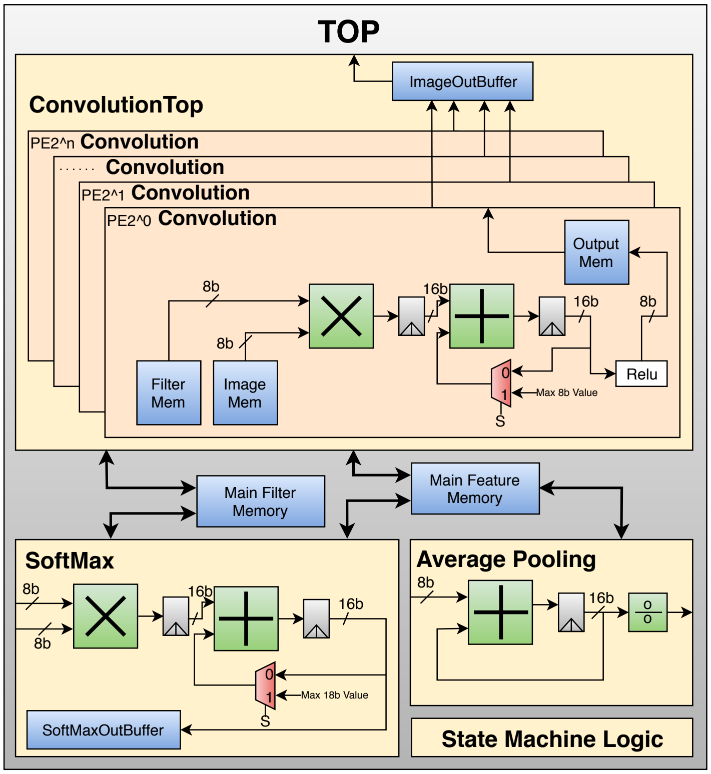

Fig. 10 Block diagram of hardware architecture used to implement the proposed model. The hardware architecture includes a top-level state-machine which controls the ConvolutionTop, AveragePooling and SoftMax blocks as well as all memory blocks. PE refers to number of convolution Processing Elements that process in parallel, and n is in the range from 0 to 3.

operation. We used tensorflow to train our model offline on a standard machine. We converted 32-bit floating point values to 8-bit fixed values to increase the computational efficiency. Floating-point arithmetic is complex in hardware and requires more resources, execution time and power. EEG data is then passed from the main feature memory to the convolution and ReLU activation function through the convolution block. The ReLU activation function output is truncated to 8-bit and stored in the output memory of the output feature and then stored in the main memory of the feature. This data is then passed to the average pooling containing registers, adder and divider as input. The results are stored in the main feature memory after average pooling. Finally, the fully connected one that is used in this work only in the last layer of the neural network. It consists of one multiplier, one adder, few registers, one multiplexer, and one SoftMaxOutBuffer memory to store the neurons of the output. After finishing the computation of the softmax layer, the results are stored in the main feature memory that overwrites previous outdated intermediate data.

Hardware Implementation and Results

FPGA Implementation Results and Analysis

The complete proposed model is implemented on a Xilinx Artix-7 FPGA which includes convolution, average-pooling and fully-connected layers. We used a Verilog HDL to describe the hardware architecture.

Figure 11 shows the power consumption breakdown of post-place and route implementation on the FPGA, which is obtained by using vivado power tool. As it can be seen from the figure, average device block ram power consumption of FPGA is around 87% of total dynamic power which is significantly larger when compared to the logic power. However, overall power and energy are 5.2× 3.0× respectively smaller compared to the previous work (Khatwani et al 2018).

Table 7 provides the implementation results for 1PE, 2PE, 4PE and 8PE. The result shows that the minimum amount of energy consumed by 8 PEs at operating frequency of 21.5 MHz. Figure 12 represents the power consumption, energy and latency with increasing number of PEs. From Figure 12, we show that increasing number of PEs leads to the increase in power consumption and decrease in latency, which leads to decrease in overall energy consumption.

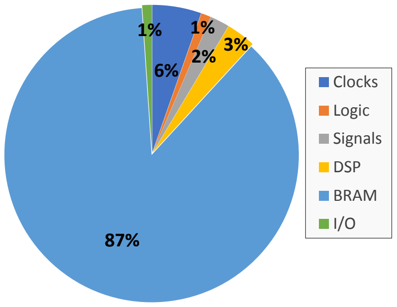

Fig. 11 Breakdown of dynamic power consumption of the design implemented on FPGA

ASIC Implementation Results and Analysis

To reduce the overall power consumption, an **A**pplication-**S**pecified **I**ntegrated **C**ircuit for proposed architecture is implemented at the post-layout level in 65-nm CMOS technology with 1.1-V power supply. A standard-cell register-transfer level (RTL) to Graphic Data System (GDSII) flow using synthesis and automatic place

Table 7 Implementation results on Xilinx Artix-7 FPGA with different number of PEs.

	Config. 1	Config. 2	Config. 3	Config. 4
No. of PEs	1	2	4	8
Frequency (MHz)	37.7	35.2	34.2	30.7
Latency (ms)	86.7	47.2	25.2	15.1
Dynamic Power (mW)	54	58	75	96
Dynamic Energy (mJ)	4.7	2.7	1.9	1.4
No. of Slices	4210	4997	6438	8412
No. of BRAM	149	165	198	264
No. of DSP	24	28	32	102

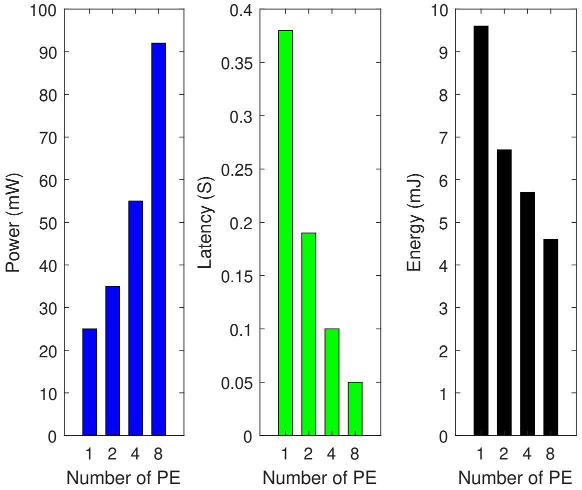

Fig. 12 Implementation results of power, energy and latency with different number of PEs

and route is used. The proposed model including convolution, average-pooling and fully-connected with activation function is implemented using Verilog HDL to describe the architecture, synthesized and placed and routed using RTL compiler and Encounter.

The ASIC layout of the proposed model contains three level of hierarchy as shown in Figure 13. The lower level of hierarchy is Convolution block which contain three memory ImageBuffer, filterBuffer and outputBuffer, and logic for convolution. The size of ImageBuffer, filterBuffer and outputBuffer are 32K, 1K and 32K bytes respectively. The next level of hierarchy is ConvolutionTop block which contain 8 Convolution block, 1 data memory size of 256K bytes and state machine logic. As ARM library can generate maximum 32K bytes size memory, we made 256K bytes size of data memory using 8 of 32K bytes size memories. The highest level of hierarchy is Top module which contains one ConvolutionTop block, one data memory size of 256K bytes, one filter memory size of 9K bytes, average-pooling block, soft-max block and state-machine logic for data transfer between each layer. The specified filter memory is smallest available memory which is sufficient to store 9,194 filter

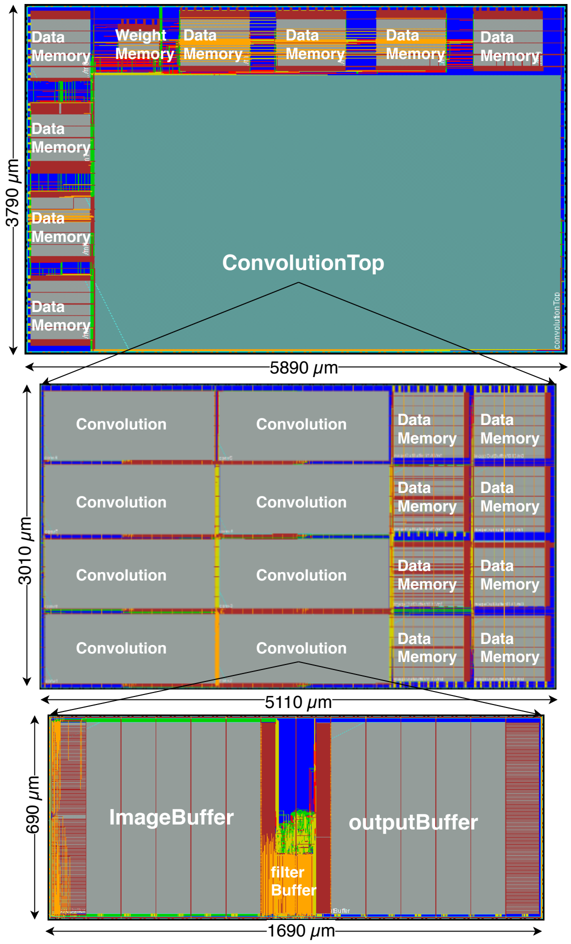

Fig. 13 Post-layout view of proposed architecture with 8 PEs ASIC implementation in 65 nm, TSMC CMOS technology with operating frequency of 100 MHz

values. Because of limitation of arm memory library, we combine 8 of 32K bytes size memories to create one 256K bytes size data memory.

Table 8 shows the comparison between implementations of proposed hardware architecture on FPGA and ASIC. As it can be seen from table, ASIC implementation achieves lowest power and energy consumption which is 3.8× and 16.7× less, respectively compared to the FPGA implementation.

Table 8 Comparison of different parameters between FPGA and ASIC at the post-layout level in 65-nm, TSMC CMOS technology

Hardware	FPGA	ASIC	Improvement
Technology	28 nm	65 nm	-
Voltage (V)	1.0	1.1	-
Frequency (MHz)	30.7	100	3.3×
Latency (ms)	15.1	4.6	3.3×
Throughput (label/s)	66.2	216.2	3.3×
Power at 21 MHz (mW)	234	61.2	3.8×
Energy (mJ)	4.7	0.3	16.7×

Comparison with Existing Work

Table 9 presents a comparative results of the proposed hardware implementation results with existing state-of-the-art implementation with same or related physiological dataset on embedded devices. Authors in (Khatwani et al 2018; Rashid et al 2020) proposed their work based on the same dataset whereas authors in (Zhang et al 2015) proposed their work on popular UCF101 - action recognition data set. When our proposed hardware model is deployed at Xilinx FPGA device with a fully parallel design and running at 30.7 MHz, it consumes 4.7 mJ energy. Authors in (Khatwani et al 2018) reported that their CNN implementation consumes 35 mJ energy with the same dataset. Our Depthwise separable CNN shows 7.44× improvement from their implementation. Authors in (Rashid et al 2020) reported theirs consumed energy as 0.021 mJ which is very lower compared to our implementation although we are using same dataset. They have used LSTM based neural network model to binary classify the EEG artifacts which is the reason of their less consumed energy. However, our Depthwise separable CNN based EEG artifact identification outperforms their LSTM based EEG artifact detection in terms of energy efficiency by 11.74× which is very promising. As authors in (Zhang et al 2015) presented their CNN hardware architecture on video dataset, it requires more power and energy compared to our EEG based depthwise separable CNN hardware architecture. However, as the energy efficiency is most common base to compare different hardware architectures, our proposed FPGA hardware implementation has 5.87 GOP/s/w which outperforms previous implementations in (Zhang et al 2015) in terms of energy efficiency. The ASIC implementation further shows more energy efficiency with 58.82× highest improvement over the previous implementations.

Experimental Study: Eye Blink Artifact Detection Using Emotiv EPOC+ Headset

To demonstrate the real time EEG artifact detection with our current model, we used Emotiv EPOC+ headset. It is high resolution 14-channel EEG system. The bandwidth for this system is 0.2-45Hz. There are digital notch filters at 50 Hz and

Table 9 Comparison of this work with previous work implemented on FPGA with Config. 4

	(Fan et al 2019)	(Rashid et al 2020)	(Khatwani et al 2018)	**This Work**	
Application	Human Activity Recognition	EEG Artifact Detection	EEG Artifact Detection	EEG Artifact Identification	
Platform	Arria 10 SX660	Artix7 100t	Artix7 200t	Artix7 200t	TSMC 65 nm
Frequency (MHz)	150	52.6	37.4	30.7	100
Latency (ms)	35.3	1.2	200	15.1	11.1
Power (mW)	36000	109	194	234	61.2
Energy (mJ)	1270	0.021	35	4.7	3.1
Energy Efficiency (GOP/s/W)	1.47	0.5	3.47	5.87	29.41

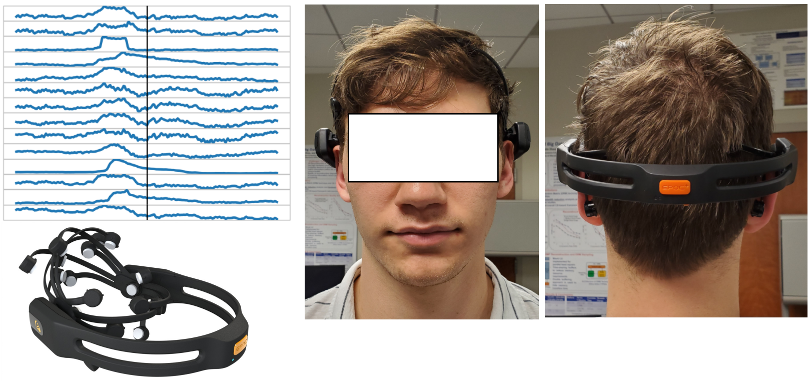

Fig. 14 Model deployment and eye blink artifact detection experiment with the Emotiv EPOC+ 14 channel headset capturing EEG data. Eye blink is performed once every two seconds.

60 Hz. A built-in digital 5th order Sinc filter is also used. Data is collected with sampling rate of 128Hz. The user was instructed to blink once every two second. First part of this 2 second windows was extracted and labeled as artifact. Second part of this window is labeled as artifact-free data. The data was collected in 10 different sessions. Each session consisted of 10 eye blinks. Figure 14 shows two second window of EEG data captured. We collected data from 7 different subjects. We used leave-one-subject-out (LOSO) technique for testing purposes. Our network was trained on 90% of all the training data captured from 6 different subjects, 10% was used for validation and remaining one subject data was used for testing.

The EEG epochs of size 14 × 128 is passed to the first 2-d convolution layer consisting 16 filters of size 14 × 4. This ensures that adequate spatial filter is learned in the first layer. Zero padding is avoided to avoid large computations. After traditional convolution, a depthwise convolution is used with filter size of 1×32 and depth multiplier of 1 which means there will be 1 filters associated with each depth. This is followed by an average pooling layer with pool size of 1×32. A separable convolution is further used with 1×32 filter size which is again followed by an average pooling layer with pool size of 1×8. All layers are followed by a rectified linear unit (ReLU) activation function. Once these convolution operations have been performed, the output from the last average pooling layer is flattened into a single vector so that fully connected layer can be added. Only one fully connected layer is employed in the model which has 2 nodes with Softmax activation for this binary classification application. The network was trained using the Adam optimization method and a learning rate of 0.001. Categorical cross-entropy was used as the loss function. In this experiment we achieved 93.5% accuracy for detecting eye blink artifact.

Conclusion

In this chapter, we proposed an convolution neural network model using depthwise and separable CNN for identification of multiple EEG artifacts with average accuracy of 93.13%. Our CNN does not require any manual feature extraction and works on raw EEG signal for artifact identification. Our proposed network is implemented on Artix-7 FPGA and ASIC at post-layout level in 65nm CMOS technology. Our FPGA implementation is 1.7× to 5.15× higher energy efficient than some previous works. Moreover, our ASIC implementation is also 8.47× to 25.79× higher energy efficient compared to the previous works. We have also shown that the proposed network can be reconfigured to detect artifacts from another EEG dataset obtained by a 14-channel Emotiv EPOC+ headset in our lab and achieved an accuracy of 93.5% for eye blink artifact detection.

References

Abdelhameed AM, Daoud HG, Bayoumi M (2018) Deep convolutional bidirectional lstm recurrent neural network for epileptic seizure detection. In: 2018 16th IEEE International New Circuits and Systems Conference (NEWCAS), pp 139–143, DOI 10.1109/NEWCAS.2018.8585542

Barachant A, Bonnet S, Congedo M, Jutten C (2010) Riemannian geometry applied to bci classification. In: International Conference on Latent Variable Analysis and Signal Separation, Springer, pp 629–636

Bhardwaj S, Jadhav P, Adapa B, Acharyya A, Naik GR (2015) Online and automated reliable system design to remove blink and muscle artefact in eeg. In: 2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), IEEE, pp 6784–6787

Chander J, Bisasky J, Mohsenin T (2011) Real-time multi-channel seizure detection and analysis hardware. IEEE Biomedical Circuits and Systems (Biocas) Conference

Chollet F (2017) Xception: Deep learning with depthwise separable convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1251–1258

Chou CC, Chen TY, Fang WC (2016) Fpga implementation of eeg system-on-chip with automatic artifacts removal based on bss-cca method. In: 2016 IEEE Biomedical Circuits and Systems Conference (BioCAS), IEEE, pp 224–227

Dutande PV, Nalbalwar SL, Khobragade SV (2018) Fpga implementation of filters for removing muscle artefacts from eeg signals. In: 2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS), IEEE, pp 728–732

Fan H, Luo C, Zeng C, Ferianc M, Que Z, Liu S, Niu X, Luk W (2019) F-e3d: Fpga-based acceleration of an efficient 3d convolutional neural network for human action recognition. In:

2019 IEEE 30th International Conference on Application-specific Systems, Architectures and Processors (ASAP), vol 2160-052X, pp 1–8, DOI 10.1109/ASAP.2019.00-44

Hosseini M, Mohsenin T (2020) Binary precision neural network manycore accelerator. ACM Journal on Emerging Technologies in Computing Systems (JETC)

Howard AG, Zhu M, Chen B, Kalenichenko D, Wang W, Weyand T, Andreetto M, Adam H (2017) Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:170404861

HRen, et al (2020, in press) End-to-end scalable and low power multi-modal CNN for respiratory-related symptoms detection. In: 2020 IEEE 33rd International System-on-Chip Conference (SOCC) (SOCC 2020)

Iriarte J, Urrestarazu E, Valencia M, Alegre M, Malanda A, Viteri C, Artieda J (2003) Independent component analysis as a tool to eliminate artifacts in eeg: a quantitative study. Journal of clinical neurophysiology 20(4):249–257

Islam MK, Rastegarnia A, Yang Z (2016) Methods for artifact detection and removal from scalp eeg: a review. Neurophysiologie Clinique/Clinical Neurophysiology 46(4):287–305

Islam R, Hairston D, Mohsenin T (2017a) An eeg artifact detection and removal technique for embedded processors. In: IEEE Signal Processing in Medicine and Biology Symposium (SPMB), IEEE, DOI 10.1109/SPMB.2017.8257049

Islam R, Hairston WD, Oates T, Mohsenin T (2017b) An eeg artifact detection and removal technique for embedded processors pp 1–3, DOI 10.1109/SPMB.2017.8257049

Islam S, Razi A (2019) A path planning algorithm for collective monitoring using autonomous drones. In: 2019 53rd Annual Conference on Information Sciences and Systems (CISS), IEEE, pp 1–6

Islam S, Huang Q, Afghah F, Fule P, Razi A (2019) Fire frontline monitoring by enabling uav-based virtual reality with adaptive imaging rate. In: 2019 53rd Asilomar Conference on Signals, Systems, and Computers, IEEE, pp 368–372

Jafari A, Gandhi S, Konuru SH, Hairston WD, Oates T, Mohsenin T (2017) An eeg artifact identification embedded system using ica and multi-instance learning. In: 2017 IEEE International Symposium on Circuits and Systems (ISCAS), pp 1–4, DOI 10.1109/ISCAS.2017.8050346

Jafari A, et al (2019) Sensornet: A scalable and low-power deep convolutional neural network for multimodal data classification. IEEE Transactions on Circuits and Systems I: Regular Papers 66(1):274–287, DOI 10.1109/TCSI.2018.2848647

Jung TP, Humphries C, Lee TW, Makeig S, McKeown MJ, Iragui V, Sejnowski TJ (1998) Extended ica removes artifacts from electroencephalographic recordings. In: Advances in neural information processing systems, pp 894–900

Kaiser L, Gomez AN, Chollet F (2017) Depthwise separable convolutions for neural machine translation. arXiv preprint arXiv:170603059

Khatwani M, Hosseini M, Paneliya H, Mohsenin T, Hairston WD, Waytowich N (2018) Energy efficient convolutional neural networks for eeg artifact detection. In: 2018 IEEE Biomedical Circuits and Systems Conference (BioCAS), IEEE, pp 1–4

Krishnamoorthi R (2018) Quantizing deep convolutional networks for efficient inference: A whitepaper. arXiv preprint arXiv:180608342

Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems, pp 1097–1105

Lawhern V, Hairston WD, McDowell K, Westerfield M, Robbins K (2012) Detection and classification of subject-generated artifacts in eeg signals using autoregressive models. Journal of neuroscience methods 208(2):181–189

Lawhern VJ, Solon AJ, Waytowich NR, Gordon SM, Hung CP, Lance BJ (2018) EEGNet: a compact convolutional neural network for EEG-based brain–computer interfaces. Journal of Neural Engineering 15(5):056,013, DOI 10.1088/1741-2552/aace8c, URL `https://doi.org/10.1088%2F1741-2552%2Faace8c`

Mahabub A (2018) Design and implementation of a novel complete filter for eeg application on fpga. International Journal of Image, Graphics & Signal Processing 10(6)

Majidov I, Whangbo T (2019) Efficient classification of motor imagery electroencephalography signals using deep learning methods. Sensors 19(7):1736

MHosseini, HRen, HRashid, AMazumder, BPrakash, TMohsenin (2020) Neural networks for pulmonary disease diagnosis using auditory and demographic information. In: epiDAMIK 2020: 3rd epiDAMIK ACM SIGKDD International Workshop on Epidemiology meets Data Mining and Knowledge Discovery, ACM, pp 1–5, in press

Nuwer MR (1988) Quantitative eeg: I. techniques and problems of frequency analysis and topographic mapping. Journal of clinical neurophysiology: official publication of the American Electroencephalographic Society 5(1):1–43

Page A, Pramod S, et al (2015a) An ultra low power feature extraction and classification system for wearable seizure detection. In: Engineering in Medicine and Biology Society (EMBC), 2015 37th Annual International Conference of the IEEE

Page A, Sagedy C, et al (2015b) A flexible multichannel eeg feature extractor and classifier for seizure detection. Circuits and Systems II: Express Briefs, IEEE Transactions on 62(2):109–113

Paneliya H, Hosseini M, Sasan A, Homayoun H, Mohsenin T (2020) Cscmac-cyclic sparsely connected neural network manycore accelerator. In: 2020 21st International Symposium on Quality Electronic Design (ISQED), IEEE, pp 311–316

Prakash B, et al (2020) Guiding safe reinforcement learning policies using structured language constraints. In: SafeAI workshop Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI

Rashid HA, Manjunath NK, Paneliya H, Hosseini M, Mohsenin T (2020) A low-power lstm processor for multi-channel brain eeg artifact detection. In: 2020 21th International Symposium on Quality Electronic Design (ISQED), IEEE

Schetinin V, Schult J (2004) The combined technique for detection of artifacts in clinical electroencephalograms of sleeping newborns. IEEE Transactions on Information Technology in Biomedicine 8(1):28–35

Shiri A, Mazumder AN, Prakash B, Manjunath NK, Homayoun H, Sasan A, Waytowich NR, Mohsenin T (2020) Energy-efficient hardware for language guided reinforcement learning. In: Proceedings of the 2020 on Great Lakes Symposium on VLSI, pp 131–136

Sundaram K, et al (2016) Fpga based filters for eeg pre-processing. In: 2016 Second International Conference on Science Technology Engineering and Management (ICONSTEM), IEEE, pp 572–576

Szegedy C, Ioffe S, Vanhoucke V, Alemi AA (2017) Inception-v4, inception-resnet and the impact of residual connections on learning. In: Thirty-First AAAI Conference on Artificial Intelligence

Wang P, Jiang A, Liu X, Shang J, Zhang L (2018) Lstm-based eeg classification in motor imagery tasks. IEEE Transactions on Neural Systems and Rehabilitation Engineering 26(11):2086–2095, DOI 10.1109/TNSRE.2018.2876129

Winkler I, Haufe S, Tangermann M (2011) Automatic classification of artifactual ica-components for artifact removal in eeg signals. Behavioral and Brain Functions 7(1):30

Zhang C, Li P, Sun G, Guan Y, Xiao B, Cong J (2015) Optimizing fpga-based accelerator design for deep convolutional neural networks. In: Proceedings of the 2015 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays, pp 161–170

Zheng Y, Liu Q, Chen E, Ge Y, Zhao JL (2014) Time series classification using multi-channels deep convolutional neural networks. In: International Conference on Web-Age Information Management, Springer, pp 298–310