A Scalable and Low Power DCNN for Multimodal Data Classification

Ali Jafari, Morteza Hosseini, Houman Homayoun, and Tinoosh Mohsenin

***Abstract*—This paper presents SensorNet which is a scalable and low power embedded Deep Convolutional Neural Network (DCNN), designed to classify multimodal time series signals. Time series signals generated by different sensor modalities with different sampling rates are first converted to images (2-D signals) and then DCNN is utilized to automatically learn shared features in the images and perform the classification. SensorNet is scalable with respect to different types of multi-channel time series data, and does not require expert knowledge for extracting features for each sensor data. Additionally, it can achieve very high detection accuracy for different case studies, and has a very efficient architecture which makes it suitable to be employed at IoT and wearable devices. A custom low power hardware architecture is also designed for the efficient deployment of SensorNet at embedded real-time systems. SensorNet performance is evaluated using three different case studies including Physical Activity Monitoring, stand-alone Tongue Drive System (sdTDS) and Stress Detection and it achieves an average detection accuracy of 98%, 96.2% and 94% for each case study, respectively. We implement SensorNet using our custom hardware architecture on Xilinx FPGA (Artix-7) which on average consumes 0.3 mJ energy per classification while meeting all applications time requirements. To further reduce the power consumption, SensorNet is implemented using ASIC at the post layout level in 65-nm CMOS technology which consumes approximately 8× lower power compared to the FPGA.**

I. Introduction

Time series data is a sequence of data points in time order, that is gathered in different kinds of domains from healthcare where one can track a patient's vital signals to fitness and wellness where one can monitor a person's activity to engines in cars and power plants that employ sensor [1]. All these datasets are represented by a time series which either univariate or multivariate (multimodal) depending on the number of sensor modalities being measured. Multimodal signals are generated by different sensors such as accelerometers, magnetometers, and heart rate monitor, usually with different sampling frequencies.

Traditionally, time series classification problems have been solved with approaches like Dynamic Time Warping (DTW) [2] and k-nearest neighbor (k-NN) [3]. These methods or a combination of them provide a benchmark for current time series classification research. However, there are some challenges, such as feature extraction pre-process, requirement for expert knowledge in designing the features, and unscalablity, with these methods.

Deep Neural Networks (DNN) have recently become popular for multimodal time series signal processing [4]–[9]. However, DNN solutions usually have large and high power architectures that is not suitable for deployment at Internet of Things (IoT) and wearable devices.

In this paper, Sensornet shown in Fig. 1 is proposed which is a scalable Deep Convolutional Neural Network (DCNN) designed to classify multimodal time series signals in embedded, resource-bound settings with strict power and area budgets [10]. SensorNet: (1) is scalable as it can process different types of time series data with variety of input channels and sampling rates. (2) does not need to employ a separate signal processing techniques for pre-processing sensor data. (3) does not require expert knowledge for feature extraction. (4) achieves very high detection accuracy for different case studies. (5) has a very efficient architecture which makes it suitable to be deployed at low power and resource-bound embedded devices.

II. Related Work

In recent years, several multimodal data classification approaches have been proposed which are discussed in this section. [5] first converted time series into images and then used CNN for processing. The authors converted the time series into an image using two types of representations, i.e., Grammian Angular Fields (GAF) and Markov Transition Fields (MTF). The above mentioned architectures either modeled each variable separately before correlating them or required preprocessing the stream into an image. Authors in [6] proposed DeepSense which is a deep learning framework for time series mobile sensing data processing. DeepSense integrates convolutional and RNN to exploit and merge local interactions among similar mobile sensors and extract temporal relationships to model signal dynamics. They deployed DeepSense at Nexus5 and Intel Edison and reported latency and power consumption results. [7] proposed an approach that assemble signal sequences of accelerometers and gyroscopes into a novel activity image, which enables Deep CNN to automatically learn features from the activity image for the activity recognition. 2D Discrete Fourier Transform is applied to the signal image and its magnitude is chosen as their activity image. [8] proposed a Cardiologist-Level Arrhythmia Detection using a 34-layer CNN and they exceed the average cardiologist performance in both sensitivity and precision. [11] proposed a solution for multimodal activity recognition on FPGA, which uses different signal processing algorithms including feature extraction, Principle Component Analysis and a 2-layer Neural Network. In summary, most of the previous work do not propose a real-time hardware solution for multimodal time-series data classification or use general-purpose processors [6], [12] which results in high power consumption.

978-1-7281-1968-7/18/31.00©2018*IEEE*

Authorized licensed use limited to: University of Maryland Baltimore Cty. Downloaded on June 29,2021 at 12:18:54 UTC from IEEE Xplore. Restrictions apply.

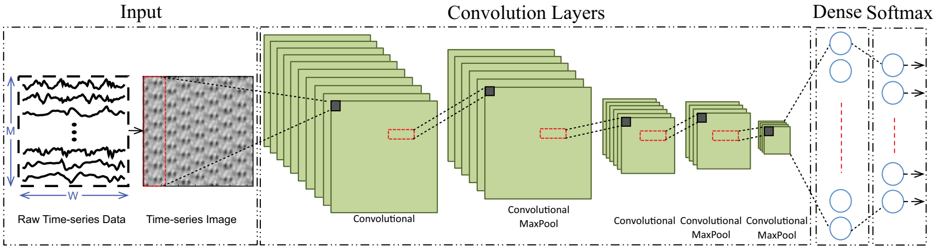

Fig. 1. The proposed SensorNet takes a 1-channel image (constructed from the raw sensor signals) as input, and consists of 5 convolutional layers, 3 pooling layers, 1 dense layer and a softmax layer. The architecture is denoted by 32(Mx5)-16(1x5)-2-16(1x5)-8(1x5)-2-8(1x5)-2-64-N and its generic notation is C_1(size)-C_2(size)-S_1-C_3(size)-C_4(size)-S_2-C_5(size)-S_3-H_1-N.C_1..C_5 are the convolutional layers, S_1..S_3 are the max-pooling layers, H_1 is the size of dense layer and N is the softmax layer of size N.

III. SensorNet Architecture Design

An overview of the proposed SensorNet architecture is discussed here. SensorNet is designed to capture correlations between various modalities simultaneously. To capture these correlations, depends on the application a snapshot (of fixed window size) of raw signals is converted to an image (2-D signal) and is passed to the network and is processed. Fig. 1 shows a high-level block diagram of the proposed system which consists of pre-processing, convolutional, fully connected and softmax layers.

A. Signal Preprocessing

Consider a given time series that consists of M modalities/variables with same or different sampling frequency. Prior to training, each variable is independently normalized using the $l2$ norm. To generate an image from the normalized variables, a sliding window of size W and step-size S is passed through all variables, creating a set of images of shape $1 \times W \times M$ (single channel image). The label associated with this image depends on the dataset. The datasets used to test the network in this paper contain a label for every time step. Since a single label is assigned to each image, the label of the current time step is taken as the label of the image (and the label that needs to be predicted subsequently while testing). A given image generated at time-step I_t has the prior states of each variable from $(t - W + 1)...t$. Thus, the network can look back W prior states of each variable and given the current state of each variable, predicts the label.

B. Neural Network Architecture

Fig. 1 shows SensorNet architecture. It consists of 5 convolutional layers, 1 fully connected and a softmax layer that is equivalent in size to the number of class labels (depending on the case study). In the pre-processing stage, SensorNet takes the input time series data and fuses them into images. Then, the images are passed into the convolutional layers and some features which are shared across multiple modalities are generated using a set of local filters. Then, these features are fed to the fully-connected and the softmax layers. SensorNet architecture including number of layers, number of filters and filter shapes for each layer are chosen based on an extensive hyperparameter optimization process which will be discussed in details in Section V.

First, second, third, fourth and fifth convolutional layers contain 32, 16, 16, 8 and 8 filter sets, respectively. The convolution filters have a height of either M or 1, because it's assumed that there are no spatial correlations between the variables. Also, the ordering of variables prior to generating images doesn't affect the ability of the network to perform classification. A filter of height M or 1 remains unaffected by the ordering of the variables. Therefore, the filter size for the first convolutional layer is $M \times 5$ where M is number of input modalities. For other layers, filter shape of 1×5 is chosen. Max-pooling is applied thrice, once after the second convolutional layer, then after the fourth convolutional layer and the last one after the fifth convolutional layer. A max-norm regularization of 1 is used to constrain the final activation output. The pooling size for all max-pooling layers is 1×2. Two fully connected layers are employed in SensorNet which the first one has a size of 64 nodes the second one has a size equivalent to the number of class labels with Softmax activation. All the layers of the network have their weights initialized from a normal distribution. A learning rate of 0.0001 is used to train the network. Rectified Linear Unit (ReLU) is used as activation functions for all the layers. The network is trained using backpropagation and optimized using RMSprop.

IV. Experimental Results and Analysis

In this section, SensorNet is evaluated using three real-world case studies including Physical Activity Monitoring [14], stand-alone dual-mode Tongue Drive System (sdTDS) [15] and Stress Detection [16] and in depth analysis and experimental results are provided. For all the case studies, SensorNet is trained using Keras with the TensorFlow as backend on a NVIDIA 1070 GPU and 8 GB RAM. Models are trained in a fully-supervised way, backpropagating the gradients from the Softmax layer through to the convolutional layers. For all the experiments in this paper we train SensorNet with 100 epochs. We experienced that after 100 epochs the validation loss and accuracy for the case studies are stable and satisfactory.

A. Case study 1: Physical Activity Monitoring

Physical Activity Monitoring dataset (PAMAP2) [14] records 12 physical activities performed by 9 subjects. The physical activities are, for instance: 'standing', 'walking', 'lying' and 'sitting'. Three IMUs (inertial measurement units)

Authorized licensed use limited to: University of Maryland Baltimore Cty. Downloaded on June 29,2021 at 12:18:54 UTC from IEEE Xplore. Restrictions apply.

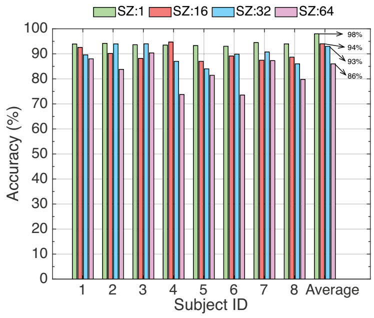

Fig. 2. Comparison of SensorNet classification accuracy for Physical Activity Monitoring case study. The results are for different subjects with a sliding window of size 64 samples and step-size (SZ) of 1-16-32-64.

and one heart rate monitor are placed on chest, arm and ankle to record the data. The sampling frequency of the IMU sensors is 100 Hz and the heart rate monitor sensor has a sampling frequency of 9 Hz. In total, the dataset includes 52 channels of data but 40 channels are valid according to [14]. Therefore, the first convolutional layer for this case study in the SensorNet has 32 filter sets and each filter size is 40×5. Other convolutional layers have 16, 16, 8 and 8 filter sets with a size of 1×5. For this experiment, 80%, 10% and 10% of the entire data for each subject is chosen randomly as the training, validation and testing set.

Fig. 2 shows the classification accuracy of SensorNet for the Physical Activity Monitoring case study for different subjects with a sliding window of size 64 samples and step-size of 1-16-32-64. As can be seen from the figure, all subjects with step-size 1 achieve a high detection accuracy. However, as the step-size increases from 1 to 64 the detection accuracy decreases. The average accuracy of all subjects with step-sizes of 1, 16, 32 and 64 are 98%, 94%, 93% and 86%, respectively.

B. *Case study 2: Stand-alone Tongue Drive System*

In [15], authors proposed and developed a stand-alone Tongue Drive System (sTDS) which is a wireless wearable headset and individuals with severe disabilities can use it to potentially control their environment such as computer, smartphone and wheelchair using their voluntary tongue movements [17]. The sdTDS prototype includes a local processor, four magnetic and acceleration sensors, a BLE transceiver, and a magnetic tracer which is glued to the user's tongue. The raw data generated by 4 magnetometers and accelerometers are transferred into a FPGA processor where the entire signal processing including feature extraction and classification is performed by SensorNet and 12 different user-defined commands can be generated.

Several different data sets are captured using sdTDS for training and testing purpose. sdTDS generates 24 channels of time series data that corresponds to tongue and head movements. For the sdTDS, first convolutional layer of the SensorNet has 32 filter banks and each filter size is 24×5. Other convolutional layers have 16, 16, 8 and 8 filter banks with a size of 1×5. For this experiment, 80%, 10% and 10% of the entire data for each trivial is chosen randomly as the training, validation and testing sets, respectively. For sdTDS case study, SensorNet detection accuracy for tongue and head movements detection is approximately 96.2%.

C. *Case study 3: Stress Detection*

This database contains non-EEG physiological signals used to infer the neurological status including physical stress, cognitive stress, emotional stress and relaxation of 20 subjects. The dataset was collected using non-invasive wrist worn biosensors. A wrist worn Affectiva collects electrodermal activity (EDA), temperature and acceleration (3D); and a Nonin 3150 wireless wristOx2 collects heart rate (HR) and arterial oxygen level (SpO2) data [16]. Therefore, in total the dataset includes 7 channels of data. The sampling frequency of wrist worn Affectiva is 8 Hz and wristOx2 has a sampling frequency of 1 Hz.

First convolutional layer for this dataset in the SensorNet has 32 filter sets and each filter size is 7×5. Other convolutional layers have 16, 16, 8 and 8 filter sets with a size of 1×5. For this experiment also 80%, 10% and 10% of the entire data is chosen randomly as the training, validation and testing set. After 100 epochs the average classification accuracy of SensorNet for Stress Detection case study for 20 different subjects is approximately 94%.

V. SensorNet Complexity Reduction

As it was discussed in section III, SensorNet utilizes 5 convolutional layers, followed by 2 fully-connected layers. First convolutional layer has 32 filter sets and each filter size is $M \times 5$, where M is the number of input channels. Other convolutional layers have 16, 16, 8 and 8 filter banks with a size of 1×5. The first fully-connected layer has 64 nodes and the number of nodes in the last one is equivalent to the number of labels for any specific application. One of the primary objectives of this paper is to be able to efficiently deploy SensorNet at IoT and wearable devices which have strict power and area budgets. Therefore, we perform extensive hyperparameter optimization for SensorNet with the goal of reducing memory requirements, hardware complexity and power consumption while achieving high detection accuracy. In this section, we specifically explore the impact of changing the following parameters or configurations on SensorNet performance: *1) Number of convolutional layers, 2) Number of filters, and 3) Filter shapes.*

A. *Number of Convolutional Layers*

In this experiment, we compare six SensorNet configurations with an increase in the number of convolutional layers, for the three different case studies. The comparison has been made in terms of detection accuracy, number of operations, number of parameters and memory requirements. Fig. 3-A, 3-B, 3-C and 3-D depict the impact of increasing the number of convolutional layers on the number of model parameters and memory requirements. As is shown in the figures, by increasing the number of convolutional layers, the number of model parameters and memory requirements decrease which is

Authorized licensed use limited to: University of Maryland Baltimore Cty. Downloaded on June 29,2021 at 12:18:54 UTC from IEEE Xplore. Restrictions apply.

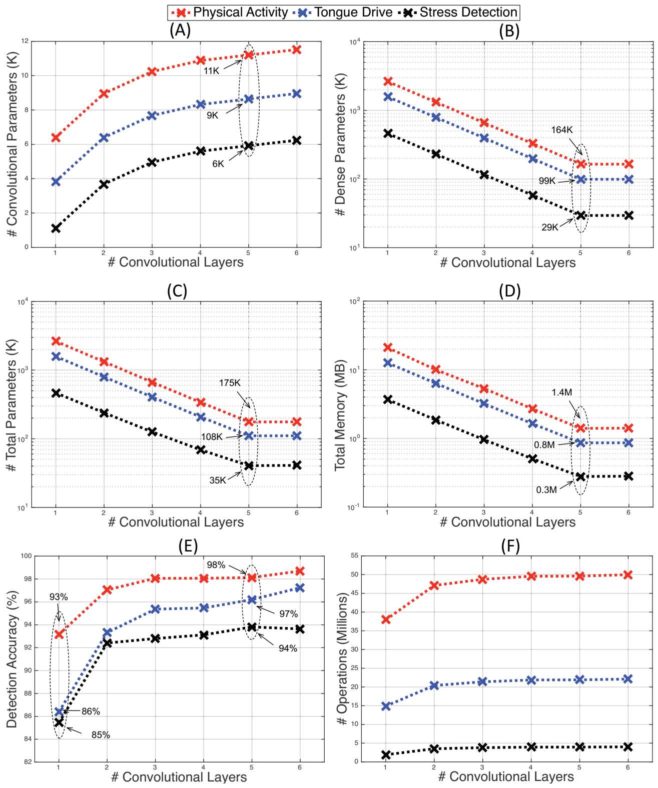

Fig. 3. Impact of increasing number of convolutional layers on memory requirements, detection accuracy and number of operations of SensorNet, for three different applications.

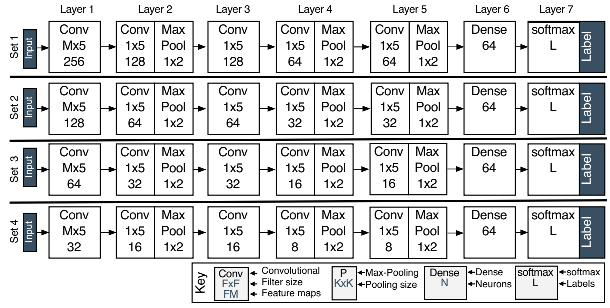

Fig. 4. SensorNet configurations with four different filter sets. Number of filters for convolutional layers are doubled for each filter set. M and L are the number of data channels and labels for different case studies.

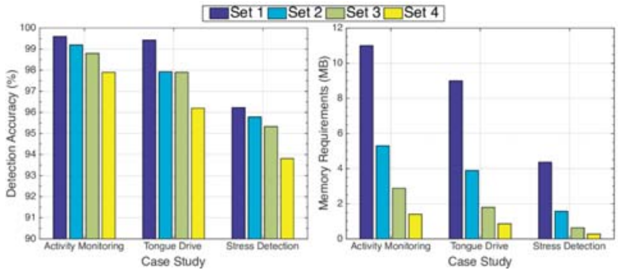

Fig. 5. Comparison of four different SensorNet configurations in terms of detection accuracy, and memory requirements. By adding additional model weights to each layer, the computation and memory grow dramatically with only modest improvement in detection accuracy.

desired. The reason is that we use three max-pooling layers after the convolutional layers, therefore by adding more convolutional layers the size of the time series images shrink and the fully-connected layer needs to process less number of data and thus requires less memory. Fig. 3-E shows the impact of increasing the number of convolutional layers on detection accuracy. As it can be seen from the figure, if the neural network is too shallow high-level features can not be learned, therefore the detection accuracy is low. However, the results show that, by increasing the number of convolutional layers detection accuracy increases but up to 5 convolutional layers. After that, for Activity Monitoring and sdTDS case studies the accuracy improve slightly but for the Stress Detection reduces because the useful features may be filtered out during the convolutional and max-pooling processes. Also, by adding additional convolutional layer the number of operations to finish a classification task increases slightly which is shown in Fig. 3-F. This analysis results show that SensorNet with 5 convolutional layers is the best candidate with regards to detection accuracy, number of convolutional operations and memory requirements.

B. Number of Filters

The number of filters are another hyperparameter for implementing SensorNet on resource-limited devices because the number of model weights affect the memory requirements and computations time. In this experiment, as shown in Fig. 4, we keep the number of convolutional layers fix (5 layers) and increase the number of filters for each layer in four different configurations and measure the impact of the number of filters on the neural network for the three case studies. Fig. 5 shows a comparison of number of required parameters for different trained models. As is shown in the figure, as we increase the number of filters, the detection accuracy improves. However, the number of operations, memory requirements and the number of model parameters increase which is not desired for hardware implementation. Based on the results, SensorNet with different filter sets achieves similar detection accuracies, but Set 4 needs less parameters and requires smaller memory compared to other filter sets and therefore is chosen to be implemented on hardware.

VI. SensorNet Hardware Architecture Design

Fig. 6 depicts SensorNet hardware architecture with implementation details. The main components of SensorNet on hardware consists of Convolutional, Max-pooling and Fully-connected blocks. ReLU is used as the activation for the convolutional blocks and the first fully-connected layer, and Softmax is used for the last fully-connected layer and will perform classification task. Fig. 6-A shows the convolution block composing of one multiplier, one adder/subtractor, one cache for saving filters, input feature maps and output feature maps, a multiplexers, a few registers, and a state machine block.

Authorized licensed use limited to: University of Maryland Baltimore Cty. Downloaded on June 29,2021 at 12:18:54 UTC from IEEE Xplore. Restrictions apply.

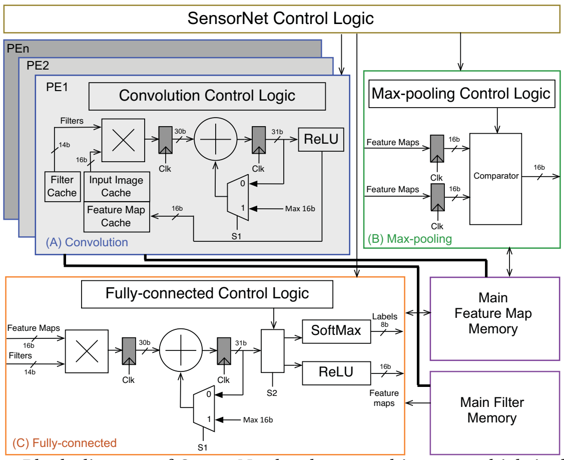

Fig. 6. Block diagram of SensorNet hardware architecture which includes convolution, max-pooling and fully-connected blocks and also a top-level state-machine which controls all the blocks.

TABLE I

IMPLEMENTATION RESULTS OF THE PROPOSED SENSORNET ON XILINX FPGA (ARTIX-7). THE RESULTS OBTAINED AT CLOCK FREQUENCY OF 100 MHZ

Merits/ Case Studies	Physical Activity Monitoring		Tongue Drive System		Stress Detection	
	Serial	Fully Paral.	Serial	Fully Paral.	Serial	Fully Paral.
# Used PE	1	8	1	8	1	8
BRAM	14	35	12	26	11.5	22
# DSP slices	3	10	3	10	3	10
# of Slices	982	2506	961	2430	939	2388
Latency (ms)	14.8	2	11.2	1.5	7.3	1
Throughput (label/s)	67	491	89	641	136	952
Total Power	116	175	113	160	112	154
Total Energy	1.7	0.35	1.3	0.24	0.8	0.15

When the convolution operations are done for all the input feature maps, the output feature maps will be saved into the main feature map memory. The input data coming from the sensors are 16-bit two's complement. Also, the filters are considered to be 16-bit two's complement. After performing the convolution, the data will pass to ReLU activation function. The output of ReLU is truncated to 16 bits and saved in feature map memory. An offline training is performed to obtain model weights using keras. The model weights are converted to fixed-point format and are represented by 16 bits. The floating-point arithmetic is complex and requires more area, therefore use of fixed point arithmetic will avoid complex multipliers. Fig. 6-B shows the max-pool block which contains some registers and a comparator. The input to the max-pool is feature maps data, which is formed by convolution block. After max-pooling operations finish, the results will be saved into the main feature map memory. 6-C shows the fully-connected block. As is shown, the architecture consists of a serial dot product engine, a dynamic sorting logic for the Softmax activation function, ReLU logic and a state machine for controlling all sub-blocks. Depends on the layer either ReLU or Softmax can be used. After finishing computations for the fully-connected layers the results will be saved into the main feature memory.

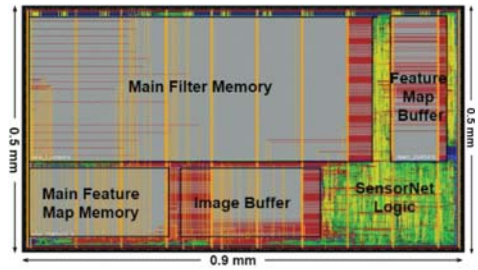

Fig. 7. Post-layout view of SensorNet ASIC implementation for PAMAP dataset in 65 nm, TSMC CMOS technology with frequency of 100 MHz.

VII. HARDWARE IMPLEMENTATION RESULTS

A. FPGA Implementation Results and Analysis

The complete proposed SensorNet which includes convolution, max-pooling, fully-connected and activation functions are implemented on an Xilinx Artix-7 FPGA at clock frequency of 100 MHz. Verilog HDL is used to describe SensorNet hardware architecture. Table I provides SensorNet performance results for different architectures, including serial, semi-serial and fully-parallel designs. As is understood, for a given network increasing the number of PE, increases power consumption slightly but improves both latency and energy consumption.

B. ASIC Implementation Results and Analysis

To reduce the overall power consumption, an application-specified integrated circuit (ASIC) for SensorNet is implemented at the postlayout level in 65-nm CMOS technology with 1-V power supply. Due to the ability of ASIC designs to run at high clock frequencies, only SensorNet with a serial architecture is implemented, for all the case studies including Physical Activity Monitoring, sdTDS and Stress Detection. A standard-cell register-transfer level (RTL) to Graphic Data System (GDSII) flow using synthesis and automatic place and route is used. The proposed SensorNet including convolution, max-pooling, fully-connected with activation functions is implemented using Verilog to describe the architecture, synthesized with Synopsys Design Compiler, and place and routed using Cadence SOC Encounter. The ASIC layouts for all the case studies are shown in Fig. 7. The implementation results are provided in table II. SensorNet is able to operate at maximum clock frequency of 888 MHz. However, the clock frequency has been reduced to 100 MHz to reduce the power consumption and have a fair comparison with FPGA performance results. The ASIC implementation reduces the power consumption by a factor of 8 on average.

C. Comparison with Existing Work

Table III shows a comparison of the proposed SensorNet hardware implementation results with existing deep learning solutions on embedded devices. When SensorNet is deployed at Xilinx FPGA device with a fully-parallel design and running at 100 MHz, it consumes 154 mW and 175 mW power for Stress Detection and Activity Recognition

Authorized licensed use limited to: University of Maryland Baltimore Cty. Downloaded on June 29,2021 at 12:18:54 UTC from IEEE Xplore. Restrictions apply.

TABLE II

SENSORNET ASIC IMPLEMENTATION RESULTS AT OPERATING FREQUENCY OF 100 MHZ. CMOS FABRICATION PROCESS IS 65 NM WITH 1 V POWER SUPPLY

Metrics/ Case Studies	Physical Activity Monitoring	Tongue Drive System	Stress Detection
Area utilization	95	94	95
Clock freq. (MHz)	100		
Max. clock freq. (MHz)	857	867	888
Core area (mm^2)	0.4	0.3	0.2
Latency (ms)	14	11	7
Throughput (label/s)	67	89	136
Total power (mW)	18.5	15.9	9.2
Energy (mJ)	0.25	0.17	0.06

TABLE III

COMPARISON OF SENSORNET PERFORMANCE WITH RELATED WORKS.

Metrics	[6]	[12]	[11]	**This work**		**This work**	
Application	Human Activity			Stress Detection		Human Activity	
# Channels	6	6	40	7		40	
Technique	DeepSense	RBM	PCA+NN	SensorNet		SensorNet	
Platform	Atmel Edison	Snapdragon	FPGA	FPGA	ASIC 65nm	FPGA	ASIC 65nm
Latency (ms)	105	50	146	1	0.8	2	1.6
Power (W)	6	1.9	0.29	0.15	0.06	0.17	0.10
Energy (mJ)	700	96	43	0.15	0.05	0.35	0.16

case studies, respectively. For the same Activity Recognition case study, FPGA-based SensorNet consumes 1.7×, 73× and 123× lower power, latency and energy compared to [11]. SensorNet with a fully-parallel architecture for Activity Monitoring case study, uses 12 DSP Slices, 35 BRAM, 2500 Slices. Whereas [11], uses 19 DSP Slices, 65 BRAM, 2175 Slices. Therefore, SensorNet utilize less DSP slices and BRAM, but slightly more slices. Compared to [6] and [12], FPGA-based SensorNet achieves 38×, 12× lower power, 105×, 50× lower latency and 4666×, 640× lower energy, for the applications with similar number of data channels.

VIII. CONCLUSION

This paper presents Sensornet, a low power embedded deep convolutional neural network for multimodal time series signal classification. First, the time series data generated by different sensors are converted to images (2-D signals) and then a single DCNN is employed to learn features over multiple modalities and perform the classification task. The proposed SensorNet is scalable with processing different types of time series data, and has a very efficient architecture which makes it suitable to be employed at IoT and wearable devices. SensorNet was tested using three real-world case studies including Physical Activity Monitoring, dual-mode Tongue Drive system and Stress detection and based on the results, it achieved an average classification accuracy of 98%, 96.2%, 94% for each application, respectively. A custom low power hardware architecture is also designed for the efficient deployment of SensorNet at embedded real-time systems and when is implemented on Xilinx FPGA (Artix-7), on average consumes 0.3 mJ energy. To further reduce the power consumption, SensorNet is implemented using ASIC at the post layout level in 65-nm CMOS technology which consumes almost 8× lower power compared to the FPGA.

IX. ACKNOWLEDGEMENT

This research was sponsored partially by National Science Foundation under Grant No. 1652703 and the Army Research Laboratory under Cooperative Agreement Number W911NF-10-2-0022.

REFERENCES

[1] A. Jafari *et al.*, "A low power seizure detection processor based on direct use of compressively-sensed data and employing a deterministic random matrix," in *BioCAS*, 2015.

[2] S. Seto, W. Zhang, and Y. Zhou, "Multivariate time series classification using dynamic time warping template selection for human activity recognition," in *Computational Intelligence, 2015 IEEE Symposium*.

[3] H. Ding, G. Trajcevski, P. Scheuermann, X. Wang, and E. Keogh, "Querying and mining of time series data: experimental comparison of representations and distance measures," *Proceedings of the VLDB Endowment*, vol. 1, no. 2, pp. 1542–1552, 2008.

[4] A. Jafari, M. Hosseini, A. Kulkarni, C. Patel, and T. Mohsenin, "Binmac: Binarized neural network manycore accelerator," in *Proceedings of the 28th Edition of the Great Lakes Symposium on VLSI*. ACM, 2018.

[5] Z. Wang *et al.*, "Encoding time series as images for visual inspection and classification using tiled convolutional neural networks," in *Workshops at the 29th AAAI Conference on Artificial Intelligence*.

[6] S. Yao *et al.*, "Deepsense: A unified deep learning framework for time-series mobile sensing data processing," in *Proceedings of the 26th International Conference on World Wide Web*, 2017, pp. 351–360.

[7] W. Jiang and Z. Yin, "Human activity recognition using wearable sensors by deep convolutional neural networks," in *Proceedings of the 23rd ACM international conference on Multimedia*.

[8] P. Rajpurkar, A. Y. Hannun, M. Haghpanahi, C. Bourn, and A. Y. Ng, "Cardiologist-level arrhythmia detection with convolutional neural networks," *arXiv preprint arXiv:1707.01836*, 2017.

[9] R. Azencott, V. Muravina, R. Hekmati, W. Zhang, and M. Paldino, "Automatic clustering in large sets of time series," in *Contributions to Partial Differential Equations and Applications*.

[10] A. Jafari, A. Ganesan, C. S. K. Thalisetty, V. Sivasubramanian, T. Oates, and T. Mohsenin, "Sensornet: A scalable and low-power deep convolutional neural network for multimodal data classification," *IEEE Transactions on Circuits and Systems I: Regular Papers*, 2018.

[11] K. Basterretxea *et al.*, "A wearable human activity recognition system on a chip," in *Design and Architectures for Signal and Image Processing (DASIP), 2014 Conference on*. IEEE, 2014, pp. 1–8.

[12] V. Radu *et al.*, "Towards multimodal deep learning for activity recognition on mobile devices," in *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct*. ACM, 2016, pp. 185–188.

[13] A. Kulkarni, A. Page, N. Attaran, A. Jafari, M. Malik, H. Homayoun, and T. Mohsenin, "An energy-efficient programmable manycore accelerator for personalized biomedical applications," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 2017.

[14] A. Reiss and D. Stricker, "Introducing a new benchmarked dataset for activity monitoring," in *Wearable Computers (ISWC), 2012 16th International Symposium on*. IEEE, 2012, pp. 108–109.

[15] A. Jafari *et al.*, "A low-power wearable stand-alone tongue drive system for people with severe disabilities," *IEEE transactions on biomedical circuits and systems*, vol. 12, no. 1, pp. 58–67, 2018.

[16] J. Birjandtalab *et al.*, "A non-eeg biosignals dataset for assessment and visualization of neurological status," in *Signal Processing Systems (SiPS), IEEE International Workshop on*. IEEE, 2016, pp. 110–114.

[17] A. Jafari *et al.*, "An embedded fpga accelerator for a stand-alone dual-mode assistive device," in *BioCAS, 2017 IEEE*.

[18] T. Abtahi, C. Shea, A. Kulkarni, and T. Mohsenin, "Accelerating convolutional neural network with fft on embedded hardware," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 2018.

[19] A. Page *et al.*, "Utilizing deep neural nets for an embedded ecg-based biometric authentication system," in *BioCAS, 2015 IEEE*.

[20] H. M. Kamali, "Using multi-core hw/sw co-design architecture for accelerating k-means clustering algorithm," *arXiv preprint arXiv:1807.09250*, 2018.

Authorized licensed use limited to: University of Maryland Baltimore Cty. Downloaded on June 29,2021 at 12:18:54 UTC from IEEE Xplore. Restrictions apply.