Low Power Tiny Binary Neural Network with improved accuracy in Human Recognition Systems

Antonio De Vita*, Danilo Pau†, Luigi Di Benedetto*, Alfredo Rubino*, Fre´de´ric Pe´trot‡ and Gian Domenico Licciardo*

*Department of Industrial Engineering, University of Salerno, Fisciano (SA), Italy
Email: andevita@unisa.it, ldibenendetto@unisa.it, arubino@unisa.it, gdlicciardo@unisa.it
†System Research and Applications, STMicroelectronics, Agrate Brianza (MI), Italy
Email: danilo.pau@st.com
‡TIMA, Grenoble-INP, University of Grenoble Alpes, Grenoble, France
Email: frederic.petrot@univ-grenoble-alpes.fr

***Abstract*— Human Activity Recognition requires very high accuracy to be effectively employed into practical applications, ranging from elderly care to microsurgical devices. The highest accuracies are achieved by Deep Learning models, but these are not easily deployable in handheld or wearable devices with very constrained resources. We therefore present a new HAR system suitable for a compact FPGA implementation. A new Binarized Neural Network (BNN) architecture achieves the classification based on data from a single tri-axial accelerometer. From our experiments, the effect of gravity and the unknown orientation of the sensor cause a degradation of the accuracy. In order to compensate for these issues, we propose a HW-friendly algorithm to pre-process the raw acceleration signal. Moreover, the very low power and hardware friendly BNN has been trained and validated on the PAMAP2 dataset, for which the pre-processing operations increase the accuracy from 51% to 99% in the best case. Aiming for a low-power design, we designed both a custom circuit to perform the pre-processing operations and a hardware accelerator for the BNN. The design on FPGA features a power dissipation of 72 mW and occupies 6788 LUTs.**

Keywords—binarized neural networks; FPGA; human activity recognition; inertial sensors

I. INTRODUCTION

One of the most widespread features in modern handheld and wearable devices is Human Activity Recognition (HAR), which enables the automatic recognition of human activities by analyzing data collected by sensors [1]. HAR is employed in various application fields requiring a very high accuracy level such as Parkinson's disease monitoring [2], rehabilitation [3], and assisted living [4]. Modern MEMS, such as accelerometers, gyroscopes, and magnetometers [5], are accurate enough to be effectively used in conjunction with Deep Learning (DL) models in order to build HAR systems with high accuracy requirements. However, we must face the demands in physical resources required to implement the high computational complexity of DL models [6], [7]. In addition, in HAR systems, inertial sensors can have an arbitrary and changeable orientation in space, with the gravity acceleration providing a variable contribution to each sample, which is related to the reference frame integral to the sensor. As the orientation of the sensor is generally unknown, this contribution needs some specific signal processing techniques to be removed [8]. In this context, the delay of the data transfer and the availability of connections make cloud unusable when real-time operation is required [9], let alone when privacy is at stake. Instead, processing "at the edge", namely locally on the device and close to the sensors is highly desirable [10]. To enable this, many reduced-precision DL models have been proposed, such as Binarized Neural Networks (BNNs) [11] and Ternary Neural Networks (TNNs) [12], which use only 1 or 2 bits, respectively, to encode the network parameters. Compared to a single precision Floating Point (FP) [13] model, the use of the BNNs, for example, could lead to a reduction of the memory requirements up to 32×, and to multiplier-less designs with significant area and power reductions for the derived HW implementations [14], [15]. Unfortunately, the price of these benefits is a significant loss in accuracy, which is an obstacle to the use of reduced precision DL models in actual systems [11].

In this work, the performance of a partially-binarized hybrid neural network (HNN) [16] is evaluated when it is used to build a HAR system designed to overcome the above limitations. We evaluate the accuracy of the model when data is acquired from only one tri-axial accelerometer, along with the amount of physical resources for its HW implementation. Next, we evaluate the accuracy advantage of a custom pre-processing stage considering its HW resource requirements. Results show that the HAR system, made of accelerometer, pre-processing stage, and HNN, exhibits an average recall up to 99% when trained on the PAMAP2 dataset [17]. Moreover, the modularity of the calculation scheme and HNN model allows a very compact design, which induces a total power consumption of 72 mW, (mostly quiescent power), and maps on 6788 LUTs, when implemented on a Xilinx Artix-7 FPGA. We obtained these results on an actual board that we developed to demo the real time operation of the system running at a maximum operating frequency of 41 MHz. The use of the FPGA allows to easily adapt the architecture to different classification tasks. In fact, our design can be reinitialized with new weights from an offline training phase. Also, we can straightforwardly reprogram our FPGA design when a

change in the topology of the model is needed. All the results overcome the state of the art for this kind of systems.

II. Hybrid Neural Network

The HNN is similar to a BNN, but while all weights are binarized, i.e. constrained to +1 and –1, the output activations for some layers are not. In accordance with [16], the HNN can classify between 5 different human activities. To test, train, and validate the model, we used the PAMAP2 public dataset [17], selecting 5 standard activities. The HNN model, as schematized in Fig. 1, has been built using Lasagne [18].

A. Model Description

Four different stages can be identified. The first stage is made up by a convolutional (CONV) layer and a normalization (NORM) layer. The input to the model is a tri-axis acceleration. An input window of 24 samples is required to produce an output label, with each sample made up by the 3 components of the acceleration vector. The input samples are not binarized, and 16 bits are used for their binary representation. The CONV layer applies a set of 8 filter kernels to each one of the 3 axes independently. The dimension of each kernel is 5×1. After that, a NORM layer implements the Batch Normalization [19]. To binarize the output, we use the sign function as the activation function in the fi rst stage: each value among the output activations thus is constrained to +1 or −1 and can be represented with a single bit. The second stage is made by a CONV layer, which applies a set of 8 filter kernels with dimensions 5×8. As shown in Table I, 60.6% of the total ADD/SUB operations are performed in this layer. In this case, the ReLU function is used as activation function, and no binarization occurs. Then, a Max Pooling layer is applied independently to each of the 3 axes, with dimensions 4×1. The third stage is a Fully Connected (FC) layer with 64 neurons and a NORM layer. As the input activations to the second stage are binarized, 7 bits are needed to cover the whole range of values for the output activations in the second and in the third stage. It is worth noting that 77.0% of the total memory requirement is due to the first FC layer. The fourth stage is made up by a FC layer with 5 neurons, that are the number of output classes. Finally, a SoftMax classifier returns the probability associated to each class.

B. Training and Accuracy Results

The accuracy of the proposed HAR system has been evaluated by using data from the PAMAP2 dataset. The dataset provides data from 9 users performing 12 daily human activities. Three Inertial Measurement Units (IMUs) and a heart rate sensor are used to collect the data. Each IMU was located in a different part of the body, thus 3 sensor positions can be chosen: *ankle*, *hand*, and *chest*. In our system, the input comes from a single accelerometer. Therefore, only the samples from the accelerometers have been considered. Also, for each position, two different ranges can be selected for the accelerometer: ±16g, and ±6g, but in both cases the sensor use 13 bits to represent the data. Thus, a different sensitivity is related to the two ranges. In total, data from 6 different sensors can be considered, namely: *ankle16g*, *ankle6g*, *hand16g*, *hand6g*, *chest16g*, *chest6g*. The data have been down-sampled from 100 Hz to 25 Hz, considering the usual frequencies in HAR [20]. To be compliant with the topology of the HNN model in [16], 5 human activities have been selected among the 12 standard activities provided by the PAMAP2 dataset, namely: *standing*, *walking*, *running*, *biking*, and *rope jumping*. The method presented in [21] has been used to train the model, by setting the number of epochs to 30 and the batch size to 500. For training and testing purposes, the dataset is split in sets of consecutive samples that are associated to the same activity. The number of samples in each set is called FL. During training process, we sequentially feed the HNN with these sets. As the batch size is 500, parameters are updated every 500 sets during each epoch. The number of samples used during training, testing, and validation are reported in Table II for each FL. Note that the FL refers to the number of samples before the down-sampling process. Thus, a FL equal to 512 corresponds to a time window of 5.12 s. Training results are summarized in Table III, where the average recall is reported for each one of the 6 sensors and considering 3 different Frame Lengths (FLs), namely 5120, 1024, and 512. The average recall tends to increase with the FL in most cases. The HNN model has poor performance in recognizing the human activities from the raw data provided by the dataset. In fact, the average recall is lower than 90% in 88.89% of the cases. Better results are achieved only when the sensor position is *chest* and the FL is equal to 5120. This can be explained by considering the effect of the sensor orientation when performing human daily activities, such as walking, running, and the like. In fact, while the sensors at the hand and ankle positions are subject to rotational movements, the ones at the chest position are not. Thus, in the first case there is an issue related to the orientation of the sensor, i.e. the orientation is different for each sample. We can imagine that the unknown orientation of the sensors behaves like a noise, that overlaps with the human motion data, making it harder to achieve a correct classification without increasing the model size.

III. Pre-Processing Operations

To increase the accuracy of the HNN, we propose a pre-processing scheme that removes the noise resulting from the varying orientation of the sensor. Two sequential phases can be distinguished, that are *filtering* and *reference frame rotation*. The first one is used to separate the gravity component from the measured acceleration. Then, gravity is removed from the measured acceleration and used to define a new fixed reference frame, which does not depend on the sensor orientation. The filtering operation is based on the idea that the gravity acceleration signal is located in the very low frequency region of the measured acceleration spectrum, whereas the human motion component is located in the high frequency one. In fact, a trivial solution is to consider the gravity as a DC component, and to compute the mean value of the measured acceleration over an observation window [22]. However, more accurate results can be obtained using a Low-Pass (LP) filter to obtain gravity, and a High-Pass (HP)

filter to obtain the human motion component [4]. In this work, we propose a 5th order IIR Butterworth filter, with a cutoff frequency of 0.4 Hz. A Coupled All-Pass (CA) structure [23] has been used to implement the filter, which allows to obtain both the HP and the LP components simultaneously. The scheme of the filter is shown in Fig. 2, where A1 and A2 are the 2 all-pass filters, X(z), $Y_{LP}(z)$, and $Y_{HP}(z)$ are the frequency-domain representations of the input signal, the LP output signal, and the HP output signal respectively. The CA structure has the advantage of being highly regular, thus allowing to easily identify a fundamental cell and being well suited to an iterative implementation. From a HW point of view, this allows to save resources at the expense of a multicycle mode of operation. Also, each fundamental cell requires only one multiplier, which again is a great benefit in terms of required HW resources. The fundamental cells are highlighted by grey boxes in Fig. 2. The advantage of the CA structure with respect to other well-known filter structures is summarized in Table IV, in which we show that the CA structure allows obtaining a saving of roughly 50% in terms of number of multipliers and registers under the same filter order. With a view to implement a custom low-power and low-resources HW solution, a fixed-point (FI) coding has been used, because this allows to implement more compact arithmetic circuits with respect to their FP counterpart [24]. We determined the optimal wordlength, considering that the quantization of the filter coefficients causes a change in the frequency response with respect to the ideal one, with the possibility for the filter to become unstable. We quantified the deviation in the cutoff frequency by performing many tests using the "Fixed-Point Designer" tool in the MATLAB environment. Three different codelengths have been considered: 20 (8.12), 24 (8.16), and 28 (8.20) bits. The FP-64 bits frequency response has been considered as reference. Results show a change of the 13.75%, 0.90%, and 0.005% in the cutoff frequency for the 20 bits, 24 bits, and 28 bits coding respectively, and the stability was preserved for each case. Thus, to limit the error under 1%, we selected the 24 bits coding. During the second stage of pre-processing operations, the HP component of the measured acceleration, i.e. the human motion acceleration, is projected on a new reference frame, which is independent from the orientation of the sensor. Indeed, a tri-axial accelerometer measures the 3 components of the acceleration vector related to the *sensor-fixed* reference frame, as shown in Fig. 3. Therefore, the acceleration vector needs to be represented in the *Earth-fixed* reference frame, which is obtained by using the extracted gravity vector as reference. To this aim, the HW-friendly algorithm proposed in [24] has been used. In fact, the conventional methods for the reference frame transformation, such as Euler angles, quaternions, and Rodrigues' rotation formula, require the computation of complex mathematical functions (i.e. trigonometric functions, square root, normalizations), which in turn require an additional circuity to be efficiently computed [25]. On the other hand, the method we use allows to perform the reference frame transformation completely avoiding trigonometric functions and requiring only a square root and three divisions.

IV. Accuracy Improvement with Data Pre-Processing

A. Training results on 5 classes from PAMAP2

We demonstrate the effectiveness of the proposed pre-processing method in removing the noise deriving from the sensor orientation, by training the HNN model using pre-processed data from the PAMAP2 dataset according to the scheme explained in the previous paragraph. The results are summarized in Table V, when the 5 activities mentioned above are classified. A significant improvement is obtained thanks to the pre-processing operations when data come from sensors in *ankle* or *hand* position. The best case is for the sensor *hand16g* when the FL = 5120, where the average recall increases from 51.32% to 99.92%, thus gaining the huge value of 48.60 percentage points. On the other hand, no improvement is obtained when the sensor is in the *chest* position, and the average recall decreases from 98.28% to 89.17% in the worst case, that is when the sensor is *chest6g* and the FL = 5120, thus losing 9.11 percentage points. As mentioned before, this may be attributed to the absence of rotational movements when the sensor is located at the chest.

B. Training Results on 5 classes from PAMAP 2 for different Numbers of Neurons in the FC1 layer

Considering the very good results shown in Table V for FL = 5120, we carried out some tests on a reduced version of the HNN model. Some alternative models with a lower number of neurons for the first FC layer has been built, considering that most of the memory is required to store the weights for this layer. Results are summarized in Table VI, in which we consider 32, 16, and 8 neurons, which corresponds to a saving of the 44.9%, 67.4%, and 78.6% respectively in terms of memory requirements. Also, a saving of the 12.2%, 18.4%, and 21.4% can be obtained in terms of number of ADD/SUB operations for the 32, 16, and 8 neurons cases respectively. The saving in terms of ADD/SUB operations is lower than the one in terms of memory requirements, because the highest number of operations is associated to the second CONV layer. Results show that a high level of accuracy can be obtained with a lower number of neurons as well, especially when the sensor is located at the hand, and an average recall higher than 90% is obtained in 62.5% of the cases. The best results are obtained for the *hand6g* and *hand16g* sensors when the number of neurons of the first FC layer is equal to 32. In these two cases, an average recall of 99.99% is obtained.

C. Training Results on the 12 PAMAP2 classes

Considering that up to 12 standard activities can be selected from the PAMAP2 dataset, we also made a version of the HNN with 12 neurons in the last FC layer. In this case, the memory required to store the weights for this layer increases from 40 bytes to 96 bytes, with a total increase in the memory requirement of 5.6%. Also, the number of ADD operations increases from 320 to 768, with a total increase in the ADD/SUB operations of 1.8%. Results are summarized

in Table VII for FL = 1024, with and without pre-processing data. The number of neurons of the first FC layer is equal to 64. The HNN shows poor performance in recognizing the 12 activities from the dataset, with a reduction of roughly 50% with respect to the 5 classes case. This can be explained considering that only one accelerometer is used as input sensor in our system, and some classes cannot be distinguished with such a configuration. For instance, the activities *lying*, *sitting*, and *standing* (which are included in the PAMAP2 dataset) can only be classified by evaluating the relative position of more sensors in different parts of the body. Nevertheless, again an improvement can be observed with the use of data pre-processing when the sensor position is *ankle* or *hand*.

V. HW Architecture and Implementation Results

We designed a custom HW architecture to investigate the possibility to implement the proposed HAR system on resource-constrained devices. Two different modules implement the operations explained above, namely the pre-processing module and the HNN accelerator.

A. Pre-Processing Module

To minimize the amount of required resources, the datapath has been carefully managed to allow the implementation of the filtering operation and the reference frame transformation by using a shared circuitry. The input of the pre-processing module is the 3-axis acceleration from the sensor. The pre-processing module corresponds to the circuitry proposed in [24], which is made up of 3 parallel FI-24 Booth multipliers, and 2 FI-24 cascaded adders. This allows to perform the transformation of the reference frame by means of an iterative processing, storing the intermediate results in 500 bytes register bank. To avoid the use of additional resources, we also mapped the filtering operation on this circuitry. Considering the calculation scheme in Fig. 2, 1 multiplication and 3 sums are required to implement each fundamental cell. Consequently, this can be implemented by running 2 iterations with the pre-processing module.

B. HNN Accelerator

The HNN accelerator has been designed to reduce the amount of resources by exploiting the advantages of the HNN model. Firstly, the compactness of the model allows to strongly limit the need to access to higher levels of the memory hierarchy, which is a major issue in neural networks accelerators [26]. Local data reuse has been enhanced in the proposed architecture by an intensive use of FIFO memories. Moreover, weight binarization allows to reduce the MAC operations to ADD/SUB operations, with a significant reduction of mapped resources [27]-[29]. The block diagram of the HNN accelerator is shown in Fig. 4, where we suppose 64 neurons for the first FC layer and 5 output classes. The architecture exploits 3 cores, since this is the minimum number of cores which allows to process in parallel the 3 components of the pre-processed acceleration. A RAM is instantiated to store weights and biases of the model. As the weights are binarized, only 1 bit per weight is required. On the other hand, biases are not binarized, and 16 bits are used for their representation. The RAM dimensions are 31×696 bits, that is 2.63 KB. The most significant 15 bits of each word of the RAM are used to store the weights, therefore each core receives 5 binarized weights at each cycle. The remaining 16 bits are used for the biases. In Fig. 5 the block diagram of each core is shown. Thanks to the weight binarization, the Processing Element (PE) is a 3-levels adder tree which uses 16-bits FI arithmetic. The first level of the adder tree is made up by 3 adders, so that a dot product between vectors of length 5 can be performed in one cycle, and a bias or a result from the previous cycle can be summed up as well. "FIFO_o" stores the output activations of each layer. Each "FIFO_o" is divided in up to 5 blocks, in order to provide up to 5 different output activations in parallel to the PE. In detail, the 3 "FIFO_o" memories store the output activations of the first stage, the second CONV layer and the Max-Pool layer, respectively. Each axis is processed separately in the CONV layers, thus the memory for the output activations is locally associated to each core. As shown in Fig. 4, an external FIFO memory is also used to store the output activations of the first FC layer, since all the input activations from the previous layer cannot be separated in this case.

C. Flexible HW Design

Our design easily adapts to changes in the topology of the model, such as a change in the number of neurons of a FC layer or the number of output classes. In the first case, what changes is the RAM size, the size of the "FIFO_o" memories, and the way the Control Unit (CU) manages the architecture. Let us assume that we want to change the number of neurons of the first FC layers from 64 to 32, which leads to a lower memory requirement. In particular, the RAM size will be reduced from 31×696 to 31×395, that is from 2.6 KB to 1.5 KB. Also, the number of output activations of the third stage will be reduced from 64 to 32, thus a consequent reduction of the FIFO in Fig. 4 is required. Finally, the model will require a lower number of cycles to be computed. However, the FSM-based CU manages the transitions between layers by checking the value of a program counter. Thus, we require a slight modification in the logic of the CU to change the number of cycles associated to each layer. Another option is the change of the number of output classes. In this case, again we need to modify the number of neurons of a FC layer, namely the last FC layer. Thus, similar changes apply to this case but no change in the FIFO memory is required.

D. HW Implementation Results

We implemented this design on a Xilinx Artix-7 FPGA by using the Xilinx Vivado environment. We also realized a demo-board to prove the actual functioning of the proposed system. We deployed our design on the Digilent CMOD A7-35T [30], while the STM32F411RE [31] microcontroller is used to manage the data transfer between the sensor and the FPGA, and to display the processed results [32]. In Table VIII the results of the FPGA implementation and a comparison with the state of the art for HW HAR solutions is

provided. Results refers to an HNN configuration with 64 neurons in the first FC layer and 5 output classes. The operating frequency (OpFreq) is the minimum frequency that allows real-time operations. Since the number of cycles required to obtain a class prediction is 12600, and the Output Data Rate (ODR) of the sensor is equal to 25 Hz, OpFreq is 12600×25 Hz = 315 kHz. The power consumption has been estimated by using the Vivado power tool. To obtain a high level of confidence, we generated Switching Activity Interchange Format (SAIF) files from post-implementation simulations. At the OpFreq, the total power consumption is 72.04 mW, which mostly corresponds to the static power consumption of the FPGA. A power reduction of 37.9% and 70.1% can be observed with respect to [33] and [34] respectively. The maximum operating frequency is 41 MHz, which allows to set a maximum ODR for the sensor of 3.2 kHz. This bodes well for the use of our system in applications where a higher throughput is required, such as fault diagnosis [35]. 5988 LUTs, 4299 FFs and 1 BRAM are required to implement the design. The number of LUTs and FFs is higher than the ones in [33] and [34]. However, 3 DSPs and 14 BRAMs are used in [33], whereas 81 DSPs are used in [34], and we need none.

VI. Conclusions

In this work, a pre-processing scheme enhances the accuracy of a reduced-precision neural network. We propose a HW implementation of the system, which suits real-time operation and provides very high accuracy. Results show a very low use of resources and a low power consumption. Thus, future works will be aimed to silicon implementation, which can bring down the power consumption to µW and enable the integration in wearable or portable devices.

References

[1] O. D. Lara and M. A. Labrador, "A Survey on Human Activity Recognition using Wearable Sensors," in *IEEE Communications Surveys & Tutorials*, vol. 15, no. 3, pp. 1192-1209, Third Quarter 2013.

[2] B. M. Eskofier et al., "Recent machine learning advancements in sensor-based mobility analysis: Deep learning for Parkinson's disease assessment," *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, Orlando, FL, 2016, pp. 655-658.

[3] I. Bisio, A. Delfino, F. Lavagetto and A. Sciarrone, "Enabling IoT for In-Home Rehabilitation: Accelerometer Signals Classification Methods for Activity and Movement Recognition," in *IEEE Internet of Things Journal*, vol. 4, no. 1, pp. 135-146, Feb. 2017.

[4] T. R. Bennett, J. Wu, N. Kehtarnavaz and R. Jafari, "Inertial Measurement Unit-Based Wearable Computers for Assisted Living Applications: A signal processing perspective," in *IEEE Signal Processing Magazine*, vol. 33, no. 2, pp. 28-35, March 2016.

[5] H. Yu, S. Cang, and Y. Wang, "A review of sensor selection, sensor devices and sensor deployment for wearable sensor-based human activity recognition systems," *2016 10th International Conference on Software, Knowledge, Information Management & Applications (SKIMA)*, Chengdu, 2016, pp. 250-257.

[6] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015.

[7] J. Wang, Y. Chen, S. Hao, X. Peng, L. Hu, "Deep learning for sensor-based activity recognition: A Survey," *Pattern Recognition Letters*, 2018, ISSN 0167-8655.

[8] A. De Vita *et al*., "Low-Power Integrated Circuit for Orientation Independent Acquisitions from Smart Accelerometers," in *Lecture Notes in Electrical Engineering*, , vol. 629, pp. 35-41, Feb. 2020.

[9] J. Pagán *et al*., "Toward Ultra-Low-Power Remote Health Monitoring: An Optimal and Adaptive Compressed Sensing Framework for Activity Recognition," in *IEEE Transactions on Mobile Computing*, vol. 18, no. 3, pp. 658-673, 1 March 2019.

[10] Z. Zou, Y. Jin, P. Nevalainen, Y. Huan, J. Heikkonen and T. Westerlund, "Edge and Fog Computing Enabled AI for IoT-An Overview," *2019 IEEE International Conference on Artificial Intelligence Circuits and Systems (AICAS)*, Hsinchu, Taiwan, 2019, pp. 51-56.

[11] T. Simons, and D. J. Lee, "A Review of Binarized Neural Networks," in *MDPI Electronics*, vol. 8, no. 6, pp. 661-686, June 2019.

[12] H. Alemdar, V. Leroy, A. Prost-Boucle and F. Pétrot, "Ternary neural networks for resource-efficient AI applications," *2017 International Joint Conference on Neural Networks (IJCNN)*, Anchorage, AK, 2017, pp. 2547-2554.

[13] IEEE Standard for Floating-Point Arithmetic," in IEEE Std 754-2008 , vol., no., pp.1-70, 29 Aug. 2008.

[14] M. Courbariaux, I. Hubara, D. Soudry, R. El-Yaniv, and Y. Bengio, "Binarized Neural Networks: Training Deep Neural Networks with Weights and Activations Constrained to +1 or -1," arXiv preprint arXiv: 1602.02830 (2016).

[15] A. Prost-Boucle, A. Bourge, and F. Pétrot. 2018. "High-Efficiency Convolutional Ternary Neural Networks with Custom Adder Trees and Weight Compression," in *ACM Transactions on Reconfigurable Technology and Systems*, vol. 11, no. 3, art. 15, December 2018.

[16] D.P. Pau, E. Plebani, F.G. De Ambroggi, F. Guido, A. Bosco "Recognition method, corresponding system and computer program product," - US Patent App. 16/189,264, 2019.

[17] A. Reiss and D. Stricker, "Introducing a new benchmarked dataset for activity monitoring," in *Proc. 16th Int. Symp. Wearable Comput. (ISWC)*, Jun. 2012, pp. 108–109.

[18] "Welcome to Lasagne", 2015. [Online]. Available: https://lasagne.readthedocs.io/en/latest/index.html#

[19] S. Ioffe and C. Szegedy, "Batch normalization: accelerating deep network training by reducing internal covariate shift," in *ICML'15: Proceedings of the 32nd International Conference on Machine Learning*, vol. 37, Lille, 2015, pp. 448-456.

[20] A. Bulling, U. Blanke, and B. Schiele, "A tutorial on human activity recognition using body-worn inertial sensors," in *ACM Computing Surveys*, vol. 46, no. 3, Article no. 33 , Jan. 2014.

[21] M. Courbariaux, I. Hubara, D. Soudry, R. El-Yaniv, and Y. Bengio, "Binarized Neural Networks: Training Deep Neural Networks with Weights and Activations Constrained to +1 or -1," arXiv preprint arXiv: 1602.02830 (2016).

[22] D. Mizell, "Using gravity to estimate accelerometer orientation," *Seventh IEEE International Symposium on Wearable Computers, 2003. Proceedings.*, White Plains, NY, USA, 2003, pp. 252-253.

[23] P. Vaidyanathan, S. Mitra and Y. Neuvo, "A new approach to the realization of low-sensitivity IIR digital filters," in *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 34, no. 2, pp. 350-361, April 1986.

[24] A. De Vita, G. D. Licciardo, A. Femia, L. Di Benedetto and D. Pau, "µW Pre-processing Unit for Virtual Sensors Based on Tri-axial Smart Accelerometers," *2019 17th IEEE International New Circuits and Systems Conference (NEWCAS)*, Munich, Germany, 2019, pp. 1-4.

[25] A. De Vita, G. D. Licciardo, A. Femia, L. Di Benedetto, A. Rubino and D. Pau, "Embeddable Circuit for Orientation Independent Processing in Ultra Low-Power Tri-Axial Inertial Sensors," in *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 67, no. 6, pp. 1124-1128, June 2020.

[26] V. Sze, Y. Chen, T. Yang and J. S. Emer, "Efficient Processing of Deep Neural Networks: A Tutorial and Survey," in *Proceedings of the IEEE*, vol. 105, no. 12, pp. 2295-2329, Dec. 2017.

[27] G. D. Licciardo, C. Cappetta, L. Di Benedetto, A. Rubino and R. Liguori, "Multiplier-Less Stream Processor for 2D Filtering in Visual Search Applications," in *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 1, pp. 267-272, Jan. 2018

[28] G. D. Licciardo, C. Cappetta, L. Di Benedetto and M. Vigliar, "Weighted Partitioning for Fast Multiplierless Multiple-Constant Convolution Circuit," in *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 64, no. 1, pp. 66-70, Jan. 2017.

[29] G. D. Licciardo, C. Cappetta and L. Di Benedetto, "FPGA optimization of convolution-based 2D filtering processor for image processing," *2016 8th Computer Science and Electronic Engineering (CEEC)*, Colchester, 2016, pp. 180-185, doi: 10.1109/CEEC.2016.7835910.

[30] Xilinx, "7 Series FPGAs Data Sheet: Overview," XC7A35T-1CPG236C datasheet, Feb. 2018. [Online]. Available: https://www.xilinx.com/support/documentation/data_sheets/ds180_7Series_Overview.pdf.

[31] STMicroelectronics, "STM32F411xC STM32F411xE," Dec. 2017. [Online].Available: ttps://www.st.com/resource/en/datasheet/stm32f411re.pdf

[32] A. D. Vita, D. Pau, C. Parrella, L. D. Benedetto, A. Rubino and G. D. Licciardo, "Low-Power HWAccelerator for AI Edge-Computing in Human Activity Recognition Systems," *2020 2nd IEEE International Conference on Artificial Intelligence Circuits and Systems (AICAS)*, Genova, Italy, 2020, pp. 291-295,

[33] 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," in *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 66, no. 1, pp. 274-287, Jan. 2019.

[34] N. B. Gaikwad, V. Tiwari, A. Keskar and N. C. Shivaprakash, "Efficient FPGA Implementation of Multilayer Perceptron for Real-Time Human Activity Classification," in *IEEE Access*, vol. 7, pp. 26696-26706, 2019.

[35] S. Zhang, S. Zhang, B. Wang and T. G. Habetler, "Deep Learning Algorithms for Bearing Fault Diagnostics - A Review," *2019 IEEE 12th International Symposium on Diagnostics for Electrical Machines, Power Electronics and Drives (SDEMPED)*, Toulouse, France, 2019, pp. 257-263.