An Ultra-Low Power, “Always-On” Camera Front-End for Posture Detection in Body Worn Cameras Using Restricted Boltzman Machines

Soham Jayesh Desai, Mohammed Shoaib, *Member, IEEE*, and Arijit Raychowdhury, *Senior Member, IEEE*

Abstract—The Internet of Things (IoTs) has triggered rapid advances in sensors, surveillance devices, wearables and body area networks with advanced Human-Computer Interfaces (HCI). One such application area is the adoption of Body Worn Cameras (BWCs) by law enforcement officials. The need to be ‘always-on’ puts heavy constraints on battery usage in these camera front-ends, thus limiting their widespread adoption. Further, the increasing number of such cameras is expected to create a data deluge, which requires large processing, transmission and storage capabilities. Instead of continuously capturing and streaming or storing videos, it is prudent to provide “smartness” to the camera front-end. This requires hardware assisted image recognition and template matching in the front-end, capable of making judicious decisions on when to trigger video capture or streaming. Restricted Boltzmann Machines (RBMs) based neural networks have been shown to provide high accuracy for image recognition and are well suited for low power and re-configurable systems. In this paper we propose an RBM based “always-on” camera front-end capable of detecting human posture. Aggressive behavior of the human being in the field of view will be used as a wake-up signal for further data collection and classification. The proposed system has been implemented on a Xilinx Virtex 7 XC7VX485T platform. A minimum dynamic power of 19.18 mW for a target recognition accuracy while maintaining real time constraints has been measured. The hardware-software co-design illustrates the trade-offs in the design with respect to accuracy, resource utilization, processing time and power. The results demonstrate the possibility of a true “always-on” body-worn camera system in the IoT environment.

Index Terms—Algorithms implemented in hardware, object recognition, reconfigurability, wearable computers

1 Introduction

The “Internet of Things” represents a paradigm shift in the interconnected world, leading to communication among various physical entities around us. At the same time these devices are expected to possess sufficient intelligence to be able to assimilate, analyze and process data. Constraints due to battery life and storage capacity make it imperative to have a smart front-end capable of making decisions regarding the relevance and importance of the image, before storing or transmitting it. Recently there has been an increased interest for the use of Body Worn Cameras (BWCs) for law enforcement. Automatic recognition of human actions and postures is a key enabler for both video surveillance and Body-Worn Cameras.

BWCs are gaining traction both commercially and from the law enforcements’ point of view. Multiple pilot programs are being conducted for BWCs, including those in Mesa, Arizona, in the United States [1], Plymouth, United Kingdom [2]. These studies have highlighted the potential of such video cameras to capture much more compelling evidence and also act as a deterrent to crime. These also highlight benefits such as increase in accountability and transparency. However, short battery life, limited storage capacity [3] as well as the need for a human operator to analyze the data, limit wide-spread adoption of the BWCs. Since data is analyzed off-line, it cannot be used for triggering affirmative action such as alerting law enforcement. To enhance battery life, the current cameras are manually turned on and off, which defeats the purpose of ‘always-on’ sensing. Hence, in an ‘always on’ camera front-end it is desired to enable ‘smartness’ such that the camera would be able to make intelligent and judicious decisions on when to start storing a video stream while at the same time providing a metric of human aggressiveness in the field of view. Aggressiveness is associated with human posture and hence, we propose a hardware assisted camera front-end capable of detecting human posture and identifying relevant ‘information’ in incoming video stream. To enable ultra-low power operation, the hardware architecture needs to be co-optimized with the algorithm as well as the frame-rate, data resolution and accuracy targets. Fig. 1a illustrates the proposed system. An alternative to an intelligent camera front end is to continuously capture data and either store or wirelessly transmit it. Figs. 1b and 1c illustrate the ‘energy cost’ of an H.264 encoder and transmitter. It illustrates the prohibitive cost (hundreds of mW to ~1W) of digital processing (on a GPU, ASIC and near-threshold voltage ASIC) which makes such a continuous time system unrealizable.

In this paper, we explore a camera front-end with Restricted Boltzmann Machine (RBM) based Artificial Neural Network (ANN) as the recognition and classification engine. When cascaded to the data acquisition (pixel array and

- *S.J. Desai and A. Raychowdhury are with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332. E-mail: sdesai1@gatech.edu, arijit.raychowdhury@ece.gatech.edu.*
- *M. Shoaib is with the Microsoft Research, Microsoft Corporation, Redmond, WA 98052-6399. E-mail: mohammed.shoaib@microsoft.com.*

Manuscript received 11 May 2015; revised 22 Sept. 2015; accepted 20 Nov. 2015. Date of publication 30 Dec. 2015; date of current version 27 Jan. 2016. Recommended for acceptance by S. Ray, J. Park, and S. Bhunia.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TMSCS.2015.2513741

2332-7766 © 2015 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

Authorized licensed use limited to: Georgia Institute of Technology. Downloaded on July 04,2021 at 02:48:31 UTC from IEEE Xplore. Restrictions apply.

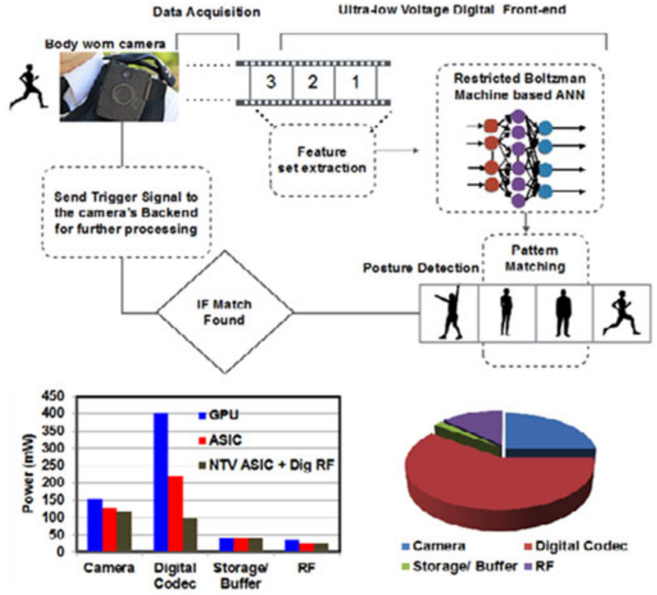

Fig. 1. (a) Usage model for a typical ‘always-on’ BWC. (b) The different components of power dissipation in a state-of-the-art camera based sensor node with continuous wireless transmission. (c) Breakdown of power illustrating a large section of the total dissipation in the digital codec (H.264).

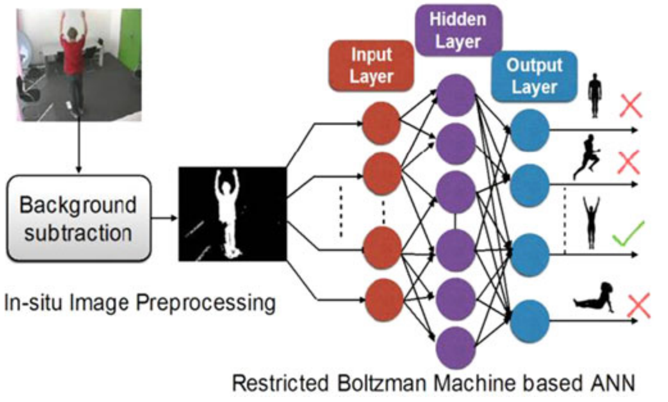
Hidden Layer
Input Layer
Output Layer
Background subtraction
In-situ Image Preprocessing
Restricted Boltzman Machine based ANN

Fig. 2. The recognizer flow highlights the layers of the restricted Boltzmann machine and the pre-processing unit. The output layer is designed as winner-take-all and the posture with highest probability is chosen.

analog-to-digital converters) unit, it can allow ultra-low power video capture as well as intelligent data assimilation. We demonstrate the efficacy of the system illustrated in Fig. 2 in recognizing human posture from the ‘Weizmann Human Actions Silhouette database’ with high accuracy while maintaining processing time constraints due to real time requirements and at a fraction of the power cost. The hardware has been implemented on Xilinx Virtex 7 XC7VX485T. By careful co-optimization between algorithm and hardware, we enable ‘always on’ sensing and recognition of 19.18 mW (excluding the power of the signal acquisition unit and the background subtraction unit. The background extraction performed in ASIC has shown to consume around 27.88uW/pixel [4].) This illustrates an order of magnitude improvement in: (1) power efficiency for ‘always on’ camera based wireless sensor nodes, which continuously capture and transmit data and (2) significant savings in storage space for systems with continuous time capture and storage.

2 Restricted Boltzmann Machine Based Recognition and Posture Detection

An ‘always-on’ smart BWC needs to be equipped with low-power hardware capable of detecting certain human posture when trained. Recent progress in Deep Neural Networks illustrates the efficacy of using neuromorphic systems in providing high accuracy even under acquisition noise and image occlusion. However very deep networks such as in [5] are not suitable for our application because: (1) such networks require tens to hundreds of thousands of neural processing units, or nodes which are typically executed in many-core servers and distributed machines (2) the power cost of such networks in prohibitive in a mobile platform and (3) they are not suitable for real-time applications. On the other hand, the accuracy targets can be relaxed from that of very deep networks based on the need of the application. The accuracy of the network is currently limited by the availability of labelled training cases in the database and can be improved further upon by optimizing the offline training, increasing the size of the training the dataset, and increasing the representational power of the network. Our target is achieving accuracies close to 90 percent (with minimum number of false rejects and limited by the size of dataset), meeting real time processing requirements, with tens of mW of power consumption and also providing capability for reconfigurability. This is almost three orders of magnitude reduction of power when compared to very deep networks and would enable true mobility. Hence we adopt Restricted Boltzmann Machine based Artificial Neural Networks (ANNs) as the algorithmic and hardware design paradigm for ultra-low power recognition. Restricted Boltzmann Machine based recognizers are probabilistic graphical models (which form the basis of deeper networks). RBMs are modular, scalable and can be efficiently mapped to hardware with well-controlled data movement between logic and embedded memory. RBMs allows us to re-use the same resources via time multiplexing because of their modularity and Single Instruction Multiple Data (SIMD) nature. We also provide an option of increasing the network depth, for potentially higher accuracy at run time and compile time and also cascading of neural networks which amounts to storing different sets of weights. We select the RBM as compared to a Support Vector Machine [6] to allow such reconfigurability, allowing deeper network with near best in class accuracies and also allowing us to partially train the network in an unsupervised manner, without putting emphasis on the feature extraction as has been shown in [7]. Most of the previous work for SVMs and Deep nets have explored image nets and MNIST, although posture detection itself is not new, the available datasets are limited and will improve in the near future.

2.1 Mathematical Description

The basic RBM consists of two layers, an output visible layer “V” representing the observable data and a hidden layer “H” which portrays the internal representation of the observable data into the system. These layers are comprised of processing elements referred to as Neurons or nodes. RBMs form a special category of Boltzmann Machines where these two layers form a bipartite graph. There are no connections between the hidden neurons. Each hidden unit describes a probability distribution over the inputs

Authorized licensed use limited to: Georgia Institute of Technology. Downloaded on July 04,2021 at 02:48:31 UTC from IEEE Xplore. Restrictions apply.

provided by the visible layer units. Further, the hidden layer provides a higher level of feature set for the input data and enables associativity between a set of observable outputs and control inputs. Using the following notation: $V = (V_1 \ldots V_m)$ representing the Visible input units, $H = (H_1 \ldots H_n)$ representing the Hidden Neurons, and the random variables V and H take binary values (v,h). The joint probability distribution for both the layers is given by the Gibbs Distribution [8]

$$p_\theta(v, h) = 1/z(\theta)^* e^{-E_\theta(v,h)}. \tag{1}$$

Here $\theta = \{w, b, c\}$ are the model parameters and refer to weights and biases in the network. $z(\theta)$ is the partition function seen in Boltzmann distributions, a normalizing constant and equals the sum of energy exponentials for all possible network configurations. Here the Energy function is given by

$$E(v, h) = -\sum_{\mathrm{i}} \sum_{\mathrm{i}} w_{ii} h_i v_i - \sum_{i} b_i v_i - \sum_{\mathrm{i}} c_i h_i. \tag{2}$$

The j and i sum over all the nodes in the visible layers and hidden layer respectively. $\mathrm{w_{ij}}$ represents real valued weights across the edge between the jth visible node and ith hidden node. $\mathrm{b_i}$ and $\mathrm{c_j}$ represent the real valued bias terms associated with the jth visible node and ith hidden node respectively.

Based on this energy, it can be shown [8] that the conditional probability of any unit being 1 can be written as

$$P(H_i = 1 \,|\, v) = sig\left(\sum_j w_{ij} v_j + c_i\right), \tag{3}$$

$$P(\mathrm{V}j = 1 \,|\, \mathrm{h}) = sig\left(\sum_i w_{ij} h_i + \mathrm{b}_i\right). \tag{4}$$

Here sig refers to the sigmoid function. These equations show that an RBM can be reinterpreted as a standard feedforward neural network with one layer of non-linear processing units.

2.2 Training in RBMs

The weights need to be modified such that the RBM produces the minimum energy across the training set of observable data. The accurate calculation of the log-likelihood gradient is computationally prohibitive. We follow the method provided in [8] for approximating the RBM log-likelihood gradient namely, “Contrastive Divergence” which was originally described in [8] and applied in [9]. Obtaining unbiased estimates of the log-likelihood gradient using Markov Chain Monte Carlo methods typically requires many sampling steps. In [8], the authors show that estimates obtained after running the chain for just a few steps can be sufficient for model training. We follow the training algorithm described in [8] for training the RBM. Since our application is ‘posture detection’ in BWCs, we perform off-line training on sample data-set using MATLAB and then transfer the weights to the Xilinx compiler using “Memory Initialization Files”. The training set is divided into mini-batches. We set the learning rate to provide us with a target recognition accuracy. The RBM is trained in an unsupervised manner using Contrastive Divergence. The features generated by the RBM are used to train the classifier. Since our input (pixel data) is real valued and not binary, we scale them to [0, 1] and treat them as probabilities [10]. As per [10], the learning process remains the same. Classification, in contrast to training, just involves a forward pass. We treat the input data as a vector and multiply this vector with corresponding trained weights along the edges of the networks. Since the network forms a bipartite graph, this is a vector – matrix multiplication followed by an application of the sigmoid non-linearity to generate the hidden node representation. A similar method is applied for the classification layer. Deep Networks can be trained in a similar method, greedily layer by layer, as described by [11].

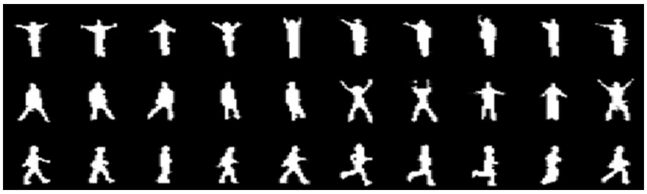

Fig. 3. The recognizer flow highlights the layers of the restricted Boltzmann machine and the pre-processing unit. The output layer is designed as winner-take-all and the posture with highest probability is chosen.

2.3 Image Database for Posture Detection

The experiments are carried out on the Weizmann human silhouette based action database [12] (Fig. 3). The database consists of video sequences (180 × 144, de-interlaced 50 fps) of nine different actors, each performing ten different actions such as “bending”, “jumping-jack”, “jumping forward-on-two-legs”, “jumping-in-place-on-two-legs”, “running”, “galloping-sideways’’, “waving-with-one-hand”, “waving with-two-hands”. To obtain the silhouettes, we perform background subtraction as described in [4]. These silhouettes are aligned and the training of the neural network is performed using these aligned silhouettes. It is interesting to note that with the popularity and deployment of BWCs, this database is evolving and better training sets are expected in the near future. Different postures from the Weizmann database correspond to basic human postures and are applicable to BWCs. For example, ‘putting both hands up’ is treated as a defensive posture while ‘running’ is treated as aggressive behavior. Once proper posture identification is enabled, the output can be used for further action detection as the situation and usage demands. Various other attributes such as sound, location, etc can be user to provide context. However, human recognition and posture identification are key primitives that can enable ‘always-on’ BWCs for law enforcement.

3 Hardware Infrastructure

To meet the extreme power constraints in ‘always-on’ BWCs, custom hardware architecture is required. We have implemented the proposed algorithm on a Xilinx Virtex 7 XC7VX485T platform. Before discussing the efficacy of the RBM based ANNs in posture detection, we explore the design implementation on the hardware platform and discuss software-hardware co-design for maximum power efficiency at a target accuracy rate. Our proposed design comprises of the camera front-end hardware used for image sensing and conversion into the raw pixel data, followed by the silhouette extraction unit through background subtraction. The algorithm and hardware implementation is straightforward and has been discussed in [13], [14], [15].

Authorized licensed use limited to: Georgia Institute of Technology. Downloaded on July 04,2021 at 02:48:31 UTC from IEEE Xplore. Restrictions apply.

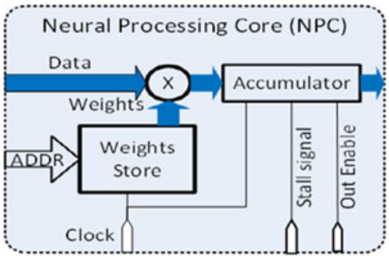

Fig. 4. Block diagram of the neuron processing core showing the incoming and outgoing control signals, the data path, and the address bus.

3.1 Hardware Design of the Recognizer

The motivation for RBM based ANNs results from modelling synaptic behavior of human neurons, by computing the function:

$$\sum_i W_{ip}X_{ip} + \theta_p. \tag{5}$$

In Equation (5), W_{ip} represents the synaptic weights, X_i represents the input feature to the neuron, θ_i is the bias for the pth neuron and these are summed over all the input image features. Each Neuron in our network models the computation represented by (5). We call the instantiation of this neuron in hardware as the neuron processing core (NPC) illustrated in Fig. 4. The NPC comprises of a fixed point signed multiplier, accumulator, and memory for storing the weights. The weights are stored as distributed memory within each neuron core. A hidden layer in a neural network comprise of many such neurons performing a similar computation as (3) but with a different set of weights and biases. Similarly, in hardware many such NPCs are grouped together to form a layer. The input to each layer is provided in parallel to all the NPCs within the Layer.

The inherent parallelism of such a neural network results from the fact that, in a fully connected network, the computation in (5) is carried out by all the neurons in parallel for an input feature X_i. Ideally to obtain the least processing time we would desire as many NPCs in parallel as the number of neurons in the hidden layer. This results in very high resource utilization and consequently greater overall power and area. We provide the capability to reuse these NPCs by time multiplexing for computations belonging to the same layer. We differentiate between these as "Virtual" and "Physical" NPCs. Physical NPCs are instantiated in the physical design and consume resources. Virtual NPCs represent the actual number of hidden neurons in a layer for a particular network configuration. The ratio of Virtual NPCs and the Physical NPCs gives you the number of "phases" or the number of times these NPCs need to re-execute so that the computation for the layer gets completed shown in Fig. 5. The most serialized case comprises of a single Physical NPC executing as many times as the number of Virtual NPCs in the layer, resulting in the least amount of resource utilization and power but also a significantly higher processing time. In 'always-on' microphone-based audio sensors, such serialization of parallel workload has been shown to be effective in reducing the overall system power [16]. For a given computational complexity at a frame rate of 30 fps, a lower number of 'physical NPCs' demonstrate a favorable trade-off between power and the total computational time.

The layer as described by Fig. 6 also consists of a sigmoid approximating unit, a control unit, a bus arbitration unit and a first-in, first-out (FIFO). Direct implementation of a sigmoid unit is expensive in hardware and increases the power and the processing time. We approximate the sigmoid using a piece-wise linear approximation. We opt for a distributed control unit so that the computation of layers remains as independent from each other as possible. The control unit provides the address for the weights, communicates with other layers and provides control signals to the NPCs, bus arbitration unit and the FIFO. A bus arbitration unit is required to serialize the neuron outputs generated and store it in the output FIFO, before the next computation of the layer can take place.

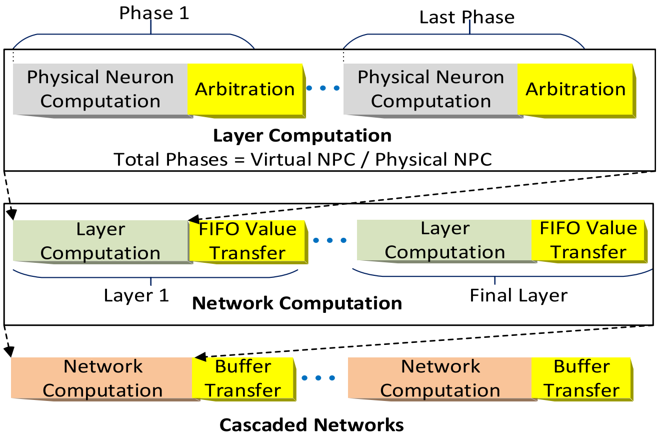

Fig. 5. Depicts the virtualization of neuron computation, layer computation, and the entire network. The physical NPCs are reused for a count equal to total phases so as to compute for all the virtual NPCs. The layer can be reused to provide an increase in depth of the network. Similarly, the entire network can then be reused for a different purpose or even for recognizing the same image with higher accuracy.

We pipeline the layers using a FIFO. The FIFO status signals are used for the communication between the layers. If any succeeding stage is still processing the data, the preceding layer is stalled from transferring the values from its output FIFO. The system is thus a pull-based pipelined system. To allow multiplexing the FIFO length equals the number of Virtual NPCs. Similar to the concept of re-using the NPCs, we provide the capability of reusing the entire layer by allowing

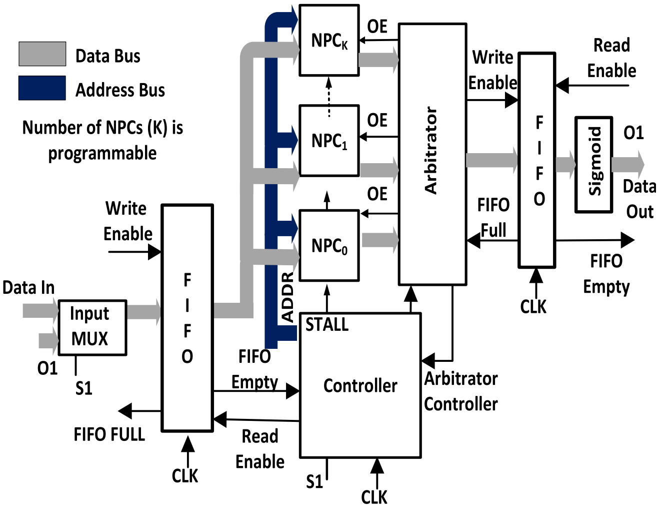

Fig. 6. The block design of a single layer showing the control and data flow. O1, the output of the layer or input back into the layer if selected by multiplexer. S1, a control signal from the controller is used for selecting the input.

Authorized licensed use limited to: Georgia Institute of Technology. Downloaded on July 04,2021 at 02:48:31 UTC from IEEE Xplore. Restrictions apply.

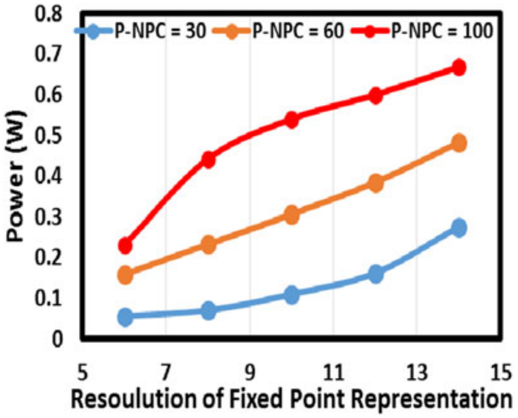

Fig. 7. Power versus Fractional bit resolution for three different network configurations with different number of physical NPCs.

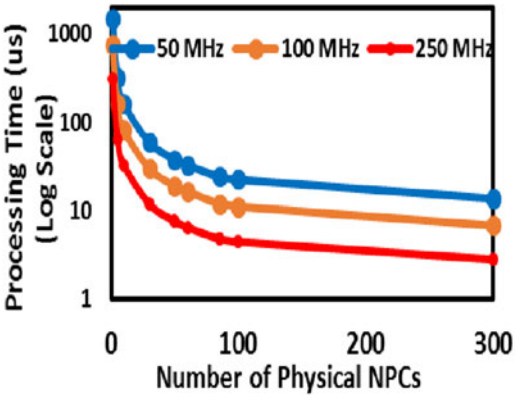

Fig. 8. Describes the increase in processing time as the parallelization is reduced by reusing the physical NPCs for computation so as to save resources and power.

the output FIFO to feed data back as input. This path is multiplexed with the original input path. The end of the network consists of a final layer, which comprises of a store buffer, counter and a comparator in addition to the NPCs, FIFO and the control unit. The store buffer is used for storing the largest value read from the FIFO. The counter keeps track of the output number of the NPC because this corresponds to the classification label. Input is serially fed from the FIFO into the comparator and signed compared with the value stored in the store buffer. The store register and the counter are updated if the input value is greater than the store buffer. It is beneficial to keep final layer as parallel as possible, since it allows us to avoid the replay of outputs from the previous layer. The weights, input features and the data transferred are represented using a signed fixed-point notation of Q2.6. Fixed point data representation of resolution more than Q2.6 show no impact on recognition accuracy over a floating point representation, detailed in the sections ahead. This results in significant cost savings with respect to resource utilization and power. The accumulator output buffer resolution is kept significantly greater than the resolution of the input to the accumulator to prevent any overflows which impacts the accuracy of the network severely. The entire design is made configurable by extensively parameterization. This allows us to perform design space exploration where the role of these parameters on performance, power and resource utilization can be studied for design optimization. More details are provided in Section 5. The configurable parameters comprise of the following:

1) Fixed Point Data Resolution:
2) Number of Input features
3) Number of Virtual and Physical NPCs
4) Number of Virtual or Physical Hidden Layers
5) Frequency of the clock

Run Time Configurability is of significant interest since it allows the same hardware to be used for different applications or for allowing the processing of a cascade of neural networks by the same hardware. For example in the case of BWCs, the front end can be made to primarily detect humans as a measure of relevancy of a frame followed by posture recognition using a deeper network at the back end, all done using a single hardware component. Reusing physical NPCs, layers and also networks as a whole, facilitates the above. Hardware reuse and re-evaluating the network in phases thus allows our hardware to be extended to achieve run time configurability requirement as shown in Fig. 5. This does result in increased storage requirements for storing multiple sets of weights for multiple networks.

4 Design Space Exploration

To minimize the overall network power and utilization at real time and accuracy performance constraints, we jointly optimize algorithms and hardware. We choose 8 bits for data representation. The number of virtual NPCs is chosen as 300, based on the redundancy in input features, accuracy results and following the guidance in [17]. We explore the entire design space of power and the bit width of data representation as a function of the total number of physical NPCs (Fig. 7). We observe that increasing the number of NPCs increase the total power dissipation but results in faster compute (Fig. 8). Further, the power increases rapidly with the data width. Finally it is important to understand the impact of serialization to the total energy cost of the design (i.e., the energy required to compute per frame). Fig. 9 illustrates the total energy cost of the design as a function of the number of physical NPCs when operated at 50 MHz For a large number of NPCs, the total (leakage and dynamic) power increases whereas for a small number of

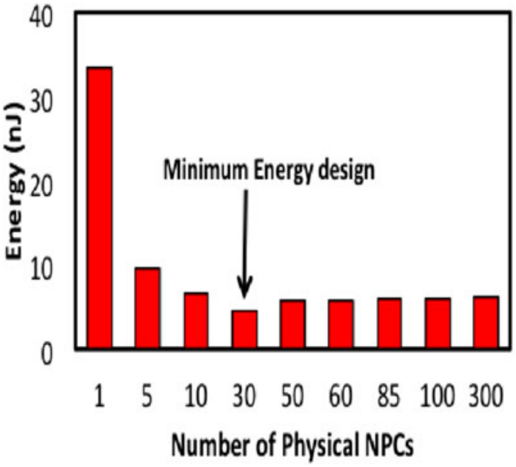

Fig. 9. Total energy/frame as a function of the number of physical NPCs. The 'Minimum Energy' design is obtained for 30 physical NPCs running at a clock frequency of 50 MHz. The processing time is measured from the first input pixel to the classification result.

Authorized licensed use limited to: Georgia Institute of Technology. Downloaded on July 04,2021 at 02:48:31 UTC from IEEE Xplore. Restrictions apply.

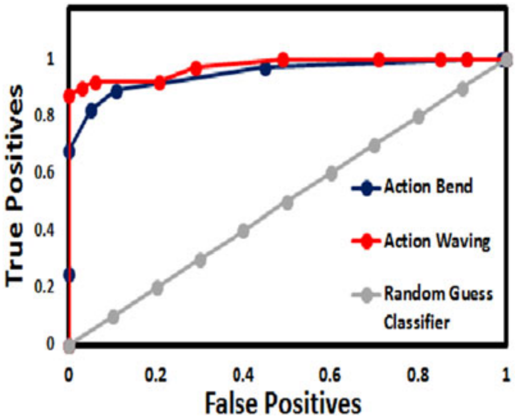

Fig. 10. ROC Curve. Classification threshold incremented by 0.1 from 0 to 1. Due to multiple classes, ROC is drawn by taking one action class and against a group of all other classes. Two actions selected out of 10 so that plots are visible. True positives and false positives can be viewed as probability rate.

NPCs the total data movement and time to process increases rapidly (Fig. 8). The point of minimum energy is measured for 30 physical NPCs (with 300 virtual NPCs). This illustrates the need for hardware-software co-design & by joint optimization of the accuracy-energy-resource utilization space, an optimum design point is attained. At this design point, we obtained a processing time for classification of a single input frame of 58.66 us. The size of the input frame is 3,072 pixel features. We note that this is less than 5nJ of energy/frame for processing. This illustrates three orders of magnitude improvement in total power (19.18 mW) compared to a camera based wireless sensor node.

5 Expiremental Results

Our main goal is to study the tradeoffs of power, timing and resource utilization with different network configurations and also the resolution of the data within the network. The configurable parameters in our design consist of size of input features, number of virtual and physical NPCs, number of virtualized hidden layers and physical hidden layers, fixed point data resolution and frequency. The FPGA platform used for measurements is the Xilinx Virtex-7 XC7VX485T. Software based simulations are used to train the network weights. The weights are then extracted and fed into the FPGA platform at compile time as memory initialization files. The baseline Neural Network configuration selected for experimentation is a shallow network comprising of 256 feature inputs, 300 virtual NPCs, and 30 physical NPCs for hidden layer 1 and a final layer comprising of 10 NPCs corresponding to 10 silhouette actions. The RBM at each layer is separately trained using contrastive divergence, and the final layer is trained as multinomial logistic classifier using the features provided by the RBM layers below. We do not perform back-propagation to further tune the parameters, because it may cause over fitting since the labelled data of the Weizmann database is limited.

Power Measurement is performed using the Xilinx Vivado Power Analysis. During design development we use vector based estimation for the synthesized design, taking the switching activity into consideration from simulation results and providing also the control activity expected leading to more accurate results. Energy is calculated by measuring the average power and time for classification for the input frame.

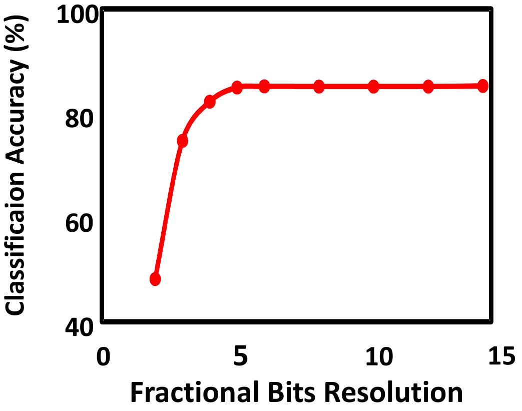

Fig. 11. Classification accuracy with varying fixed point representation resolutions. The integer bits kept constant at 1. We make sure there is low probability of overflow by having higher bit width for the accumulator.

5.1 Algorithmic Accuracy

Fig. 10 illustrates a Receiver Operating Characteristic (ROC) curve for the classifier for two different actions. The performance of the classifier can be compared with the random classifier by looking at the Area under the curve. Since we do not want the BWC to miss a relevant event, our objective is to set the threshold conservative, aiming to remove false negatives. We would like to highlight that the accuracy of the classifier can be enhanced by increasing the number of labelled training data, performing pre-processing for better features, increasing the capability of the network to model the input set for example by increasing the depth of the network and by improving the training parameters. Training to achieve high accuracy can be carried offline and separately and is not the focus of this paper Fig. 11 illustrates the dependence of recognition accuracy on the bit width of the data representation. With a fractional bit width of 6 (Q2.6 format) and avoiding overflow while accumulating, the accuracy tends to that of a floating point representation and has been chosen for our design. This results in lower design complexity and power without compromising the accuracy of recognition.

5.2 Hardware Measurement Results

The most important design criteria is the choice of the number of physical NPCs. As seen in Fig. 12, resource utilization of the network can be improved by reducing the number of physical

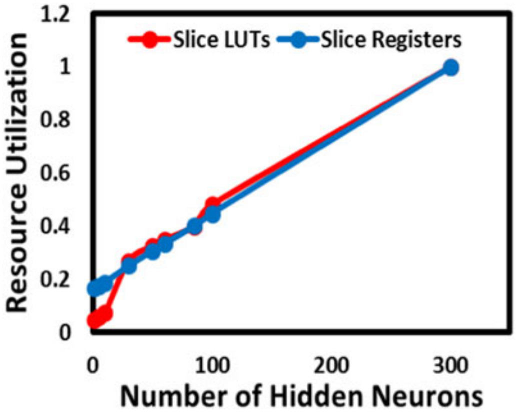

Fig. 12. Describes the normalized resource utilization for increase in parallelization by increasing the number of physical NPCs. The normalization is with respect to the resource utilization of NPC = 300 (Slice LUT = 97,572, Slice registers = 29,582).

Authorized licensed use limited to: Georgia Institute of Technology. Downloaded on July 04,2021 at 02:48:31 UTC from IEEE Xplore. Restrictions apply.

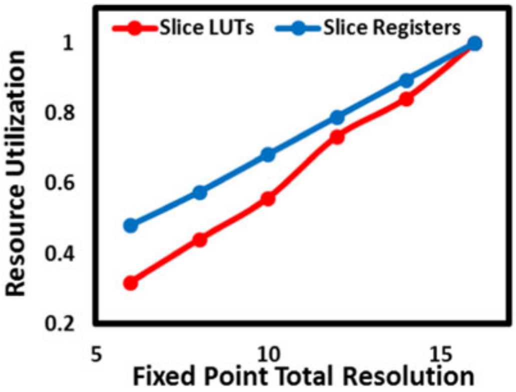

Fig. 13. Normalized resource utilization versus fractional bit resolution. Integer bits kept constant. Normalization is carried with respect resource utilization of 16 bits resolution (Slice LUTs = 35,787, Slice registers = 9,437).

NPCs. This however results in an increase in the processing time. It should however be noted, that at 30 fps, amount of time available for processing the data is sufficient even for a small number of physical NPCs. The choice of data bit width has significant impact on the resource utilization of the network and is shown in Fig. 13, which further justifies the notion of using a low bit width (8 bits here) for data representation.

6 Conclusions

This paper presents a hardware design of a run time and compiler time reconfigurable RBM based ANN for ‘human posture’ identification in ‘always-on’ BWCs. Design space exploration reveals the need for algorithm-hardware co-optimization and illustrates a minimum energy design point for 30 physical NPCs. At the minimum energy point, we spend less than 5nJ per frame and achieve an accuracy of 85 percent for such a limited training set and shallow network, while still maintaining the real time constraints.

References

[1] L. Miller and J. Toliver, “Implementing a body-worn camera,” Police Executive Research Forum (PERF), and United States of America, 2014.

[2] M. Goodall, “Guidance for the Police Use of Body-Worn Video Devices,” Home Office, 2007, http://library.college.police.uk/docs/homeoffice/guidance-body-worn-devices.pdf

[3] (2015, Mar.). National Law Enforcement and Corrections Technology Center (NLECTC), and United States of America. “Primer on Body-Worn Cameras for Law Enforcement.” [Online]. Available: https://www.justnet.org/pdf/00-Body-Worn-Cameras-508.pdf

[4] Z. M. Sami, H. Saleh, H. Bhaskar, E. Salahat, and M. Ismail, “A low-power 65-nm ASIC implementation of background subtraction,” in *Proc. IEEE 10th Int. Conf. Innovations Inform. Technol.*, 2014, pp. 71–74.

[5] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[6] T. Huixuan. (2015, Feb.). “A comparative evaluation of deep belief nets in semi-supervised learning,” [Online]. Avalable: http://www.cs.toronto.edu/~hxtang/projects/dbn_eval/dbn_eval.pdf

[7] B. Ruihan and T. Shibata, “A hardware friendly algorithm for action recognition using spatio-temporal motion-field patches,” *Neurocomputing*, vol. 100, pp. 98–106, 2013.

[8] A. Fischer and C. Igel, “Training restricted Boltzmann machines: An introduction,” *Pattern Recog.*, vol. 47, no. 1, pp. 25–39, 2014.

[9] H. Geoffrey, “Training products of experts by minimizing contrastive divergence,” *Neural Comput.*, vol. 14, no. 8, pp. 1771–1800, 2002.

[10] G. Mayraz and G. E. Hinton, “Recognizing handwritten digits using hierarchical products of experts,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 2, pp. 189–197, Feb. 2002.

[11] G. E. Hinton, S. Osindero, and Y. W. Teh, “A fast learning algorithm for deep belief nets,” *Neural Comput.*, vol. 18, no. 7, pp. 1527–1554, 2006.

[12] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri, “Actions as space-time shapes,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 12, pp. 2247–2253, Dec. 2007

[13] W. D. James, and A. F. Bobick, “A robust human-silhouette extraction technique for interactive virtual environments,” *Modelling and Motion Capture Techniques for Virtual Environments*. Berlin, Germany: Springer, pp. 12–25, 1998.

[14] K. Kim, T. H. Chalidabhongse, D. Harwood, and L. Davis, “Real-time foreground-background segmentation using codebook model,” *Real-Time Imaging*, vol. 11, no. 3, pp. 172–185, 2005.

[15] I. Zafar, U. Zakir, I. Romanenko, R. M. Jiang, and E. Edirisinghe, “Human silhouette extraction on FPGAs for infrared night vision military surveillance,” in *Proc. 2nd Pacific-Asia Conf. Circuits, Commun. Syst.*, vol. 1, 2010, pp. 63–66.

[16] A. Raychowdhury, C. Tokunaga, W. Beltman, M. Deisher, J. W. Tschanz, and V. De, “A 2.3 nJ/frame voice activity detector-based audio front-end for context-aware system-on-chip applications in 32-nm CMOS,” *IEEE J. Solid-State Circuits*, vol. 48, no. 8, pp. 1963–1969, Aug. 2013.

[17] H. Geoffrey, “A practical guide to training restricted Boltzmann machines,” *Momentum*, vol. 9, no. 1, p. 926, 2010.

Soham Jayesh Desai received the BE degree from the Birla Institute of Technology and Science (BITS) Pilani, in 2013, and the MS degree with thesis from the Georgia Institute of Technlogy in 2015. Since then, he has been with the Security Privay Research Group, Intel Labs, Intel Corporation, Hillsboro, OR. His major areas of research interest are in systems and computer architecture design with emphasis in security, artificial intelligence, and acceleration of algorithms implemented in hardware.

Authorized licensed use limited to: Georgia Institute of Technology. Downloaded on July 04,2021 at 02:48:31 UTC from IEEE Xplore. Restrictions apply.

Mohammed Shoaib (S'08-M'13) received the BTech and MTech degrees in electrical engineering from IIT Madras, and the MA and PhD degrees in electrical engineering from Princeton University in 2007, 2008, 2010, and 2013, respectively. Since 2013, he has been a researcher in the New Experiences and Technologies Group (NExT) at Microsoft Research, Redmond. His research focuses on building energy-efficient sensing and computing systems, which include components of machine learning, signal processing, and computer vision. He has coauthored 2 book chapters, 14 patents, and more than 30 technical papers in this area. He served as a fellow of the McGraw Center for Teaching and Learning, NJ, in 2012. He received the Harold W. Dodds Honorific fellowship and the Gordon Wu Prize for Excellence from Princeton University in 2012, and the Qualcomm Innovation fellowship and the Roberto Padovani Scholarship in 2011. He is a member of the IEEE.

Arijit Raychowdhury (M-'07-SM-'13) received the BE degree in electrical and telecommunication engineering from Jadavpur University, India, and the PhD degree in electrical and computer engineering from Purdue University. He is currently an associate professor in the School of Electrical and Computer Engineering at the Georgia Institute of Technology where he currently holds the ON Semiconductor Junior Research Professorship. He joined Georgia Tech in January, 2013. His industry experience includes five years as a staff scientist in the Circuits Research Lab, Intel Corporation, and a year as an analog circuit designer with Texas Instruments Inc. His research interests include digital and mixed-signal circuit design, design of on-chip sensors, memory, and device-circuit interactions. He holds more than 25 US and international patents and has published more than 100 articles in journals and refereed conferences. He is the winner of the Intel Early Career Faculty Award, 2015; US National Science Foundation (NSF) CRII Award, 2015; Intel Labs Technical Contribution Award, 2011; Dimitris N. Chorafas Award for outstanding doctoral research, 2007; the Best Thesis Award, College of Engineering, Purdue University, 2007; Best Paper Awards at the International Symposium on Low Power Electronic Design (ISLPED) 2012, 2006; IEEE Nanotechnology Conference, 2003; SRC Technical Excellence Award, 2005; Intel Foundation Fellowship 2006, NASA INAC Fellowship 2004, and the Meissner Fellowship 2002. He is a senior member of the IEEE.

▷ **For more information on this or any other computing topic, please visit our Digital Library at** www.computer.org/publications/dlib.

Authorized licensed use limited to: Georgia Institute of Technology. Downloaded on July 04,2021 at 02:48:31 UTC from IEEE Xplore. Restrictions apply.