An Ultra-low power, “Always-On” Camera Front-End for Posture Detection in Body Worn Cameras using Restricted Boltzman Machines

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 recognition accuracy of more than 80% 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.**

Keywords—Body Worn Cameras, Smart Front-end, Restricted Boltzmann Machines, Low Power Recognition, Human Action Recognition *(key words)*

I. 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 for law enforcement. Automatic recognition of human actions and postures is a key enabler for both video surveillance and Body-Worn Cameras.

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

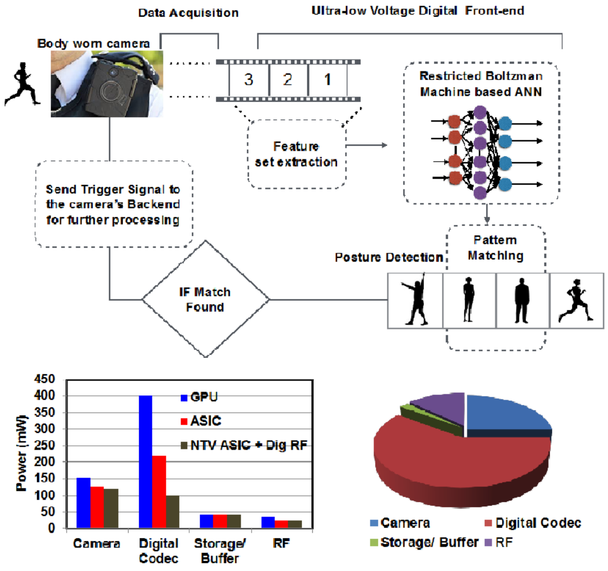

Fig. 1. (a) Usage model for a typical ‘always-on’ body worn camera (b) The different components of power dissipation in a state-of-the-art camera based sensor node with continuous wireless transmission (c) Breakdown of power illustrating a large section of the total dissipation in the digital codec (H.264).

capture data and either store or wirelessly transmit it. Fig. 1b and c 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 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 greater than 80% accuracy 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 at less than 20mW (excluding the power of the signal acquisition unit and the background subtraction unit). 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.

II. RESTRICTED BOLTZMAN 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 such deep networks 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, our accuracy targets can be relaxed from that of deep networks (accuracies > 95-97%). Our target is > 80% accuracies (with minimum number of false rejects) but with tens of mW of power consumption. This is almost three orders of magnitude reduction of power when compared to 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 (RBM) 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, which amounts to storing different sets of weights for each layer and reusing the available computational resources. These weights are pre-trained in software for our network and the usage model requires them to be programmed on the BWC before deployment. Hardware based online training can also be incorporated in the usage model, but since our primary objective is ultra-low power, we have adopted off-line training.

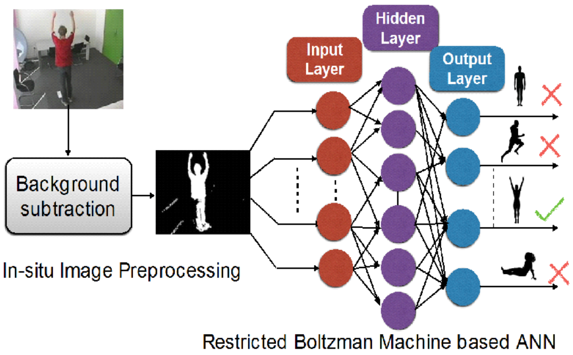
Hidden Layer
Input Layer
Output Layer
Background subtraction
In-situ Image Preprocessing
Restricted Boltzman Machine based ANN

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.

A. 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 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 [5]

$$p\ (v,h)\ \alpha\ e^{-E(v,\ h)} \qquad (1)$$

Here the Energy function is given by

$$E\ (v,h) \ = - \sum_i \sum_j w_{ij}\, h_i\, v_j\ -\ \sum_j b_j\, v_j\ -\ \sum_i c_i\, h_i \qquad (2)$$

The j and i sum over all the nodes in the visible layers and hidden layer respectively. w_{ij} represents real valued weights across the edge between the jth visible node and ith hidden node. b_i and 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 [5] that the conditional probability of any unit being 1 can be written as

$$P\ (\ H_i = 1 \mid v) \ = \ sig(\ \sum_j w_{ij}\, v_j\ +\ c_i\ ) \qquad (3)$$

$$P (V_j=1 \mid h) \;=\; sig(\textstyle\sum_i w_{ij}\, h_i \;+ b_j) \qquad (4)$$

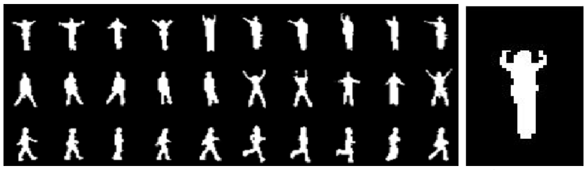
(a) (b)

Fig. 3. (a) Actions in Weizmann Human Actions Silhouette Database (b) 'Both Arms Raised' - Action Silhouette

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

B. 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 [5] for approximating the RBM log-likelihood gradient namely, "Contrastive Divergence" which was originally described in [9]. Obtaining unbiased estimates of the log-likelihood gradient using Markov Chain Monte Carlo methods typically requires many sampling steps. In [9] 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 [9] 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 [5]. As per [5] 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 application of the sigmoid non-linearity to generate the hidden node representation. A similar method is applied for the classification layer

C. Image Database for Posture Dectection:

The experiments are carried out on the Weizmann human silhouette based action database [10] (Fig. 3). The database consists of video sequences (180 x 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. 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 data base is evolving and better training sets are expected in the recent 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 as the situation and usage demands. However, posture identification is a key primitive that can enable 'always-on' BWCs for law enforcement.

III. 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 [11] [12] [13].

A. Hardware Design of the Recognizer

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

$$\textstyle\sum_{i} W_{ip}\, X_{ip} + \theta_{p.} \qquad (6)$$

In Equation (6), W_{ip} represents the synaptic weights, X_i represents the input feature to the neuron, θ_i is the bias for the p^{th} neuron and these are summed over all the input features of the image. Each Neuron in our network models the computation represented by (6). 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

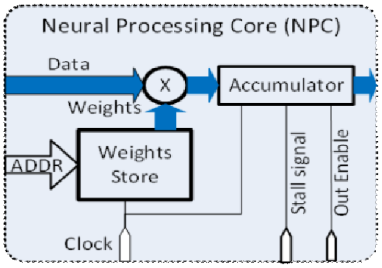

Fig. 4. Block Diagram of the Neuron Processing Core showing the incoming and outgoing control signals, the data path and the address bus

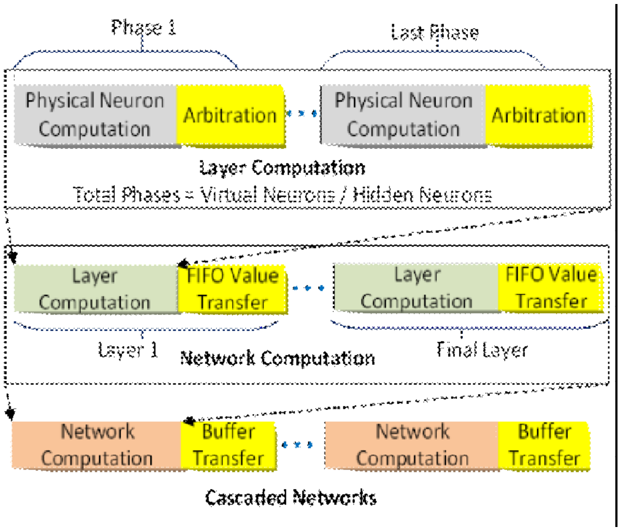

Fig. 5. Depicts the Virtualization of Neuron Computation, Layer Computation and the entire network. The Physical NPCs are reused for a count equal to Total Phases so as to compute for all the Virtual NPCs. The Layer can be reused to provide an increase in depth of the network. Similarly, The Entire Network can then be reused for a different purpose or even for recognizing the same image with higher Accuracy.

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 (6) 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 physical resources. This comes with large area and power (both leakage and dynamic). Virtual NPCs represent the actual number of neurons in a hidden layer for a particular network configuration. The ratio between Virtual Neurons and the Hidden Neurons 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 NPC executing as many times as the number of virtual NPCs or neurons in the layer, resulting in the least amount of resource utilization, power but much 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. For a given computational complexity at a frame rate of 30fps, a lower number of 'virtual NPCs' demonstrate a favorable trade-off between power and the total computational time.

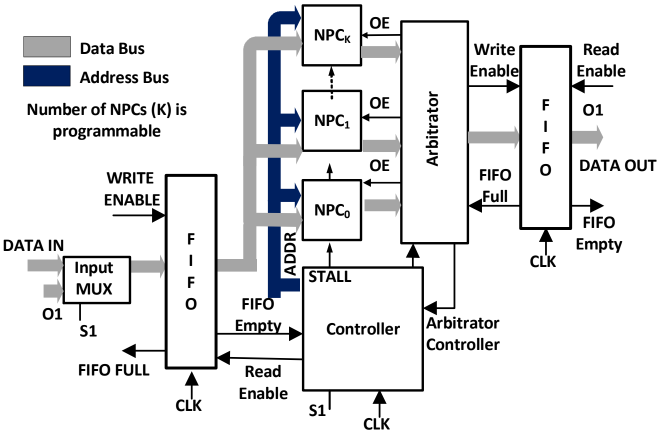

Fig. 6. The Block Design of a Single Layer showing the control and data flow. 01, the output of the layer if input back into a multiplexer. S1, a control signal from the Controller is used for selecting the input.

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 remain 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.

We pipeline the layers using a FIFO. The FIFO full and empty 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 layer by allowing 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

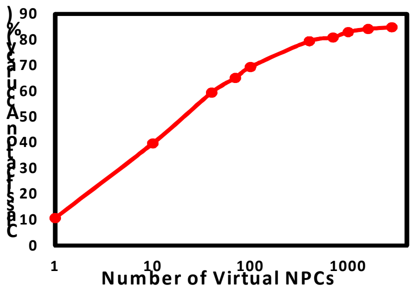

Fig. 7. Showcases the increasing in Recognition Accuracy with Increase in Number of Virtual NPCs in the hidden layer of a Network as the network gains more representation power.

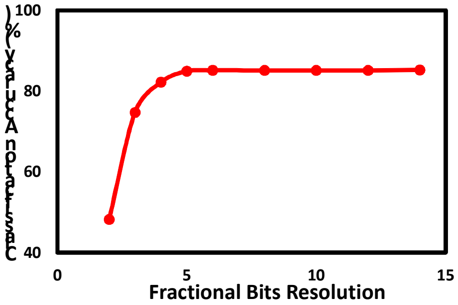

Fig. 8. Classification Accuracy with varying fixed point representation resolutions. The integer bits kept constant at 1. We make sure there is low probability of overflow by having higher bit width for the accumulator

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 Q2.10 for the base case. Fixed point data representation of resolution more than Q2.6 shows no impact on recognition accuracy over a floating point representation and 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 has shown to impact 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 IV. The configurable parameters comprise of the following:

1) Fixed Point Data Resolution : We maintain the total resolution length and the fraction length as parameters throughout the system.

2) Number of Input features

3) Number of Virtual and Physical Hidden NPCs

4) Number of Virtual or Physical Hidden Layers

5) Frequency of the clock

IV. Experimental 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 configuration knobs in our design consist of size of input features, number of virtual and physical hidden neurons, 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.

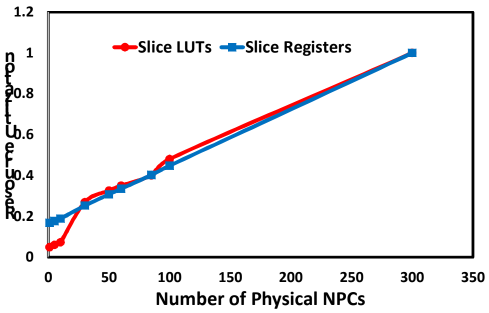

Fig. 9. Describes the Normalized Resource Utilization for increase in parallelization in the network layer. The normalization is with respect to the resource utilization of NPC = 300 (Slice LUT = 97572, Slice Registers = 29582)

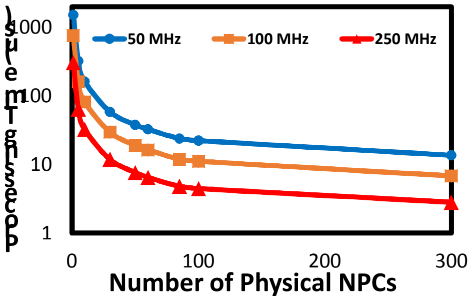

Fig. 10. Describes the Increase in Processing Time as the parallelization is reduced by reusing the Physical NPCs for computation so as to save resources and power

A. Algorithmic Accuracy

Fig. 7 illustrates the trade-off between accuracy and the

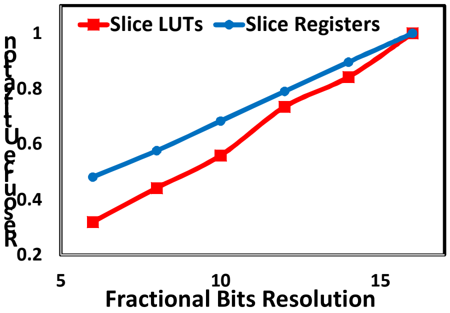

Fig. 11. Normalized Resource Utilization vs fractional bit resolution. Integer bits kept constant. Normalization is carried with respect resource utilization of 16 bits resolution (Slice LUTs = 35787, Slice Registers = 9437)

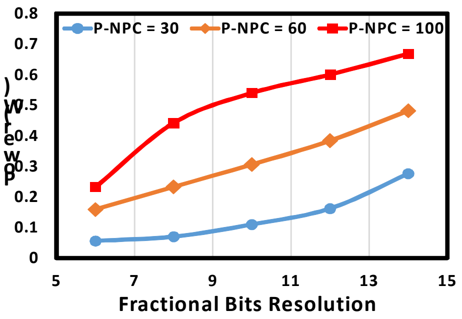

Fig. 12. Power vs Fractional bit resolution for different network Physical (P) NPCs. The integer bits are kept constant.

number of virtual NPCs. We observe an increase in accuracy of the network as the number of Virtual NPCs are increased. We note the saturating nature of the curve and for a target accuracy rate of 80% we choose a baseline design with 300 virtual NPCs. Fig. 8 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.

B. Hardware Measurement Results

The most important design criteria is the choice of the number of physical NPCs. As seen in Fig. 9 the resource utilization of the network can be improved by reducing the number of physical NPCs. This however results in an increase the in the processing time as shown in Fig. 10. It should, however be noted, that at 30 fps the amount of time available for processing the data is sufficient with a small number of physical NPCs. The choice of the data bit width also has significant impact on the resource utilization of the network and has been shown in Fig. 11, which further justifies the notion of using a low bit width (8 bits here) for data representation.

V. Design Space Exploration

To minimize the overall network power and utilization at acceptable performance, we jointly optimize algorithms and hardware. It has already been shown that for acceptable performance, we choose 8 bits for data representation. The number of virtual NPCs is chosen as 300. 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. 12). We observe that increasing the number of NPCs increase the total power dissipation but results in faster compute (Fig. 10). 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. 13 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 NPCs the total data movement and time to process increases rapidly (Fig. 10). 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 note less than 5nJ of energy/frame for processing. This illustrates three orders of magnitude improvement in total power (<20mW) compared to a camera based wireless sensor node.

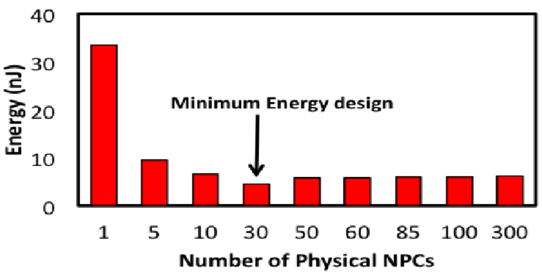

Fig. 13. Total energy/frame as a function of the number of physical NPCs. The 'Minimum Energy' design is obtained for 30 physical NPCs running at a clock frequency of 50 MHz

VI. Conclusions

This paper presents an RBM based ANN for 'human posture' identification in 'always-on' Body-Worn-Cameras. Design space exploration reveals the need for algorithm-hardware co-optimization and illustrates a minimum energy design point for thirty physical NPCs. At the minimum energy point, we spend less than 5nJ per frame and achieve greater than 80% accuracy in posture detection.

References

[1] Police Executive Research Forum (PERF), "Implementing a Body-Worn Camera"

[2] Martin Goodall, Home Office, "Guidance for the Police Use of Body-Worn Video Devices"

[3] Jonathan Hayes, Dr. Lars Ericson, ManTech and NLECTC, A primer on Body-worn cameras for law enforcement.

[4] Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks"

[5] Asja Fischer, Christian Igel, "Training Restricted Boltzmann Machines: An Introduction"

[6] Daniel Le Ly and Paul Chow, "High-Performance Reconfigurable Hardware Architecture for Restricted Boltzmann machines"

[7] Georffrey E. Hinton, Simon Osindero, Yee-Whye The, " A fast learning Algorithm for deep belief nets"

[8] Guy Mayraz, Geoffrey Hinton " Recognizing Handwritten Digits using Hierarchical Products of Experts".

[9] G. E. Hinton, "Training products of experts by minimizing contrastive divergence", *Neuron Computation*, 14: 1771-1800, 2002

[10] L. Gorelick, M. Blank, E. Shectman, M. Irani and R. Basri, "Actions as Space-Time Shapes", IEEE PAMI, vol. 29, no. 12, pp. 2247-2253, 2007

[11] James W. Davis, Aaron F. Bohick, " A Robust Human-Silhoutte Extraction Technique for Interactive Virtual Environments"

[12] Kyungnam Kim, Thanarat H. Chalidabhonse, David Harwood, Larry Davis, "Real-time foreground-background segmentation using codebook model"

[13] Iffat Zafar, Usman Zakir, Ilya Romanenko, Richard M. Jiang, Eran Edirisinghe "Human Silhoutte Extraction on FPGAs for Infrared Night Vision Military Surveillanc""