Title

Deep learning for medical image segmentation { using theIBM TrueNorth Neurosynaptic System

Authors

Moran, Steven
Gaonkar, Bilwaj
Whitehead, William
et al.

Publication Date

2018-02-15

Peer reviewed

Deep learning for medical image segmentation – using the IBM TrueNorth Neurosynaptic System

Steven Moran[a], Bilwaj Gaonkar[b], William Whitehead[a], Aidan Wolk[a], Luke Macyszyn[b], and Subramanian S. Iyer[a]

[a]Center for Heterogeneous Integration and Performance Scaling, UCLA, 420 Westwood Plaza, Engineering IV, Los Angeles, CA 90095 USA
[b]Department of Neurosurgery, UCLA, 300 Stein Plaza Driveway, Los Angeles, CA 90095 USA

ABSTRACT

Deep convolutional neural networks have found success in semantic image segmentation tasks in computer vision and medical imaging. These algorithms are executed on conventional von Neumann processor architectures or GPUs. This is suboptimal. Neuromorphic processors that replicate the structure of the brain are better-suited to train and execute deep learning models for image segmentation by relying on massively-parallel processing. However, given that they closely emulate the human brain, on-chip hardware and digital memory limitations also constrain them. Adapting deep learning models to execute image segmentation tasks on such chips, requires specialized training and validation.

In this work, we demonstrate for the first-time, spinal image segmentation performed using a deep learning network implemented on neuromorphic hardware of the IBM TrueNorth Neurosynaptic System and validate the performance of our network by comparing it to human-generated segmentations of spinal vertebrae and disks. To achieve this on neuromorphic hardware, the training model constrains the coefficients of individual neurons to {-1,0,1} using the Energy Efficient Deep Neuromorphic (EEDN)[1] networks training algorithm. Given the ∼1 million neurons and 256 million synapses, the scale and size of the neural network implemented by the IBM TrueNorth allows us to execute the requisite mapping between segmented images and non-uniform intensity MR images >20 times faster than on a GPU-accelerated network and using <0.1 W. This speed and efficiency implies that a trained neuromorphic chip can be deployed in intra-operative environments where real-time medical image segmentation is necessary.

Keywords: Neuromorphic computing, deep learning, MR imaging, semantic image segmentation, IBM TrueNorth Neurosynaptic System

1. INTRODUCTION

The purpose of this project is to demonstrate deep learning based medical image segmentation on a state-of the-art neuromorphic hardware platform. We show that our approach can automatically delineate spinal anatomy (vertebrae and disks) on T2-weighted spine MR images—collected from different MR scanners with non-uniform intensities—in a manner comparable to human-generated delineations. This is possible regardless of neuron and synapse behavioral constraints that are employed to enable fast, low power runtime execution on a neuromorphic system. Our network architecture is executed on the IBM TrueNorth Neurosynaptic System which runs at significantly lower power as compared to traditional (CPU or GPU systems used for deep learning). It also yields faster overall computation, yielding a significant increase in efficiency (FPS/W) as compared to traditional processing hardware.[2–4] The proposed technique opens the door to performing image-based diagnoses on neuromorphic hardware. Given the high speed, low power deep learning which is possible using neuromorphic hardware and the shift in industry towards the use of cloud-based hardware for deep learning, our network architecture provides a template for developing faster and more efficient image-based diagnostic tools for future use. In addition to improving throughput in large data centers specializing in machine learning, such tools may also be deployable in embedded/low power settings in the future.

E-mail: steven.moran@engineering.ucla.edu

2. METHODS

2.1 The IBM TrueNorth Neurosynaptic System

The chip is designed to contain 1 million neurons and 256 million synapses while consuming only ∼70 mW power at run-time. It consists of 4,096 cores, each with 256 input axons, a 256x256 synaptic crossbar, and 256 output neurons.[1,5] Convolutional Neural Networks (CNNs) are implemented on the IBM TrueNorth using the EEDN algorithm to train a set of trinary coefficients (synaptic weights): {-1, 0, 1}.[1]

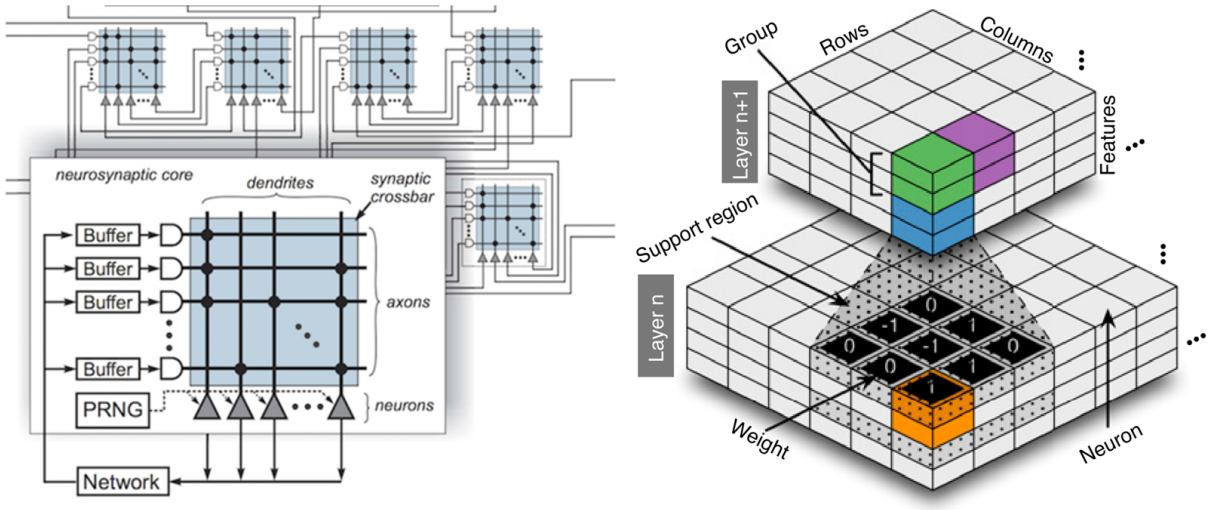

Figure 1. (a) IBM TrueNorth architecture, and (b) two layers of a convolutional neural network (CNN) employed on TrueNorth. *Reproduced with permission.*[1,6]

The configuration of a single core on the IBM TrueNorth Neurosynaptic System is shown in Fig. 1a—axons are defined as the inputs of the core, neurons are the outputs, and the 2-D connectivity map (crossbar) stores the synaptic weight information. As previously mentioned, each core has 256 axons (inputs) and 256 neurons (outputs). Each neuron output is calculated as a weighted-sum of the 256 inputs. Each input can influence each output neuron differently as determined by the strength of the synapses (connections) between the *axons* of the pre-synaptic neurons (i.e. inputs) and the corresponding dendrites of the post-synaptic *neurons* (i.e. outputs). The strength of this synapse is called the synaptic weight (also known as synaptic coefficient). The synaptic weights correlate the input neurons to the output neurons, whose values are determined from the training of the network. The IBM TrueNorth supports integer synaptic weights ranging [-255, 255]. These values are stored in a 4-entry lookup table (LUT) per neuron, supporting 4 different axon types—and hence 4 different synaptic weight values—per neuron. To reiterate, each neuron has a set of 4 possible synaptic weight values that can be used to represent the synapse between each axon (input) and that particular output neuron.

A visualization of the interaction between two layers of a convolutional neural network is provided in Fig. 1b. The output of layer n is fed into the input of layer n+1.[1] Each core on the chip can be used to implement one filter in the network. It also introduces a new set of constraints introduced by the EEDN training algorithm: synaptic weights are limited to a set of trinary coefficients, {-1,0,1}, as shown in Fig. 2. Thus, each neuron's 4-entry lookup table is identically programmed to contain {-1,1}—where 0 is represented separately by a binary synaptic crossbar variable ($S_{ij} = 0$) denoting no connection between the axon and output neuron and indicated by the absence of a black dot in the synaptic crossbar in Fig. 1a. This synaptic crossbar variable is set to 1 for all non-zero synaptic weights—contrariwise represented by a black dot in the synaptic crossbar in Fig. 1a. Neuromorphic designs must trade-off between programmability and density of neurons. Because of this, IBM TrueNorth implements fewer parameters per neuron to increase the total size of the network and its computational efficiency.

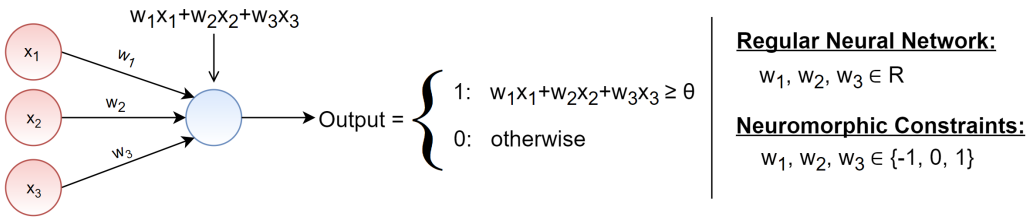

Figure 2. TrueNorth–compatible neural network constraints.

2.2 Data and method

This work utilizes a small subset of the UCLA Radiology Pictures and Communications (PACS) database. We use 200 sagittal T2-spine MR images selected from 42 patients, collected from different MR scanners with non-uniform intensities. Each image is resampled to 256x256 px and manually annotated with segmentations of intervertebral disks and vertebral bodies for training the network models using ITK-SNAP.[7] Testing is done on a separate dataset of 30 sagittal T2-spine MR images selected across 20 patients which are also resampled to 256x256 px. We use a two-step approach to segment spinal anatomy in the images. The first step uses a CNN designed to classify 31x31 px frames as spine/not spine. This CNN produces an approximate localization of the spine anatomy. This step is fast since a 256x256 px image can be divided into a relatively small number of frames (to the order of 10^2). The second step uses another CNN which classifies individual pixels—previously identified as belonging to the spine by the first stage—as vertebral body, disk, or background. We describe each stage in detail next.

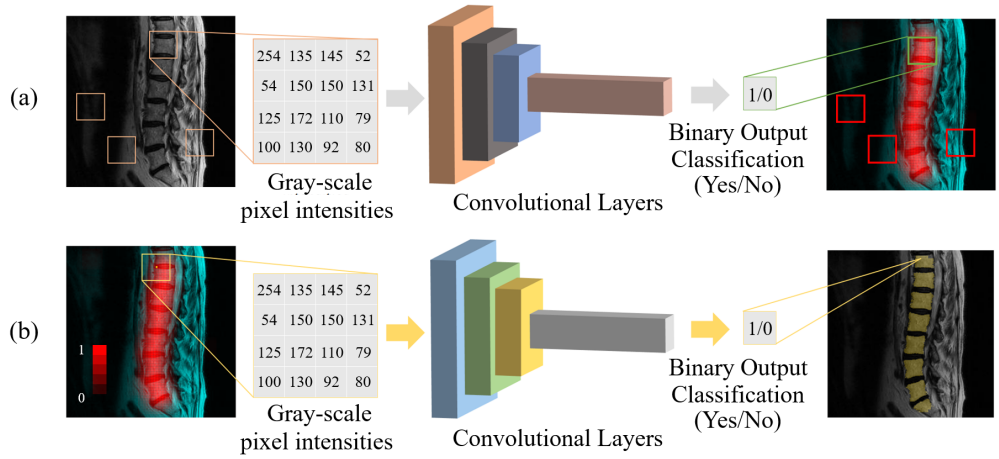

Figure 3. Two-stage network. (a) Detection (localization) network. Red pixels represent areas identified by the detection network as belonging to the spine. Brighter red pixels are more likely to be part of the spinal region than lighter red pixels. Pixel intensity is derived by classifying multiple, overlapping frames and creating a heat map which corresponds to the confidence that the network believes that each particular pixel belongs to the spinal region. b) Pixel-wise classification network.

2.3 Detection network

Rather than processing ∼65,000 frames per MRI scan (256x256 px), we first reduce the problem size by filtering out pixels that are clearly not part of the spine using a 14-layer CNN specifically trained for this purpose. We select overlapping 31x31 px frames with a stride of 4 from the 256x256 px MR image, resulting in ∼4000 frames. These frames are then inputted into a frame-wise CNN that classifies each frame as belonging or not belonging to the spinal region—shown in Fig. 3a.

By classifying overlapping frames, a heat map can be derived, which allows us to quantify our confidence that a particular set of pixels belongs to the spinal region. This is done by observing whether or not several overlapping frames (with stride of 4) containing those pixels were positively identified as belonging to the spine. Red pixels in Fig. 3a represent areas that were classified by the detection network as belonging to the spine. Red intensity corresponds to the heat (density) map where brighter red represents areas of high confidence and lighter red represents areas near the edge of the spinal region and hence lower network confidence. This approach allows us to filter out lower confidence regions and reduce the overall size of the input that is fed into the secondary, pixel-wise classification network for performing segmentation.

2.4 Segmentation network

Pixel-wise classification[8] using deep learning is performed on all the selected pixels from the detection network. For every pixel in the spinal region, we select a 31x31 px frame surrounding it and input this frame into another 14-layer CNN. This frame provides context regarding nearby features for classifying the central pixel of the frame and stays under the maximum frame size (32x32 px) that can be loaded into the chip at once. This CNN performs simultaneous multi-class segmentation and outputs trinary labels—vertebrae (2), disk (1), or neither (0). An example of single class (vertebra-only) segmentation is demonstrated in Fig. 3b.

3. RESULTS

3.1 Comparison with standard, unconstrained neural networks

In our experiments, convolutional neural networks were trained using the EEDN[1] algorithm and loaded onto the IBM TrueNorth. The EEDN platform constrains the allowable synaptic weights to a set of trinary coefficients $\{-1, 0, 1\}$. Despite this constraint accuracy of the constrained (TrueNorth) model that deployed on-chip converges to the accuracy of the accuracy of an unconstrained (software) model after ∼500 epochs for the detection network, as shown in Fig. 4. The unconstrained model used in our experiments uses the workflow described in Figure 3. The main difference is that standard stochastic gradient descent is used to train a network where neuron weights can take non-integer values. Our experiment shows that constrained neural networks required for image segmentation using neuromorphic hardware can achieve accuracies comparable to standard neural networks, given the required amount of training. Both constrained and unconstrained models were trained using the MatConvNet[9] framework on a NVIDIA GTX 1080 graphics processing unit (GPU).

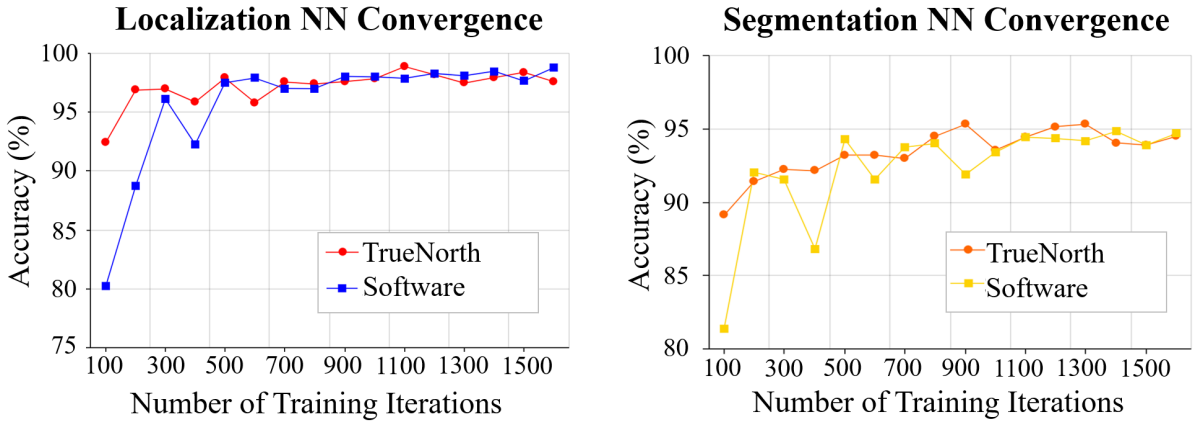

Figure 4. Convolutional neural network (CNN) training convergence.

3.2 Qualitative comparison of medical image segmentations performed using proposed hardware to human experts

A qualitative summary of results obtained using our two-stage approach is provided in Fig. 5. Manual segmentations are provided for qualitative comparison. Localization images highlight all selected pixels from the detection step. The segmentation network then performs pixel-wise classification on the selected pixels and produces the automated vertebral body and disk segmentations, shown in columns d-e.

3.3 Quantitative comparisons

The detection network accuracy was calculated using 800 test frames split equally between the two classification categories—spine and not spine. The network's classification accuracy for frames belonging and not belonging to the spinal region was 98.3% and 98.7%, respectively. Dice scores[10,11]—also known as "similarity coefficients"—are one way of evaluating overlap between segmentations. A dice score can be defined as:

$$Dice\,Score = \frac{2A(S_m \cap S_a)}{A(S_m) + A(S_a)}, \tag{1}$$

S_m and S_a represent the manual and automatic segmentations, respectively. $S_m \cap S_a$ refers to the overlap between the two segmentations and $A(\cdot)$ defines the area of the relevant object (2D object from a sagittal slice of an MR image). A dice score of 1 represents a perfect correlation between the manual and automated segmentations. The dice score statistics comparing our segmentations to manual segmentations for the pixel-wise segmentation network, calculated on the 30 test spinal MRI images, is shown in Fig 6—dice scores statistics are further summarized in Table 1.

Table 1. Summary of dice score statistics.

	Mean	**Std.**	**Max.**	**Min.**
Vertebra	0.828	0.077	0.924	0.618
Disk	0.786	0.062	0.870	0.612

4. DISCUSSION

4.1 Automated segmentation results

Both the detection and segmentation networks were trained and tested on sets of sagittal T2-weighted spine MR images collected from different MR scanners with non-uniform intensities. Train and test sets were selected randomly across all sagittal planes to prevent over-fitting. Thus, the trained system can be used to perform segmentation across any frame from an MR image.

Near perfect automated segmentations are shown for test images 1, 2, and 4 in Fig. 5. Test images 3 and 5, however, show reduced segmentation results due to misclassification that occurred primarily in the detection step. Several areas in test images 3 and 5 were classified as belonging to the spine (column c of Fig. 5). These areas were then inputted into the following network for segmentation, leading to some erroneous segmentation results. Misclassification at the detection stage is the primary cause of reduced Dice scores. Several post-processing steps including filtering or a secondary false-positive detection can be performed to further improve segmentation results.

4.2 Conventional vs. neuromorphic hardware

Neural networks have had incredible success in semantic image segmentation tasks. They are, however, incredibly inefficient on conventional hardware. In almost all cases, conventional processors rely on von Neumann architectures. The von Neumann bottleneck is an inherent limitation of conventional hardware where computation is limited by the processor's ability to fetch and transfer information from memory.[12] Graphics processing

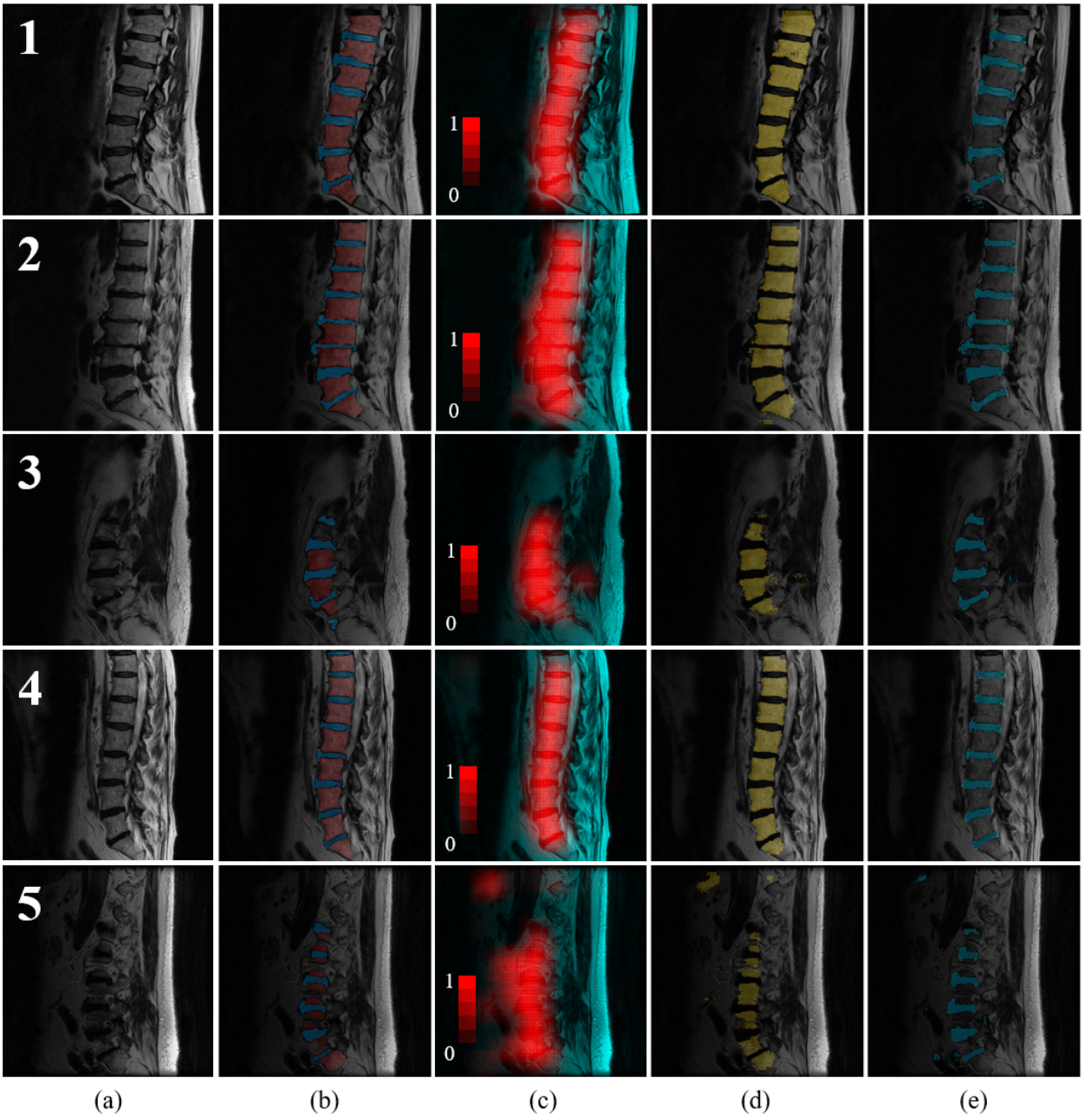

Figure 5. Automated segmentation results. (a) Original MR image. (b) Manual segmentations performed by human experts. (c) Detection (localization) results. Red pixels represent areas identified by the detection network as belonging to the spine. Red intensity corresponds to the network's confidence that each particular pixel belongs to the spinal region—brighter red represents higher confidence. (d) Automated vertebrae segmentations. (e) Automated disk segmentations.

units (GPU's) provide significant speed improvements for neural network applications by incorporating multi-threading, but they are unusable in deployable or embeddable environments due to large power budgets (>200 W). Although not technically considered von Neumann, GPU's are still thread-limited and suboptimal for massively parallel structures such as deep neural networks. The overall complexity of multi-threaded software also limits them. Neuromorphic hardware introduces a new paradigm of computing: application-specific processing.

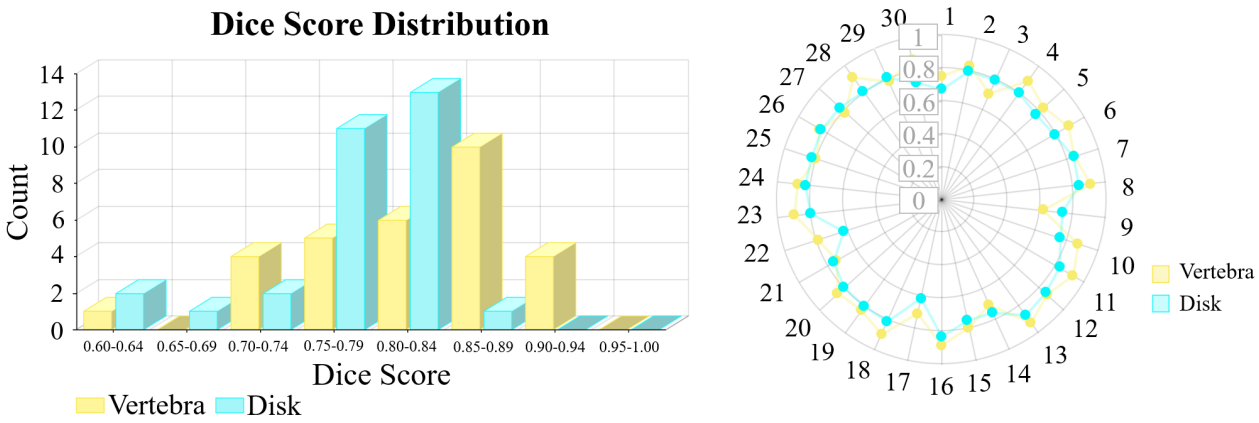

Figure 6. Dice score distribution *(left)* and disk/vertebra dice score comparison for 30 test images *(right)*.

The IBM TrueNorth exploits the massively-parallel structure of neural networks. It utilizes significantly lower power (<0.1 W) and provides high classification throughput (∼1000 FPS). This correlates to a significant increase in efficiency (FPS/W) when compared against traditional processing hardware.[2–4]

Our approach is novel as we constrain the neurosynaptic weights in both network stages (CNN's) to a set of trinary coefficients and still achieve comparable results to human-generated segmentations. This enables us to exploit the advantages of neuromorphic hardware, proving large image pixel-wise segmentation feasible and deployable, even in low power applications. Given the high speed, low power deep learning which is possible using neuromorphic hardware and the increasing trend of using specialized hardware for deep learning, we believe that our network architecture provides a template for developing image-based diagnostic tools. Such tools may be deployed in embedded/low power settings as well as to improve throughput in large data centers at the opposite side of the power spectrum.

5. CONCLUSION

We demonstrated a deep learning approach implemented on neuromorphic hardware for delineating spinal anatomy on MR images in a manner comparable to human experts. Using only trinary weights, we can achieve accuracies comparable to unconstrained NN models. This has several consequences. Without any reduction in the quality of results, we can exploit the benefits that neuromorphic hardware offers. We can envision applications where more intensive deep learning tasks can be performed in low power and embedded settings. We have also demonstrated that large scale pixel-wise segmentation using neuromorphic hardware can be executed orders of magnitude faster than on conventional GPU hardware.

5.1 On-going and future work

This work utilized a small subset of data from 62 patients for establishing the train and test datasets. At UCLA, we have access to over 500,000 imaging studies including MR, CT, and X-ray imaging. In particular, the University of California Research eXchange (UC-ReX) and UCLA Radiology Pictures and Communications (PACS) databases provide us with direct access to over 40,000 patients with lumbar MR scans accompanied by detailed diagnoses including longterm follow-ups pre- and post-surgery as well as outcome measures and medications. We plan on training current and future, more complex deep learning models on increasingly larger sets of data. We are also developing image-based biomarkers for comparing patients with similar pathologies, which will help provide objective context when performing data-driven diagnoses and developing more successful treatment plans.

ACKNOWLEDGMENTS

This work was supported in part by the UCLA CHIPS consortium, the Department of the Defense Defense Threat Reduction Agency (HDTRA1-17-1-0035), and the University of California Multicampus Research Program (MRP-17-454999). We would also like to thank the IBM Corporation for the TrueNorth hardware and support.

REFERENCES

[1] Esser, S. K., et. al., "Convolutional networks for fast, energy-efficient neuromorphic computing," *Proceedings of the National Academy of Sciences* **113**(41), 11441–11446 (2016).

[2] NVIDIA, "Whitepaper: GPU-Based Deep Learning Inference: A Performance and Power Analysis," 1–12 (2015).

[3] Jouppi, N. P., et. al., "In-Datacenter Performance Analysis of a Tensor Processing Unit$^{\mathrm{TM}}$," *44th International Symposium on Computer Architecture (ISCA)*, 1–17 (2017).

[4] Sawada, J., et. al., "TrueNorth Ecosystem for Brain-Inspired Computing: Scalable Systems, Software, and Applications," *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, 130–141 (2016).

[5] Merolla, P., et. al., "A million spiking-neuron integrated circuit with a scalable communication network and interface," *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis* **345**(6197), 668–673 (2014).

[6] Amir, A., et. al., "Cognitive Computing Programming Paradigm: A Corelet Language for Composing Networks of Neurosynaptic Cores," *Proceedings of the International Joint Conference on Neural Networks* (2013).

[7] Yushkevich, P. A., et. al., "User-guided 3d active contour segmentation of anatomical structures: Significantly improved efficiency and reliability," *NeuroImage* **31**(3), 1116–1128 (2006).

[8] Tschopp, F., Martel, J. N. P., Turaga, S. C., Cook, M., and Funke, J., "Efficient Convolutional Neural Networks for Pixelwise Classification on Heterogeneous Hardware Systems," *Proceedings of the International Symposium on Biomedical Imaging*, 1225–1228 (2016).

[9] Vedaldi, A. and Lenc, K., "Matconvnet - convolutional neural networks for matlab," *arXiv*, 1–15 (2014).

[10] Gaonkar, B., et. al., "Automated Tumor Volumetry Using computer-Aided Image Segmentation," *Academic Radiology* **22**(5), 653–661 (2015).

[11] Gaonkar, B., et. al., "Multi-Parameter Ensemble Learning for Automated Vertebral Body Segmentation in Heterogeneously Acquired Clinical MR Images," *IEEE Journal of Translational Engineering in Health and Medicine* **5**, 1–12 (2017).

[12] Thimbleby, H., "Modes, WYSIWYG and the von Neumann bottleneck," *IEE Colloquium on Formal Methods and Human-Computer Interaction: II*, 1–15 (1988).