CortexSuite: A Synthetic Brain Benchmark Suite

Shelby Thomas, Chetan Gohkale, Enrico Tanuwidjaja, Tony Chong, David Lau,
Saturnino Garcia*, and Michael Bedford Taylor
*University of San Diego
University of California, San Diego
darksilicon.org/cortexsuite

***Abstract*—These days, many traditional end-user applications are said to "run fast enough" on existing machines, so the search continues for novel applications that can leverage the new capabilities of our evolving hardware. Foremost of these potential applications are those that are clustered around information processing capabilities that humans have today but are lacking in computers. The fact that brains can perform these computations serves as an existence proof that these applications are realizable. At the same time, we often discover that the human nervous system, with its 80 billion neurons, on some metrics, is more powerful and energy-efficient than today's machines. Both of these aspects make this class of applications a desirable target for an architectural benchmark suite, because there is evidence that these applications are both useful and computationally challenging.**

This paper details CortexSuite, a Synthetic Brain Benchmark Suite, which seeks to capture this workload. We classify and identify benchmarks within CortexSuite by analogy to the human neural processing function. We use the major lobes of the cerebral cortex as a model for the organization and classification of data processing algorithms. To be clear, our goal is not to emulate the brain at the level of the neuron, but rather to collect together synthetic, man-made algorithms that have similar function and have met with success in the real world. We consulted six world-class machine learning and computer vision researchers, who collectively hold 83,091 citations across their distinct subareas, asking them to identify newly emerging computationally-intensive algorithms or applications that are going to have a large impact over the next ten years. This is coupled with datasets that reflect the philosophy of practical use algorithms and are coded in "clean C" so as to make them accessible, analyzable, and usable for parallel and approximate compiler and architecture researchers alike.

I. Introduction

In this paper, we present CortexSuite, a Synthetic Brain Benchmark Suite which captures an emerging workload that is clustered around providing information processing capabilities that human brains have today but that computers are only just now beginning to become good at. We classify and identify benchmarks within CortexSuite by analogy to the human neural processing function, as shown in Fig 1. We use the major lobes of the cerebral cortex as a model for the organization and classification of data processing algorithms. *Our goal is not to emulate the brain at the level of the neuron, but rather to collect together man-made (i.e. synthetic) algorithms that have similar capabilities and have met with success in the real world.*

Benchmark Organization. In the organization of this benchmark suite we draw parallels between three major data processing classes and the four majors lobes of the sensory cortex: the temporal, occipital, parietal, and frontal. With this abstraction, we classify the occipital and temporal lobe, which work together to provide the visual processing centers of the brain, similar to a computer vision core. The parietal lobe, which is responsible for language comprehension, is parallel to a natural language core. Finally, we link the frontal lobe with learning and preprocessing benchmarks. To account for the varied nature of data processing tasks, our proposed taxonomy helps to maintain consistency when extending this benchmark suite. For example, vision benchmark suites such as MEVBench [2] can be integrated into the occipital node of the suite.

The benchmark suite contains eight unique applications from the natural language processing, computer vision, and machine learning domains. It is intended to extend SD-VBS, which provides a core of relatively recent vision algorithm. The benchmarks are shown in Table I.

Clean C coding. The benchmarks were coded to reduce the unnecessary use of pointer calculations, I/O and machine-specific optimizations that inhibit parallelization and/or their use in prototype compilers and/or architectural simulators. Oftentimes, existing code bases have been tuned for a particular architecture, which obfuscates the underlying algorithm, making it difficult to retarget to a new execution model, and also preventing code transformations that can unlock parallelism.

Representative Datasets. Real-world datasets were used to represent actual commercial and academic scenarios in an effort to mirror real-world use-cases. Moreover, we provided a spectrum of input sizes with varying run times and in some cases, different properties, for each benchmark, when it affects execution properties significantly.

II. Benchmark Descriptions

The goal of the benchmark suite is to provide a set of algorithms that have the flexibility to cover a spectrum of domains with representative datasets in current academic and commercial uses. The applications have been coded to eliminate the use of unnecessary complex pointer operations and machine-dependent optimization to expedite the use of the suite for parallel/approximate compiler and architecture researchers. We omit the descriptions for the benchmarks that derive from SD-VBS [1] for the purposes of space.

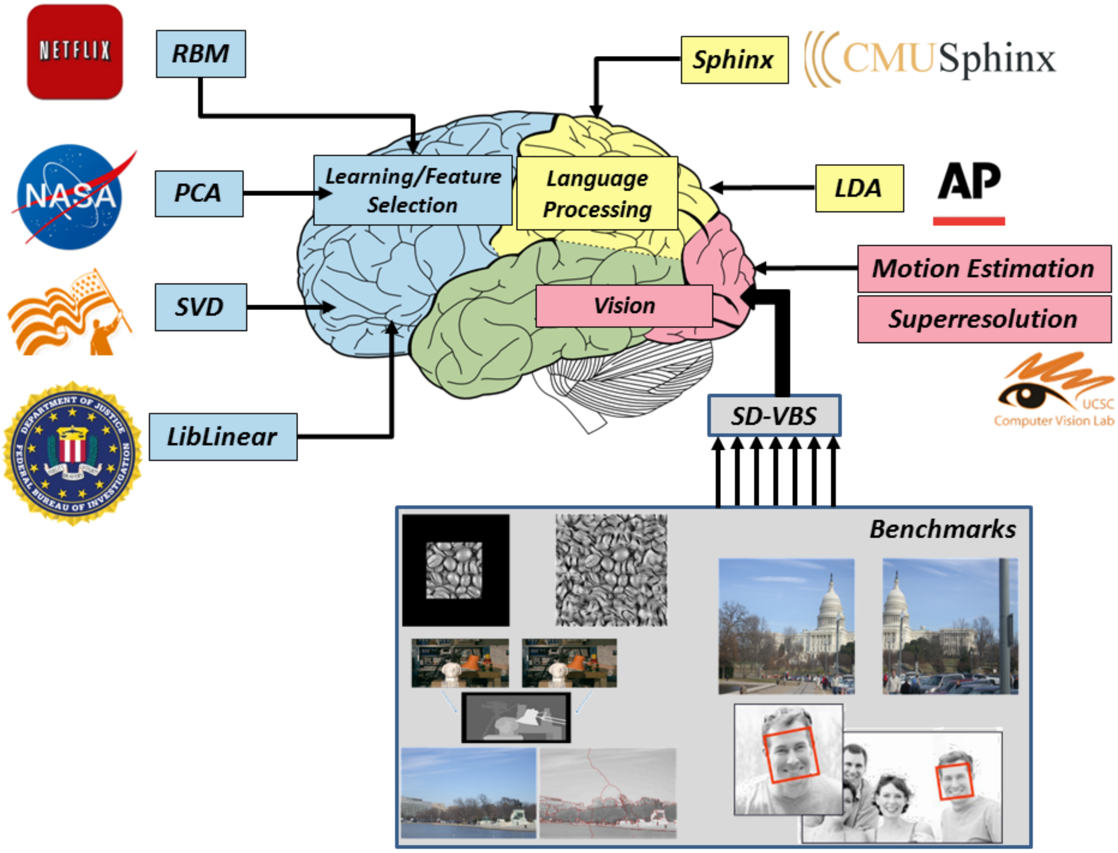

Fig. 1. ***The Synthetic Brain Benchmark Suite.*** CortexSuite captures an emerging workload that is clustered around providing information processing capabilities that human brains have today but that computers are only just now beginning to become good at. We classify and identify benchmarks within CortexSuite by analogy to the human neural processing function. We use the major lobes of the cerebral cortex as a model for the organization and classification of data processing algorithms. Our goal is not to emulate the brain at that level of the neuron, but rather to collect together man-made algorithms that have similar function and have met with success in real world use. To collect these benchmarks, we consulted six world-class vision and machine learning researchers, asking them to identify newly emerging computationally-intensive algorithms or applications that are going to have a large impact over the next ten years. This benchmark suite extends the SD-VBS vision benchmark suite [1] with eight more brain-centered applications–and provide an analysis of basic properties including the quantity of innate parallelism in each benchmark. In this diagram, we have mapped applications to the associated parts of the brain's cerebral cortex. Vision is processed primary by the occipital and temporal lobes, language processing by the parietal lobe, and learning/feature selection by the frontal lobe.

CortexSuite contains eight different applications in addition to those in SD-VBS with three dataset sizes: small, medium, and large, corresponding to exponentially increasing execution times. We include multiple algorithms for applications in natural language processing, computer vision, and machine learning. In addition, a spectrum of unique and useful computer vision applications have also been provided through SD-VBS, summarized in Table I.

Restricted Boltzmann Machines (RBM) has seen exponentially growing use over the last couple of years in the context of deep learning networks. It is a stochastic neural network algorithm with applications in collaborative filtering [3], feature learning and topic modeling. The RBM algorithm was used in the Netflix prize's winning solution in 2009 [4], where Netflix provided their database of films, with user rankings, and offered $1 million for the best algorithm that would predict ratings of films that the user had not yet rate. RBM featured prominently in many of the top solutions that emerged over time. CortexSuite utilizes RBM to implement movie suggestions (collaborative filtering) on variants of the Netflix database and provides the benchmark for the training process of RBM, the most computationally intensive aspect. To scale the run-time, we provide several scaled versions of the database; discarding minimally-connected films that have little impact on training.

Sphinx Speech Recognition is used for the translation of spoken words to text [5] by taking the raw waveform, splitting it on utterances by silences, and attempting to recognize the word in each utterance. We group all possible combinations of words and the best matching combination is chosen.

Super-resolution Reconstruction (SRR) is based on the idea that slight sub-pixel variations in the information encoded in a series of low resolution (LR) images can be used to recover one high resolution (HR) image [6], [7]. Computational resolution enhancement has many applications in the fields of photography, healthcare, security, astronomy, and military.

TABLE I

Benchmarks in the Synthetic Brain Benchmark Suite. CortexSuite contains nineteen benchmarks that seek to encompass important new applications; it includes eight new benchmarks (TOP) and a number of vision-based benchmarks from SD-VBS (BOTTOM).

Benchmark	**Category**	**Dataset**	**Application Domain**
Restricted Boltzmann Machines	Deep Learning	Netflix	Machine Learning
LibLinear	Classification/Regression	FBI Crime Statistics	Machine Learning
Principal Component Analysis	Feature Selection	NASA	Machine Learning
Singular Value Decomposition	Feature Selection	KOS Press	Machine Learning
Sphinx Speech Recognition	Speech Recognition	CMU	Natural Language Processing
Latent Dirichlet Allocation	Topic Modeling	Associated Press	Natural Language Processing
Super Resolution Reconstruction	Image Reconstruction	MDSP Research	Computer Vision
Motion Estimation	Motion, Tracking	MDSP Research	Computer Vision
Disparity Map	Motion, Tracking and Stereo Vision		Computer Vision
Feature Tracking	Motion, Tracking and Stereo Vision		Computer Vision
Image Segmentation	Image Analysis		Computer Vision
SIFT	Image Analysis		Computer Vision
Robot Localization	Image Understanding		Computer Vision
SVM	Image Understanding		Computer Vision
Face Detection	Image Understanding		Computer Vision
Image Sitch	Image Processing and Formation		Computer Vision
Texture Synthesis	Image Processing and Formation		Computer Vision

Latent Dirichlet Allocation (LDA) is a topic modeling algorithm that is commonly found in natural language processing to discover topics from unordered documents. The underlying algorithm uses the assumption that each document was generated using a Dirichlet Distribution, which serves as a prior distribution to the Multinomial Distribution[8]. The goal of the algorithm is to find values for the multinomial parameter vectors and topics for each document.

Singular Value Decomposition (SVD) is a rank reduction algorithm used in many artificial intelligence, signal processing, and computer vision applications. We utilize SVD for latent semantic analysis as a tool for natural language processing.

Principle Component Analysis (PCA) is one of the most versatile and widely used statistical techniques for feature extraction in multivariate datasets. PCA is found in applications ranging from computer vision to machine learning and its tendency to use several matrix operations lends itself to parallelization.

Motion Estimation is the process of finding motion vectors that describe the transformation of one 2D image to another. Motion Estimation is an essential element in image and video processing. It is a key part of the video encoding and applications such as frame rate upsampling and superresolution. Performance of motion estimation can directly affect the performance of these applications.

Liblinear is a versatile library for large-scale linear classification and regression with applications in computer vision, natural language processing, neuroimaging, and bioinformatics. The library has been used in a variety of applications from real-time object recognition to predicting protein solubility and supports linear SVM, linear support vector regression (SVR) and logistic regression. Liblinear is especially powerful for large scale data, i.e. with a large number of instances and features, as it is much faster (100× faster than libSVM) than other state-of-art linear or nonlinear SVM libraries [9] while keeping high accuracy.

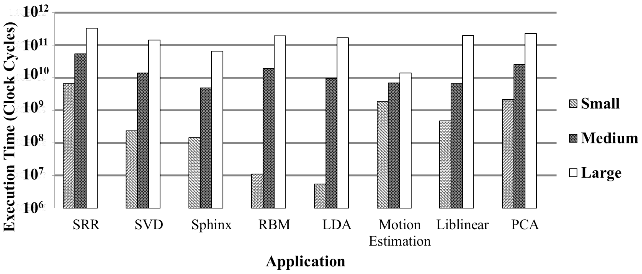

Fig. 2. ***Total Execution Times.*** To ensure kernel functions are sufficiently saturated, the large dataset for each algorithm was crafted to take a significant amount of time. This enabled us to see a trend in relative kernel execution time with increasing data. The total number of cycles is measured on an Intel Core i7-2620M CPU running at 2.70GHz.

III. Results and Scalability Analysis

A primary goal of the Synthetic Brain Benchmark Suite is to provide researchers with a platform for evaluating the scalability of various human-inspired data processing tasks. Toward that end we analyzed all applications in the benchmark suite to identify these regions of the program that act as performance bottlenecks as the dataset size increases. We identified which regions of the program dominate runtime as data scales exponentially, the so called kernels of the applications in Fig. 3. This data can be contrasted with each algorithm's absolute runtime, Fig. 2.

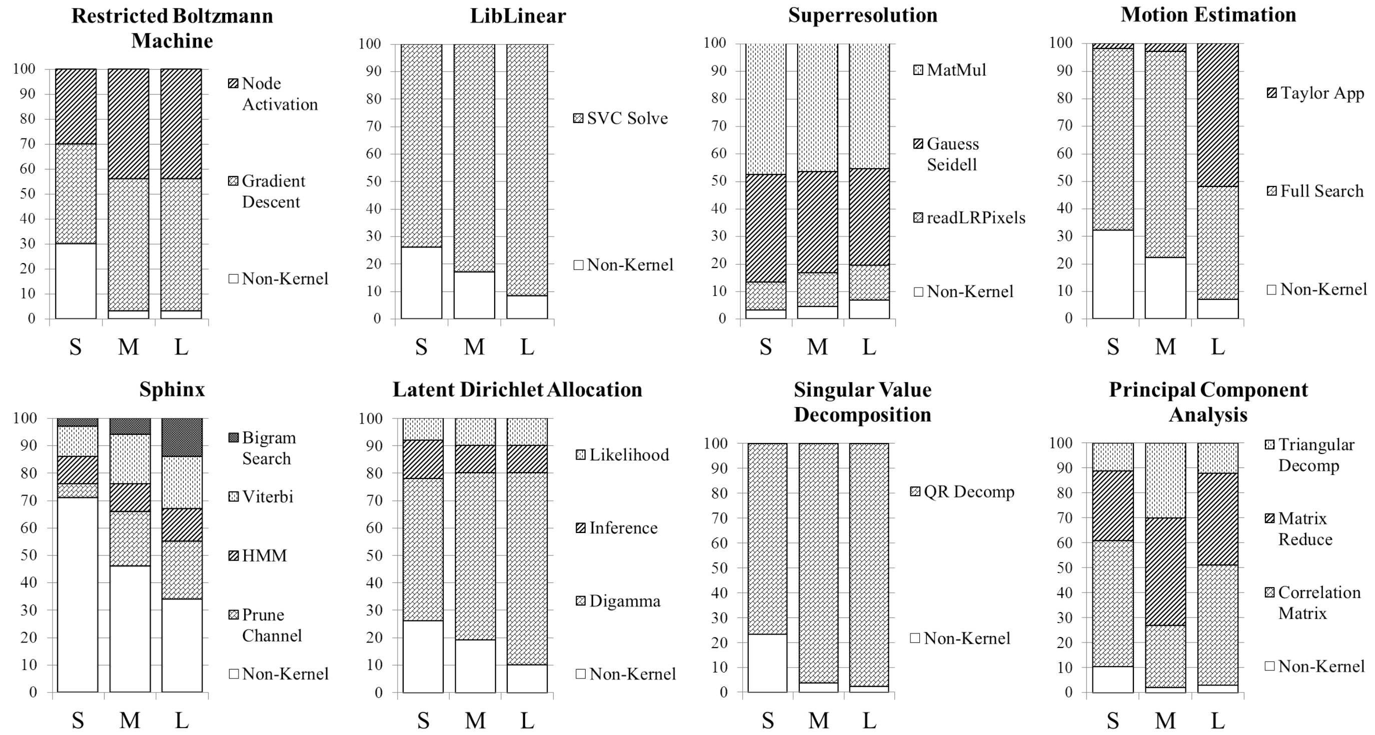

Fig. 3. ***Relative Execution Times.*** The graphs show the percentage of total execution time spent in each of the major kernels in the program. While we do not consider setup time and teardown time in the benchmark, we consider non-kernel regions which predominantly contain smaller and frequently used helper functions. Time spent in the kernels dominates, especially in the large dataset where the average is 90.50%.

CortexSuite differs from existing suites in that it strives to complete the picture of a synthetic brain while also providing architectural analysis and an analysis of scalability of each algorithm. The datasets used in this benchmark suite provide real-world applications of each algorithm within the suite.

IV. Conclusion and Future Work

CortexSuite captures a fascinating workload that is clustered around emerging algorithms in information processing, a domain traditionally relegated to the human brain. We classify and identify benchmarks within CortexSuite by analogy to the human neural processing function. We use the major lobes of the cerebral cortex as a model for the organization and classification of data processing algorithms. Our goal is not to emulate the brain at the level of the neuron, but rather to collect together man-made algorithms that have similar function and have met with success in the real world. This benchmark suite extends the SD-VBS vision benchmark suite with eight more brain-centered applications and provides an analysis of basic properties including the quantity of innate parallelism in each benchmark.

As the benchmark suite evolves we hope to help researchers get one step closer to realizing computers that outpace even human capabilities in this new domain. Our open-source benchmark suite and the datasets can be found at darksilicon.org/cortexsuite. An extended version of this paper can be found at darksilicon.org/cortexsuite/cortex-extended.pdf.

V. Acknowledgments

This work was partially supported by NSF Awards 0846152, 1018850, and 1228992, and by C-FAR, part of STARnet, a Semiconductor Research Corporation program.

References

[1] S. Venkata, I. Ahn, D. Jeon, A. Gupta, C. Louie, S. Garcia, S. Belongie, and M. Taylor, "SD-VBS: The San Diego Vision Benchmark Suite," in *IISWC*, Oct 2009.

[2] J. Clemons, H. Zhu, S. Savarese, and T. Austin, "Mevbench: A mobile computer vision benchmarking suite," in *IISWC*, Nov 2011, pp. 91–102.

[3] R. Salakhutdinov, A. Mnih, and G. Hinton, "Restricted boltzmann machines for collaborative filtering," in *Proceedings of the 24th International Conference on Machine Learning*, ser. ICML '07. New York, NY, USA: ACM, 2007, pp. 791–798.

[4] A. Tscher, M. Jahrer, and R. M. Bell, "The BigChaos Solution to the Netflix Grand Prize," 2009.

[5] X. Huang, F. Alleva, H.-W. Hon, M.-Y. Hwang, K.-F. Lee, and R. Rosenfeld, "The SPHINX-II speech recognition system: an overview," *Computer Speech & Language*, vol. 7, no. 2, pp. 137–148, 1993.

[6] S. C. Park, M. K. Park, and M. G. Kang, "Super-resolution image reconstruction: a technical overview," *Signal Processing Magazine, IEEE*, vol. 20, no. 3, pp. 21–36, May 2003.

[7] Q. Zhang, R. Guy, and R. Plemmons, "Matrix structures and parallel algorithms for image superresolution reconstruction," *SIAM Journal on Matrix Analysis and Applications*, vol. 31, no. 4, pp. 1873–1893, 2010.

[8] D. Blei, A. Ng, and M. Jordan, "Latent dirichlet allocation," *Journal of Machine Learning Research*, vol. 3, pp. 993–1022, January 2003.

[9] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "LIBLINEAR: A Library for Large Linear Classification," *J. Mach. Learn. Res.*, vol. 9, pp. 1871–1874, Jun. 2008.