Bridging Neuroscience and GPU Computing to build General Purpose Computer Vision

Nicolas Pinto[1], David Cox[2] and James DiCarlo[1]

1. The McGovern Institute for Brain Research and Department of Brain and Cognitive Sciences, Massachusetts Institute of Technology
2. The Rowland Institute at Harvard

The Rowland Institute at Harvard
HARVARD UNIVERSITY

Take home message:

***GPU Metaprogramming* and *Auto-tuning* dramatically accelerates the discovery of bio-inspired vision models that beat state-of-the-art computer vision systems**

1 Abstract

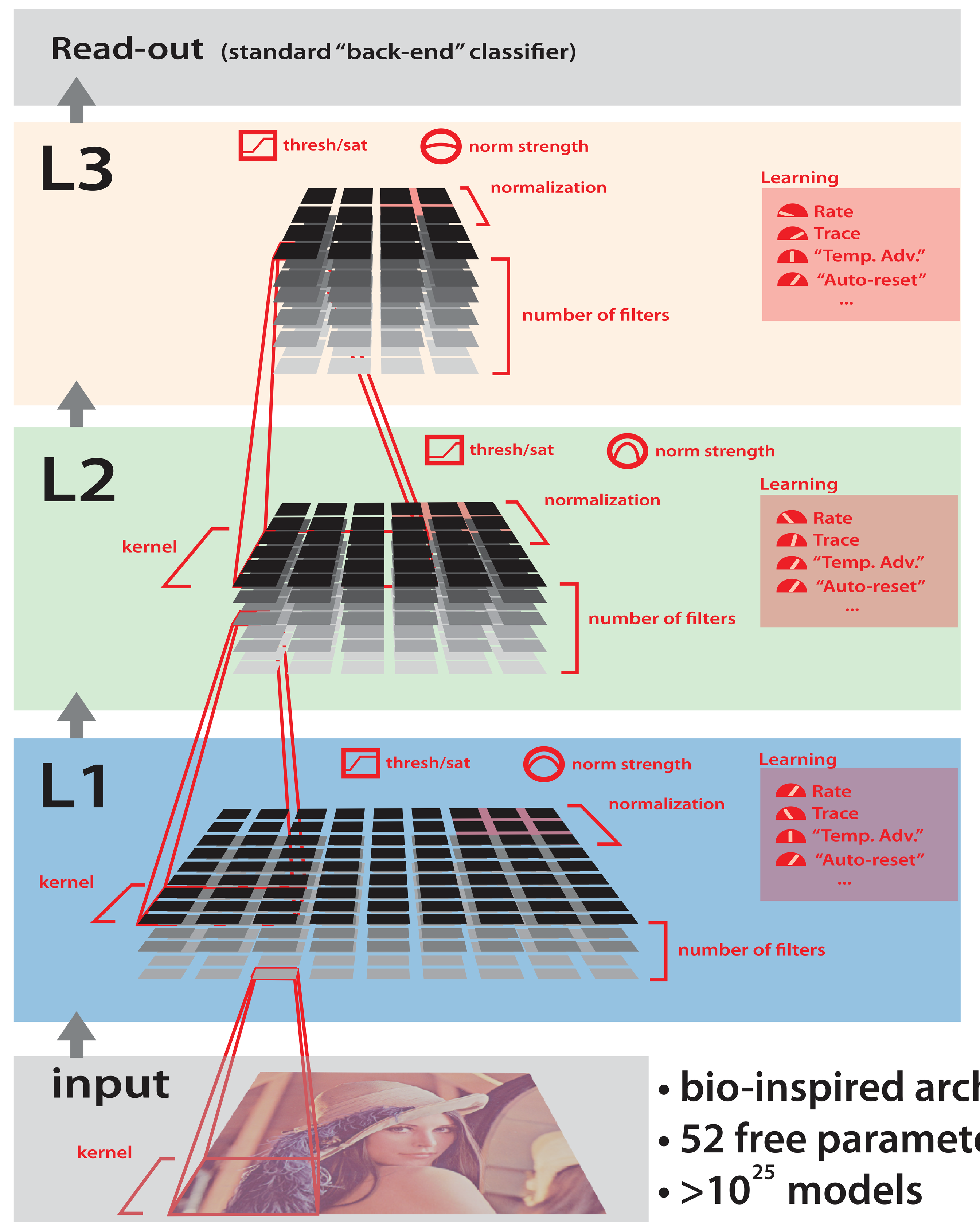
The study of biological vision and the creation of artificial vision systems are naturally intertwined – exploration of the neuronal substrates of visual processing provides clues and inspiration for artificial systems, and artificial systems, in turn, serve as important generators of new ideas and working hypotheses. However, while systems neuroscience has provided inspiration for some of the "broad-stroke" properties of the visual system, much is still unknown. Even for those qualitative properties that most biologically-inspired models share, experimental data currently provide little constraint on their key parameters. Consequently, it is difficult to truly evaluate a set of computational ideas, since the performance of a model depends strongly on its particular instantiation – the size of the pooling kernels, the number of units per layer, exponents in normalization operations, etc.

To pave a way forward, we have developed a high-throughput approach to more expansively explore the possible range of biologically-inspired models, including models of larger, more realistic scale, leveraging recent advances in commodity stream processing hardware - particularly high-end NVIDIA GPUs. In analogy to high-throughput screening approaches in molecular biology and genetics, we generated and trained thousands of potential network architectures and parameter instantiations, and "screened" the visual representations produced by these models using an object recognition task. From these candidate models, the most promising were selected for further analysis.

We show that this approach can yield significant, reproducible gains in performance across an array of basic object recognition tasks, consistently outperforming a variety of state-of-the-art purpose-built vision systems from the literature. We further show that this approach can be used to find feature representations that achieve excellent performance across a variety of test sets, including the modern "Labeled Faces in the Wild" unconstrained face recognition benchmark and a new large-scale face set derived from the popular Facebook social networking website.

We also highlight how the application of flexible programming tools, such as high-level scripting, template meta-programming and auto-tuning, can enable large performance gains, while managing complexity for the developer.

2 A Vast space of models to explore



- bio-inspired architecture
- 52 free parameters
- $>10^{25}$ models

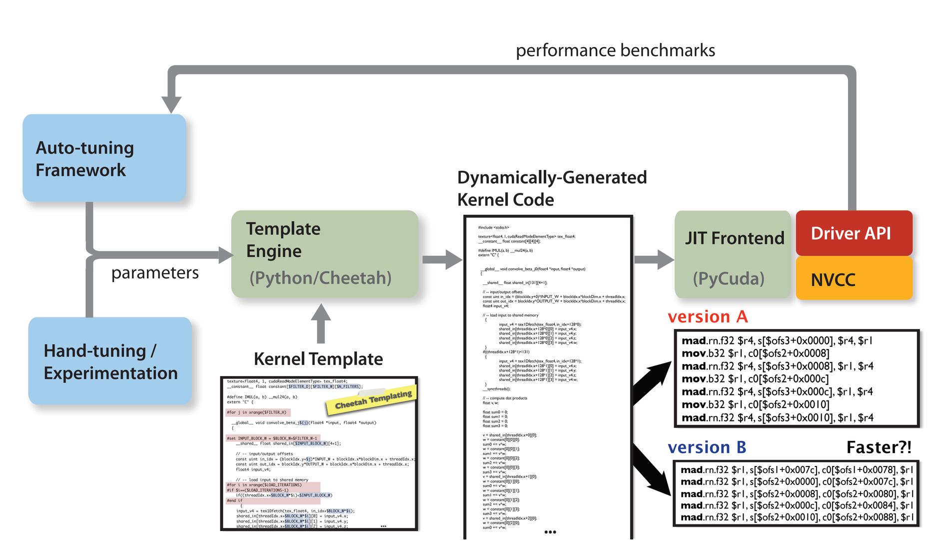
3 Metaprogramming

Recipe:

1) Use scripting (e.g. Python), templates and ***PyCUDA***,

2) Instrumentalize your code,

3) Let the computer auto-tune it!



4 GPU performance

Hardware	CPUs			GPUs			
Manufacturer	Intel	Intel	Intel	NVIDIA	Sony, IBM, Toshiba	NVIDIA	NVIDIA
Model	Q9450	Q9450	Q9450	7900 GTX	PlayStation 3	8800 GTX	GTX 280
# cores used	1	4	4	4x96	2+6	4x128	4x240
Implementation	MATLAB	MATLAB	SSE2	Cg	Cell SDK	CUDA	CUDA
Year	2008	2008	2008	2006	2007	2007	2008
Performance / Cost							
Full System Cost (approx.)	\$1,500**	\$2,700**	\$1,000	\$3,000*	\$400	\$3,000*	\$3,000*
Relative Speedup	**1x**	**4x**	**80x**	**544x**	**222x**	**1544x**	**2712x**
Relative Perf. / \$	**1x**	**2x**	**120x**	**272x**	**833x**	**772x**	**1356x**

GPU / SDK	Input	Filter-bank	Meta-prog *default* (gflops)	Meta-prog *auto-tuned* (gflops)	Boost
9600M GT CUDA3.1	256x256x8	64x9x9x8	6.710 ± 0.005	36.584 ± 0.023	**445.2 %**
	512x512x4	32x13x13x4	13.606 ± 0.002	35.582 ± 0.003	161.5 %
	1024x1024x8	16x5x5x8	20.034 ± 0.113	26.084 ± 6.243	30.2 %
	2048x2048x4	4x8x8x4	25.781 ± 0.044	46.945 ± 0.100	82.1 %
C1060 CUDA2.3	256x256x8	64x9x9x8	104.188 ± 0.051	168.083 ± 0.372	61.3 %
	512x512x4	32x13x13x4	125.739 ± 0.109	234.053 ± 0.266	86.1 %
	1024x1024x8	16x5x5x8	144.279 ± 0.764	243.697 ± 0.346	68.9 %
	2048x2048x4	4x8x8x4	180.060 ± 0.018	322.328 ± 0.348	79.0 %
GTX285 CUDA2.3	256x256x8	64x9x9x8	123.396 ± 0.016	197.006 ± 0.219	59.7 %
	512x512x4	32x13x13x4	143.277 ± 0.044	270.206 ± 0.209	88.6 %
	1024x1024x8	16x5x5x8	148.841 ± 0.465	310.276 ± 0.538	108.5 %
	2048x2048x4	4x8x8x4	205.152 ± 0.015	376.685 ± 0.070	83.6 %
GTX480 CUDA3.1	256x256x8	64x9x9x8	467.631 ± 19.100	471.902 ± 11.419	0.9 %
	512x512x4	32x13x13x4	834.838 ± 8.275	**974.266** ± 3.809	16.7 %
	1024x1024x8	16x5x5x8	542.808 ± 1.135	614.019 ± 0.904	13.1 %
	2048x2048x4	4x8x8x4	378.165 ± 0.537	806.628 ± 0.168	113.3 %

Pinto, Cox (Submitted)

5 Proof of concept: High-throughput screening

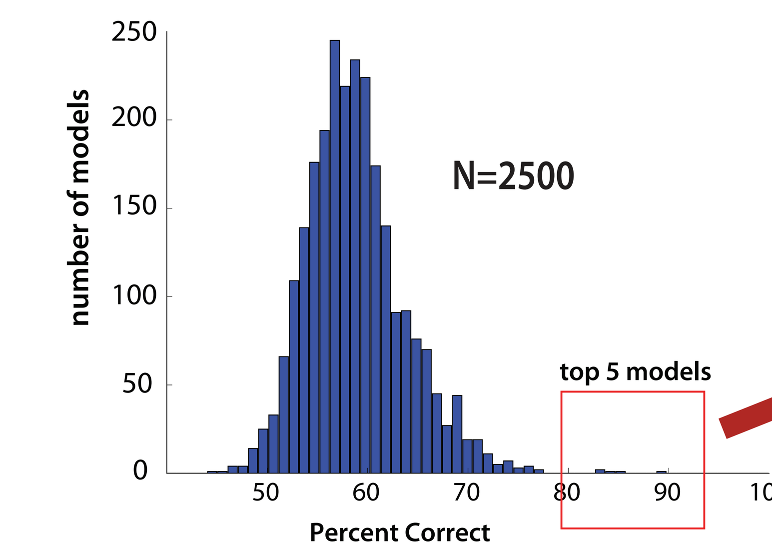
Generate Random Models → Unsupervised Learning → Test with a "screening" object recognition task → Choose the best models → Validate on other tasks

Pinto, Doukhan, DiCarlo, Cox (PLoS 2009)

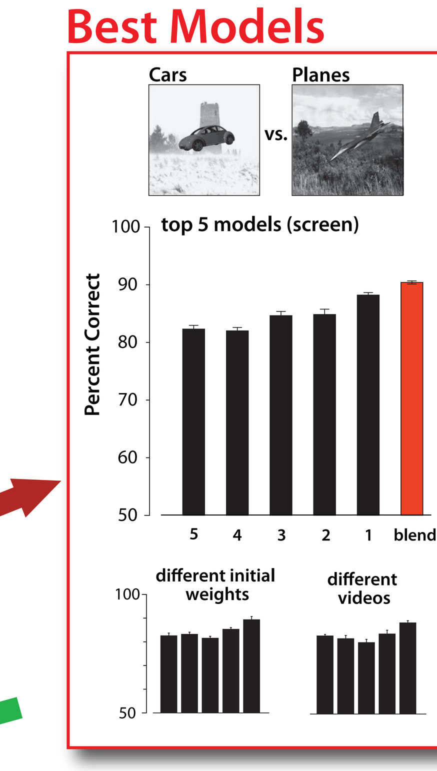
Unsupervised Learning

Law & Order "World"

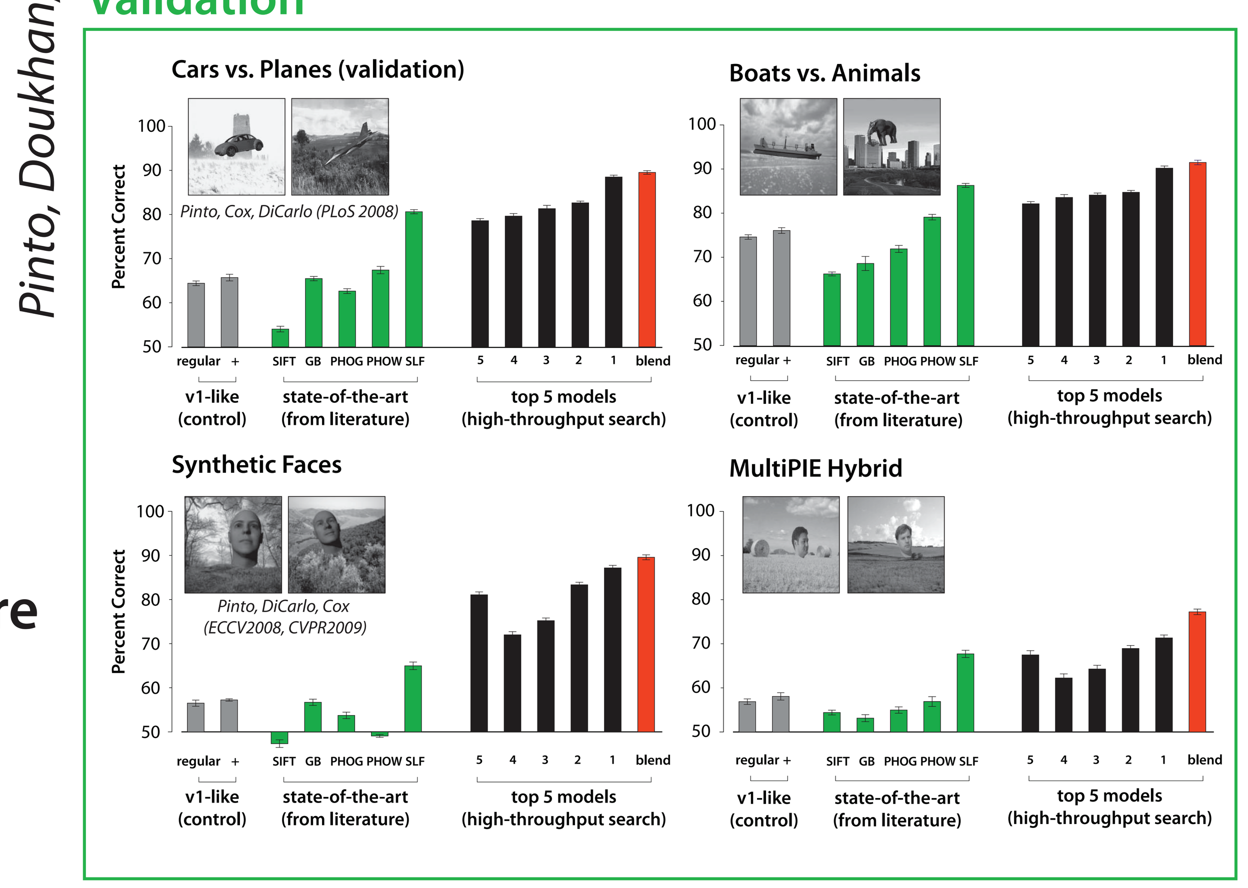
Screen Distribution



Best Models



Validation



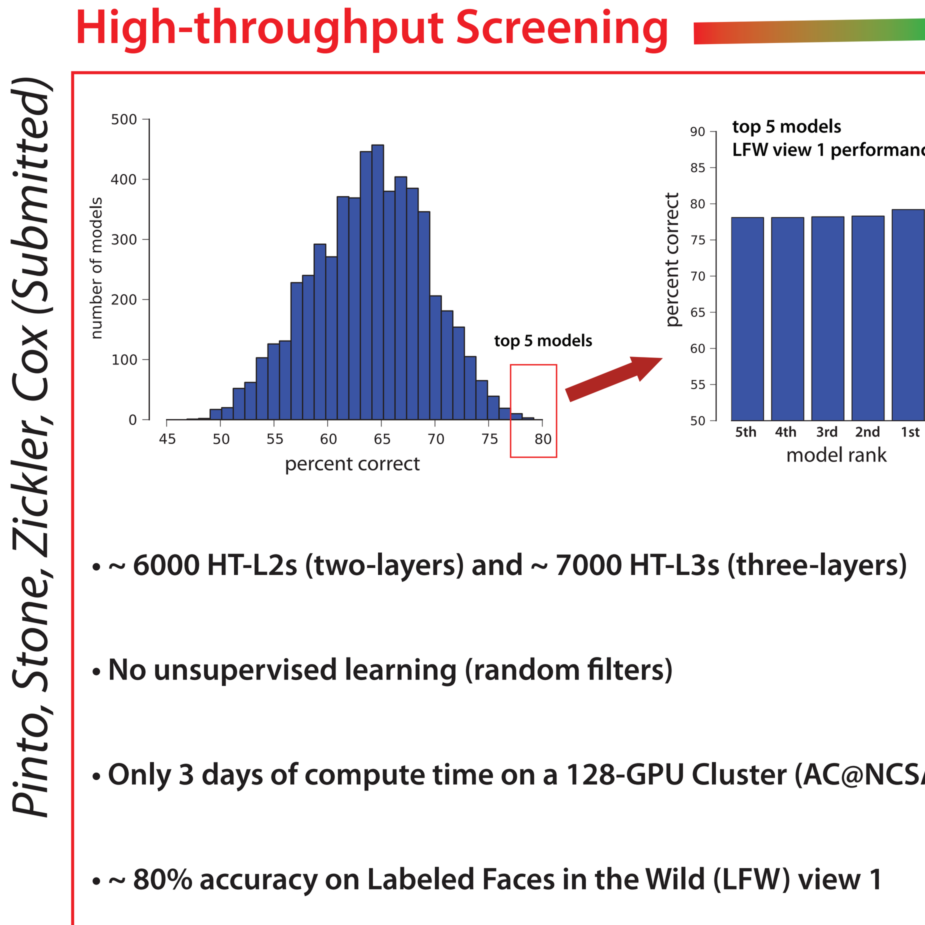
We discovered models that outperform the state-of-the-art in computer vision...

(object and face recognition)

6 Best models on LFW and Facebook100 !

High-throughput Screening → State-of-the-art Performance

Pinto, Stone, Zickler, Cox (Submitted)



- ~ 6000 HT-L2s (two-layers) and ~ 7000 HT-L3s (three-layers)
- No unsupervised learning (random filters)
- Only 3 days of compute time on a 128-GPU Cluster (AC@NCSA)
- ~ 80% accuracy on Labeled Faces in the Wild (LFW) view 1