Title: Large-Scale Functional Models of Visual Cortex for Remote Sensing

Author(s): Steven P. Brumby, Garrett Kenyon, Will Landecker, Craig Rasmussen, Sriram Swaminarayan, and Luís M. A. Bettencourt

Intended for: Proceedings of AIPR Workshop 2009

Large-Scale Functional Models of Visual Cortex for Remote Sensing

Steven P. Brumby*[a], Garrett Kenyon[a], Will Landecker[b], Craig Rasmussen[a],
Sriram Swaminarayan[a], and Luís M. A. Bettencourt[a]
a. Los Alamos National Laboratory
Mailstop D436, Los Alamos, NM 87545 USA
b. Portland State University
PO Box 751, Portland, OR 97207-0751

Abstract— **Neuroscience has revealed many properties of neurons and of the functional organization of visual cortex that are believed to be essential to human vision, but are missing in standard artificial neural networks. Equally important may be the sheer scale of visual cortex requiring ~1 petaflop of computation. In a year, the retina delivers ~1 petapixel to the brain, leading to massively large opportunities for learning at many levels of the cortical system. We describe work at Los Alamos National Laboratory (LANL) to develop large-scale functional models of visual cortex on LANL's Roadrunner petaflop supercomputer. An initial run of a simple region V1 code achieved 1.144 petaflops during trials at the IBM facility in Poughkeepsie, NY (June 2008). Here, we present criteria for assessing when a set of learned local representations is "complete" along with general criteria for assessing computer vision models based on their projected scaling behavior. Finally, we extend one class of biologically-inspired learning models to problems of remote sensing imagery.**

I. Introduction

Neuroscience has revealed many properties of individual neurons and of the functional organization of visual cortex that are believed to be essential to reach human vision performance, but are missing in standard artificial neural networks. Among these are extensive lateral and feed-back connectivity between neurons, spiking dynamics of neurons, and spike timing dependent plasticity (STDP) of synapses.

Equally important may be the enormous scale of visual cortex: ~10 billion neurons each with ~10 thousand synaptic connections, a simple simulation (10 flop/neuron to process one frame of data) therefore requiring ~1 petaflop (10^{15} flop) of computation. In a year, the 6 million cones in the retina and ~1 million fibers in the optic nerve deliver ~1 petapixel to the brain. A recent biologically-inspired model of visual cortex regions V1-V4 and inferotemporal (IT) cortex (Serre, et al., [1,2], based on Fukushima's "Neocognitron" [3] and "HMAX" models of Poggio, et al., [4]) operating at a scale of ~10 million feed-forward neurons on a ~billion pixel library of images [5], was compared to human performance on a binary classification task in a speed-of-sight psychophysics experiment: detection of animal/no animal. Under these conditions, in which subjects viewed images for a little as 20 msec before substitution of a 1/f noise mask (implying that feed-forward pathways in cortex are likely to dominate), the HMAX/ Neocognitron model achieved accuracy comparable to the human subjects (~80% accuracy, and *d'* performance score of ~2.2 [1,6]).

Human performance on the Speed of Sight task improves to near perfect accuracy as stimulus presentation time increases, thus raising a number of questions: can these hierarchical feed-forward models ultimately match human performance under natural viewing conditions as we scale the model to match the full size of human visual cortex? Do we need to train these models with petascale image datasets? How much of this training data needs to be labeled? How many distinct object categories are required to support synthetic visual cognition?

Computing hardware to support full-scale implementations of feed-forward hierarchical models now exists. The Synthetic Cognition team at Los Alamos National Laboratory is developing large-scale functional models of visual cortex that can operate on its Roadrunner petaflop supercomputer [7]. An initial run of a simple V1 code achieved 1.144 petaflops during trials at the IBM facility in Poughkeepsie, NY (June 2008). The goal of this research is to build a full-scale functional model of visual cortex that can process high definition video (1080p) in real time.

In addition to standard computer vision problems, we are also interested in exploring application of these models to remote sensing datasets of satellite and aerial imagery. Previous attempts to apply biologically inspired algorithms to satellite imagery [8,9] focused on retinal and V1 features such as edge and bar detectors, color opponency, and edge continuation, provided as input to an ARTMAP supervised classification algorithm [10]. Pinto, Cox and DiCarlo [11] have argued that V1 alone, which characterizes only local features in images (about 1 degree angle resolution) is probably not enough to capture the invariant representations of

* Work supported by Department of Energy (LDRD-DR-2009006) and National Science Foundation (Award No. NSF-OCI-0749348).

objects necessary to interpret natural scenes, and explicitly demonstrated classification accuracy breakdown using rendered image sets where the same object is presented over a large number of poses, viewpoints and backgrounds. Here, we present preliminary evidence to support several hypotheses governing biologically-inspired learning. Specifically, we address the general question of how to determine when the learned-representations over a restricted region of visual space is "complete" by analyzing the frequency with which individual features are activated by natural images. We also illustrate a general method for comparing different computer vision models by examining their scaling behavior as a function of training set size, as opposed to the more common method of comparing performance at a single scale. Finally, we explore how hierarchical models of the ventral pathway can be applied to object detection in overhead imagery.

II. Description of the Model

A. Base model.

We constructed a hierarchical model of visual cortex based on the architecture of the Neocognitron [3], as developed by HMAX models [4]. This class of model consists of alternating layers of "simple" and "complex" cells inspired by Hubel and Wiesel's model of primary visual cortex [12].

The input to our model is a grayscale image, padded by enough zeros to allow efficient looping over the image. We do not impose size constraints on the image, and have tested our C++ code, called PANN (Petascale Artificial Neural Network), with a range of image sizes.

We impose retina-like contrast equalization by carrying out a local contrast adjusting transformation. For each pixel in the image, we consider its neighborhood patch, $\mathbf{x} = \{x_i\}$. This patch is shifted and scaled to have zero mean ($\langle x \rangle = \Sigma x_i = 0$) and unit norm ($\|x\| = \Sigma x_i^2 = 1$). The central pixel value is then kept. A regularization term is used to discount contributions from image regions with negligible contrast.

Visual regions V1 and V2 are modeled with a columnar organization of S and C cells, corresponding to simple and complex cells in V1 and to their generalizations in higher visual areas. Each column sees the same receptive field size over its input (retina or V1, respectively). Each S cell has a tuning, specified by a synaptic weight vector, and a tuning width, which defines the feature to which that S cell is sensitive. In principle, each column could develop a unique set of weight vectors and tuning widths, but in practice the models are greatly simplified by the imposition of a fixed tuning width for all S cells, and by the imposition of translational invariance by making all the columns in a layer identical.

S cells are implemented as radial basis functions,

$$s_j = g_{\text{RBF}}\left(\boldsymbol{w}_j, \boldsymbol{x}\right) = \exp\left(-\left(\boldsymbol{w}_j - \boldsymbol{x}\right)^2 / 2\sigma\right),$$

where $\boldsymbol{x} = \{x_i\}$ is the input to the S cell, $\boldsymbol{w}_j$ represents the synaptic weights, and σ parameterizes the bell-shaped tuning of the neuron. The input $\boldsymbol{x}$ has its mean subtracted ($\langle \boldsymbol{x} \rangle = 0$) and is normalized ($\|\boldsymbol{x}\|=1$). Alternative implementations for S cells are described in [13].

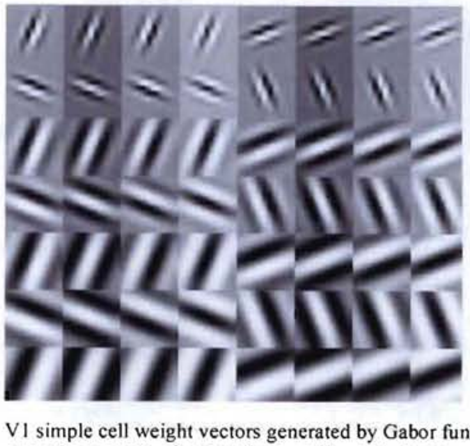

Fig. 1. V1 simple cell weight vectors generated by Gabor functions covering a range of orientations, scales, and phases.

To represent the observed range of edge-tuned and bar-tuned neurons in primary visual cortex [14,15], it is common [1] to choose the weight vectors $\boldsymbol{w}_j$ to be a set of Gabor functions parameterized by orientation θ, eccentricity γ, envelope width σ, spatial wavelength λ, and phase ϕ,

$$w_j\left(\theta,\gamma,\sigma,\lambda,\phi\right) = \exp\left(-\left(x_\theta^2 + \gamma y_\theta^2\right)/2\sigma\right)\cos\left(\frac{2\pi x_\theta^2}{\lambda} + \phi\right),$$

where the pixel co-ordinates (x,y) have been rotated through an angle θ to give (x_θ, y_θ). The weight vector has zero mean ($\langle \boldsymbol{w}_j \rangle = 0$) and unit norm ($\|\boldsymbol{w}_j\| = 1$). Typical values for these parameters are set by reference to experimental data [1], and Figure 1 shows a set of such weight vectors for V1.

C cells are implemented as winner-take-all *max* functions over the cell's receptive field,

$$c_j = g_{\text{MAX}}\left(s\right) = \max\left(\left\{s_{i\in N_j}\right\}\right),$$

where the input $\boldsymbol{s}$ is a patch of responses from the previous S cell layer, and N_j defines the receptive field of the j^{th} C cell. Alternative implementations and learning rules for complex cells in V1 are discussed in [16-18].

In cortical region V2, the S cell layer is again implemented as a layer of radial basis functions with input from the complex cell layer of V1, $s^{II}_j = g_{\text{RBF}}(w^{II}_j, c^I)$. There is no standard closed-form expression for calculating V2 S cell weight vectors, and so these are learned though either "imprinting" (memorization of patches of V1 complex cell output produced during presentation of a training set of images [1,2]) or other biologically inspired learning rules as described below.

The final stage of this model representing inferotemporal cortex (IT) is implemented using a supervised classification algorithm, typically a support vector machine (SVM) [19]. We use the standard LIBSVM package [20]. To reduce the complexity of the image representation given to the support vector machine, the final C cell layer is a global max over each of the features learned by the final S cell layer [1,2].

Additionally, in keeping with estimates of the number of neurons in different regions of visual cortex, the spatial extent of each layer is reduced by a down-sampling factor that should match the increase in the number of features calculated in a cortical column over each pixel. We use a down-sampling value of 2, consistent with experimental data showing the growth of neuron receptive fields between layers [21,22].

B. Unsupervised Learning in V1, V2.

Published results with HMAX/Neocognitron models have used a pre-selected set of Gabor tunings for V1 neurons, and imprinting of neurons in V2 [1,2]. Alternatively, several learning rules to compute these representations have been proposed [16-18].

One of the simplest of these is a modified Hebbian learning rule [23,24], applied to feed-forward hierarchical models in [16,17],

$$\Delta \boldsymbol{w}_j = \alpha \cdot y_j \cdot \left(\boldsymbol{x} - y_j \cdot \boldsymbol{w}_j \right),$$

where y_j is the activity of the neuron in response to input $\boldsymbol{x}$, and α is a parameter controlling the learning rate.

In V1, a simple cell with receptive field size $M \times M$ observing Boolean-valued pixels (a lower bound) can receive $2^{M \times M}$ possible input patches. If imprinting is a reasonable leaning rule, then either V1 needs sufficient capacity to learn a significant fraction of these patterns, or else the statistics of natural images have to be such that the most important patterns are the most frequently observed.

V1 receives input from ~1M fibers in the optic nerve projecting via lateral geniculate nucleus (LGN). Of the ~150M neurons in V1, ~30M project forward to V2. Given the spatial down-sampling in V1, we therefore expect that a cortical column in V1 has ~100-1000 neurons tuned to features of the receptive field, much less than the possible number of input patches for small receptive fields, e.g., M=5 requires a memory of $2^{25} = 10^{7.5}$ to imprint all possible patterns. This situation becomes geometrically worse in V2 (and higher regions of visual cortex), where an S cell can in principle receive input over a spatial neighborhood of cortical columns each with F features, giving $2^{M \times M \times F}$ possible input states.

Learning rules can improve this situation, by allowing the S cells in each cortical column to adapt their tuning during presentation of a large amount of training data. The issue becomes how much training data is required to stabilize the tunings of a population of neurons in a cortical column.

We consider a model of V1 with on-line Hebbian learning, with additional conditions as follows. We initialize the V1 column by imprinting each of the F simple cells in the cortical column. On presentation of an input $\boldsymbol{x}$, we calculate the activity of each simple cell,

$$\forall j = 1 \ldots F,\ s_j = g_{\text{RBF}}\left(\boldsymbol{w}_j, \boldsymbol{x} \right),$$

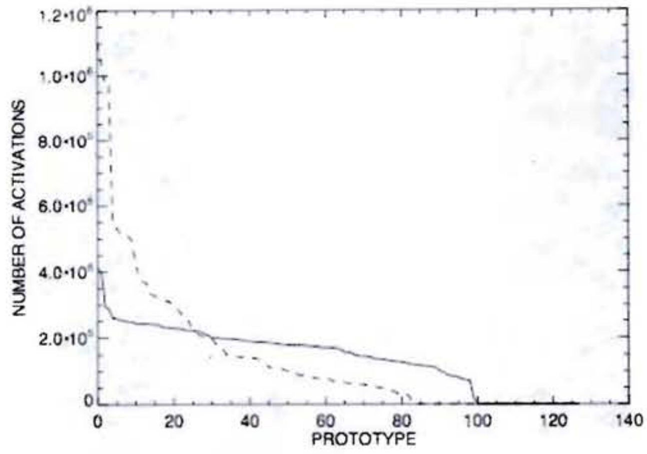

Fig. 2. Number of activations per prototype for Hebbian learning (solid line) and imprinting (dashed line) in response to 18 million input patches.

The neuron j^* with the largest activation s_{j^*} is selected (winner-take-all) for updated according to the Hebbian learning rule,

$$\Delta \boldsymbol{w}_{j^*} = \alpha \cdot s_{j^*} \cdot \left(\boldsymbol{x} - s_{j^*} \cdot \boldsymbol{w}_{j^*} \right),$$

and is then renormalized to unit norm ($\|\boldsymbol{w}_{j^*}\| = 1$). As the input pattern is mean zero ($\langle \boldsymbol{x} \rangle = 0$), Hebbian learning ensures that the weight vector has zero mean ($\langle \boldsymbol{w}_{j^*} \rangle = 0$).

To explore convergence of the cortical column tunings in V1, we extracted 18 million 5 × 5 pixel patches of images from 600 animal/no animal images from the AnimalDB dataset [25]. Patches are sampled randomly with replacement, and we expect there to be ~10 copies of each patch in the training set. We then train a set of 128 simple cells and keep track of the number of times each patch is selected for update by the winner-take-all competition. For comparison, we imprint 128 simple cells by drawing randomly from the collected patches, and then determine how many times each of these unmodified patches would have won the winner-take-all competition between simple cells.

Figure 2 shows the number of activations per prototype for Hebbian learning and for imprinting. We see that the distribution of activations flattens substantially with Hebbian learning, suggesting that with enough input patterns the column of weight vectors will converge to a population of nearly equally active units. Only 100 out of the possible 128 prototypes are active with non-negligible frequency, suggesting that this size set of features can represent the input drawn from natural scenes.

Figure 3 shows these imprinted and Hebbian-learned prototypes sorted by number of activations (most active are shown first). Similar to the results in [17], we see that the Hebbian rule learns a set of Gabor-like oriented edge and bar weight vectors. These results suggest that by examining the frequency distribution with which a column of S cells are activated over a given image database, the minimum number of feature prototypes necessary to form a "complete" set can be empirically determined.

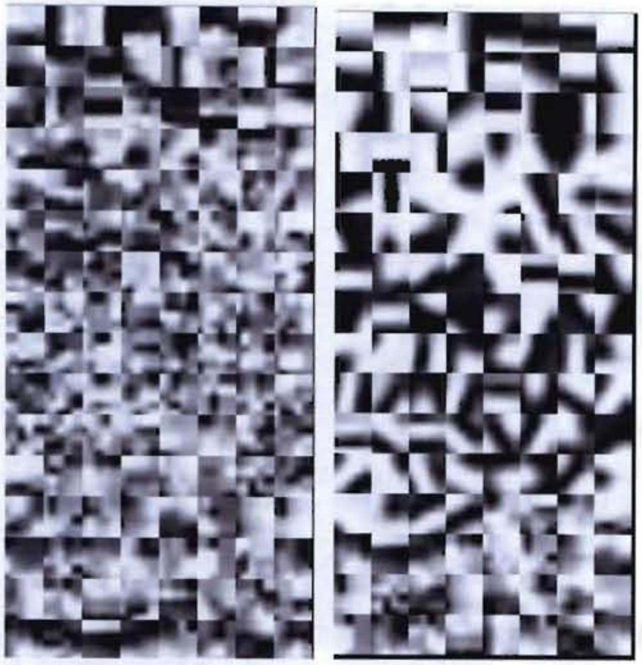

Fig. 3. V1 simple cell weight vectors learned by imprinting (left) and by Hebbian learning (right), sorted by number of activations (Fig.2). Hebbian rule learns a set of Gabor-like weight vectors (cf., Fig.1).

C. Supervised Learning in IT.

The performance of the model with imprinted and learned prototypes can be compared in terms of object classification accuracy in a standard supervised learning task. HMAX/ Neocognitron models are used to classify whole images with a single label. The IT layer of the model converts the final C cell layer into a 1 dimensional vector, where the length of the vector is equal to the number of S cells in the column. This vector is sent to support vector machine for classification. Serre, et al., [1,2], demonstrate their model on Caltech Vision Group datasets (Caltech 101 [5]) and an animal/no-animal dataset [25]. As shown in Figure 4, we achieve similar classification results on object categories from the newer Caltech 256 dataset [26]. The animal/no-animal dataset contains 1200 images, enabling an investigation of the behavior of the classifier as the training set size increases.

Figure 5 shows the result of training on different size sets of images while testing on a fixed set of testing images (not seen during training). Performance on the test set is seen to increase, and suggests that perfect performance (100% accuracy) will be achieved at a finite number of training images. If we compare the behavior of a classifier trained using the standard fixed Gabor V1 and imprinted V2, versus Hebbian learned V1 and V2, we see that the fully learned model starts at a lower accuracy, but eventually matches performance of the standard model as the training set grows.

If we fit these models with a simple functional form [27],

$$A = a + b\ln\left(N_T\right),$$

where A is the accuracy, N_T is the size of the training set, and a and b are constants. We can extrapolate these graphs to calculate the critical training set size at which accuracy reaches 100%. For the standard model, this value is N_T = 3996, whereas for the fully learned model N_T = 3130, suggesting that learning V1 and V2 can improve the performance of the model with sufficiently large datasets. These results suggest a general method for comparing different computer vision models, a criteria based not on performance at a single scale but on how performance improves as the models themselves are scaled to better match corresponding biological systems.

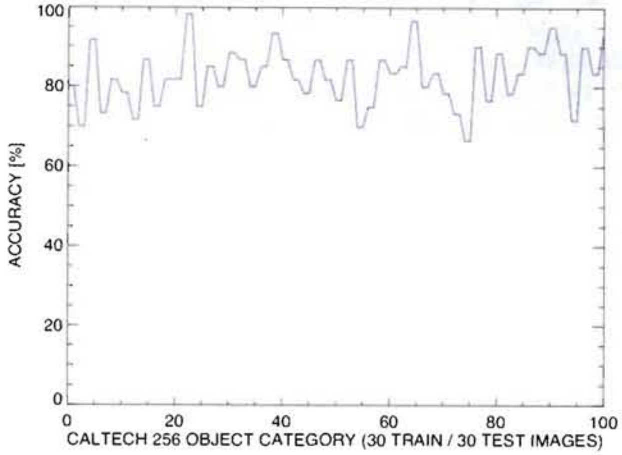

Fig. 4. Accuracy achieved on 100 categories drawn from the Caltech 256 object identification dataset (test-set accuracy for supervised binary-classification of object images versus background images).

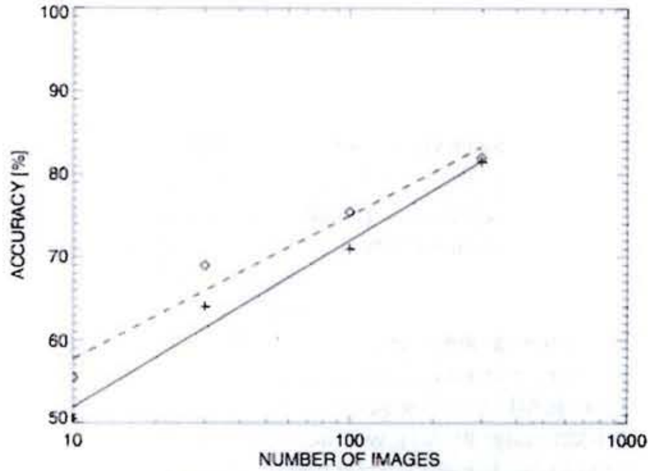

Fig. 5. Accuracy for supervised binary-classification of animal versus no animal images as a function of training set size for a visual cortex using imprinting (open diamonds and dashed line) or Hebbian learning (crosses and solid line). Lines show fit of a simple model, $A = a + b\ \ln(N_T)$.

D. Modification of the model for remote sensing data

For remotely sensed imagery, whole image classification needs to be modified to allow local detection of objects of interest. This can be achieved by modifying the final complex cell layer to calculate a *local max* rather than the standard *global max* operation. The set of *local max* vectors can then be sent a standard supervised classification algorithm to generate a local decision on the presence or absence of the object of interest. Note that the samples sent to the supervised classifier are not independent (and so are not independent and

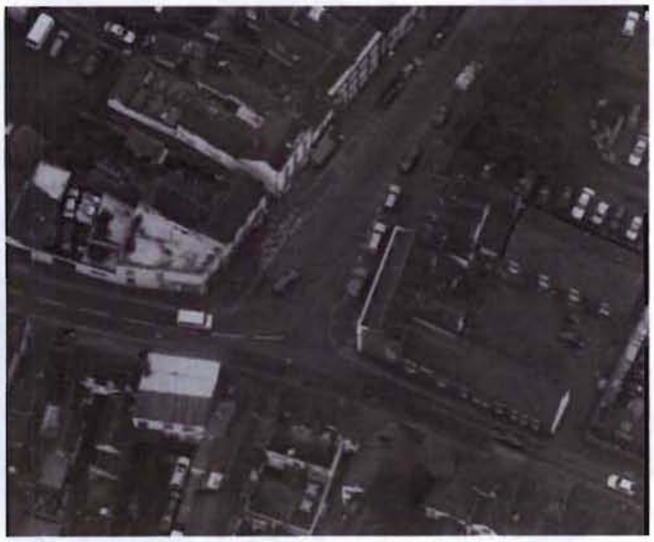

Fig. 6. Panchromatic aerial image over a typical urban scene.

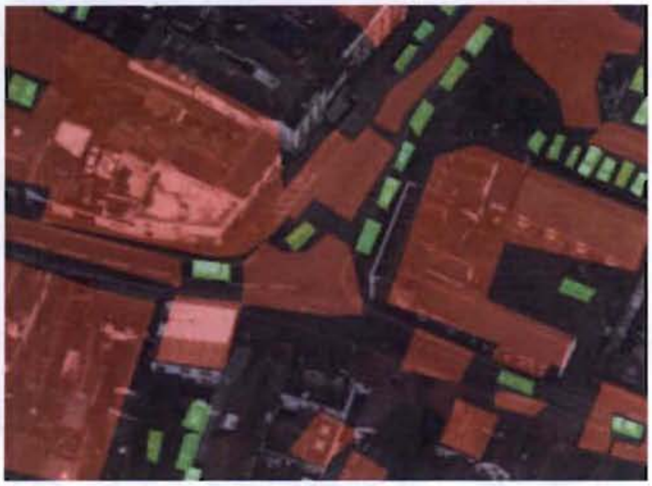

Fig. 7. Part of the training markup for a vehicle detection task. Target pixels marked green and background pixels marked red.

identically distributed, i.e., i.i.d.), but as is common in image analysis, we treat these samples as though they were.

Figure 6 shows a patch of panchromatic aerial imagery over a typical urban setting. Ground sample distance (GSD) for the scene is ~0.3m, widely available from commercial aerial platforms. As a demonstration of our approach, we consider the problem of pixel-level vehicle extraction from this imagery.

We marked up a region of the image (Figure 7) and trained a model using a set of 128 V1 simple cells and 256 V2 S cells, with all weight vectors learned using the modified Hebb rule. The whole image is 1268×1012 pixels, of which 48,320 are marked as target pixels and 403,154 are background pixels. With this mark up, after V2 processing the SVM classifier received 753 training samples (543 positive and 2031 negative). Classifier accuracy on the training set of V2 complex cell output was 89.1%. Projected back to the input image, the pixel-level accuracy was 96.0% (90% detection rate (DR) and 3.5% false alarm rate (FAR), Figure 8).

Figure 9 shows the result of applying the fully trained V1-V2 model to a test image. There were 1511 testing samples (204 positive and 1307 negative). The classifier achieves an accuracy of 86.9% at the V2 complex cell level, and a pixel-level accuracy of 92.5% (30.4% DR, 4.0% FAR). Receiver Operating Characteristic (ROC) curves for training and testing are shown in Figure 10.

III. Discussion and Future Work

Hierarchical, feed-forward models of Neocognitron/HMAX type can be applied to remotely sensed overhead imagery by modifying the final C cell layer to use *local max* operations. Biologically-inspired learning rules, such as the modified Hebb rule described in section II.B above, can compete with the use of predefined feature banks and imprinting. In future work, we will explore adding color and motion visual pathway channels to these models.

We have shown that under the operation of a learning rule, and given enough unlabeled training data, the weight vectors of a cortical column can converge to a distribution of nearly equally active units. Those units active after convergence of the population provide a "complete" representation of the input patterns present at that level of the model. This principle can be applied to the training of successive stages of a hierarchical model. Further, we have argued that model builders should consider the scaling behavior of the performance of the model as the size of the training and testing sets increase. Thorough investigation of scaling behavior greatly increases the computational requirements of studies. In future work we will explore hardware-accelerated implementations of this models that can exploit new petascale computing resources at Los Alamos National Laboratory.

References

[1] T. Serre, A. Oliva and T. Poggio. A feedforward architecture accounts for rapid categorization. Proceedings of the National Academy of Science, 104(15), pp. 6424-6429, April 2007.

[2] T. Serre, L. Wolf, S. Bileschi, M. Riesenhuber and T. Poggio. Object recognition with cortex-like mechanisms. In: IEEE Transactions on Pattern Analysis and Machine Intelligence, 29 (3), pp. 411-426 , 2007.

[3] K. Fukushima: Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position, Biological Cybernetics, 36(4), pp. 193-202 (April 1980).

[4] Riesenhuber, M. & Poggio, T. (1999). Hierarchical Models of Object Recognition in Cortex, Nature Neuroscience 2: 1019-1025.

[5] L. Fei-Fei, R. Fergus and P. Perona. Learning generative visual models from few training examples: an incremental Bayesian approach tested on 101 object categories. IEEE. CVPR 2004, Workshop on Generative-Model Based Vision. 2004.

[6] Brophy, A. L. (1986). Alternatives to a table of criterion values in signal detection theory. Behavior Research Methods, Instruments, & Computers, 18, 285-286.

[7] Paul Henning and Andrew White, Trailblazing with Roadrunner, Computing in Science & Engineering, July/August 2009, pp. 91-95.

[8] M. Chiarella, D. Fay, R. Ivey, N. Bomberger, A. Waxman, Multisensor image fusion, mining, and reasoning: Rule sets for higher-level AFE in a COTS environment, in: Proceedings of the 7th International Conference on Information Fusion, Stockholm, Sweden, June 2004, pp. 983-990.

[9] A.M. Waxman, J.G. Verly, D.A. Fay, F. Liu, M.I. Braun, B. Pugliese, W. Ross, and W. Streilein, A prototype system for 3D color fusion and mining of multisensor/spectral imagery, in 4th International Conference on Information Fusion, Montreal, 2001

[10] Carpenter, G. A., Gjaja, M. N., Gopal, S., & Woodcock, C. E. (1997). ART neural networks for remote sensing: vegetation classification from Landsat TM and terrain data. IEEE Transactions on Geoscience and Remote Sensing, 35, 308-325.

[11] Pinto N, Cox DD, and DiCarlo JJ. Why is Real-World Visual Object Recognition Hard? PLoS Computational Biology, 4(1):e27 (2008).

[12] Hubel, D.H., Wiesel, T.N.: Receptive fields and functional architecture of monkey striate cortex. J. Physiol. (Lond.) 195, 215–243 (1968).

[13] Minjoon Kouh, Tomaso Poggio, A canonical neural circuit for cortical nonlinear operations. *Neural computation*, Vol. 20, No. 6. (6 June 2008), pp. 1427-1451, 2008.

[14] Földiák, P. (1990). Forming sparse representations by local anti-hebbian learning. Biol Cybern, 64(2):165–170.

[15] Olshausen, B. A. and Field, D. J. (1996). Emergence of simple-cell receptive field properties by learning a sparse code for natural images. Nature, 381:607–609.

[16] Masquelier T and Thorpe SJ (2007). Unsupervised Learning of Visual Features through Spike Timing Dependent Plasticity. PLoS Comput Biol 3(2): e31 doi:10.1371/journal.pcbi.0030031

[17] Masquelier T, Serre T, Thorpe SJ and Poggio T (2007). Learning complex cell invariance from natural videos: a plausibility proof. CBCL Paper #269/MIT-CSAIL-TR #2007-060, MIT, Cambridge, MA.

[18] Einhäuser, W., Kayser, C., König, P., and Körding, K. P. (2002). Learning the invariance properties of complex cells from their responses to natural stimuli. Eur J Neurosci, 15(3):475–486.

[19] C. Cortes & V. Vapnik, Support-Vector Networks, Machine Learning, 20, 273-297 (1995).

[20] Chang, C.-C. and C.-J. Lin (2001). LIBSVM: a library for support vector machines. Available: http://www.csie.ntu.edu.tw/~cjlin/libsvm

[21] R. DeValois, D. Albrecht, and L. Thorell, Spatial Frequency Selectivity of Cells in Macaque Visual Cortex, Vision Research, vol. 22, pp. 545-559, 1982.

[22] R. DeValois, E. Yund, and N. Hepler, The Orientation and Direction Selectivity of Cells in Macaque Visual Cortex, Vision Research, vol. 22, pp. 531-544, 1982.

[23] Hebb, D.O., The organization of behavior, New York: Wiley, 1949.

[24] Oja, Erkki, Simplified neuron model as a principal component analyzer, Journal of Mathematical Biology 15 (3): 267–273, Nov 1982.

[25] A. Torralba and A. Oliva, Statistics of natural image categories. In Network: computation in neural systems. Vol. 14, 391-412., 2003.

[26] Griffin, G. Holub, AD. Perona, P. The Caltech-256, Caltech Technical Report, 2006. http://www.vision.caltech.edu/Image_Datasets/Caltech256

[27] L.M.A. Bettencourt, et al., Image categorization through hierarchical models of the primate visual system, Annual Meeting Society for Neuroscience, 2009.

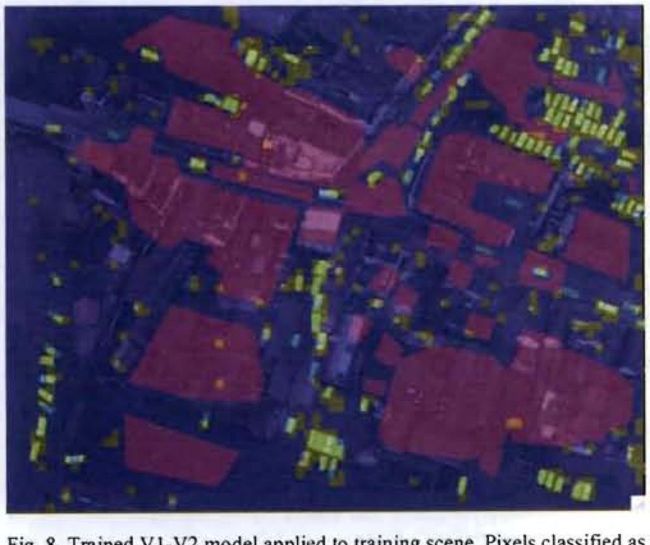

Fig. 8. Trained V1-V2 model applied to training scene. Pixels classified as containing a vehicle are marked yellow (true positives) and orange (false positives). False negatives are marked in cyan. True negatives are purple.

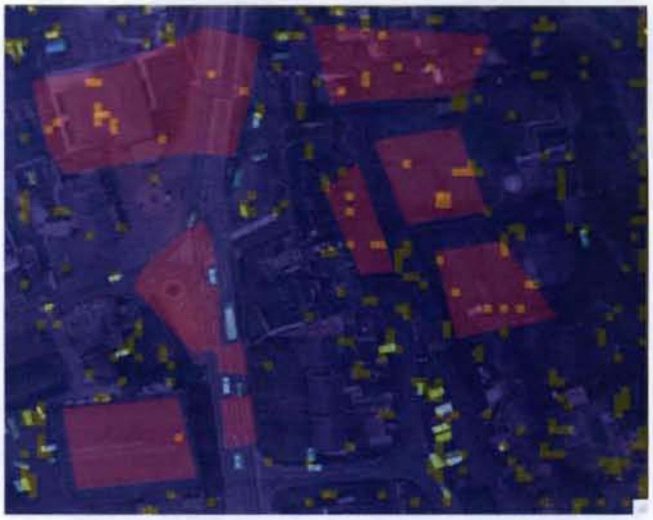

Fig. 9. Trained V1-V2 model applied to test scene (same color scheme).

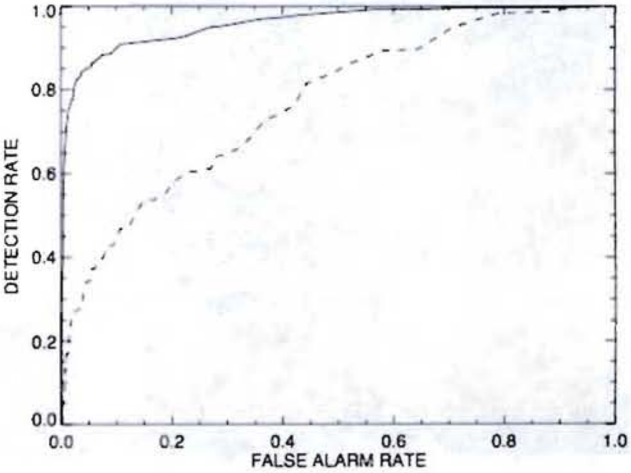

Fig. 10. Receiver Operating Characteristic (ROC) curve for training scene (solid line) and test scene (dashed line).