A Framework for Machine Vision based on Neuro-Mimetic Front End Processing and Clustering

Emre Akbas[†], Aseem Wadhwa[⋆], Miguel Eckstein[†] and Upamanyu Madhow[⋆]
[†]Department of Psychology and Brain Sciences, University of California Santa Barbara, CA 93106, USA,
{akbas, eckstein}@psych.ucsb.edu
[⋆]Department of ECE, University of California Santa Barbara, CA 93106, USA, {aseem, madhow}@ece.ucsb.edu

***Abstract*—Convolutional deep neural nets have emerged as a highly effective approach for machine vision, but there are a number of open issues regarding training (e.g., a large number of model parameters to be learned, and a number of manually tuned algorithm parameters) and interpretation (e.g., geometric interpretations of neurons at various levels of the hierarchy). In this paper, our goal is to explore alternative convolutional architectures which are easier to interpret and simpler to implement. In particular, we investigate a framework that combines a front end based on the known neuroscientific findings about the visual pathway, together with unsupervised feature extraction based on clustering. Supervised classification, using a generic radial basis function (RBF) support vector machine (SVM), is applied at the end. We obtain competitive classification results on standard image databases, beating the state of the art for NORB (uniform-normalized) and approaching it for MNIST.**

I. Introduction

Neuro-inspiration has played a key role in machine learning over the years. In particular, the recent impressive advances in machine vision are based on multilayer (or "deep") convolutional nets [1], [2], [3], [4], which loosely mimic the natural hierarchy of visual processing. Neuro-inspired operations such as local contrast normalization [5], [6], rectification [7] and sparse autoencoding [8] have been found to be central to improving performance [6]. Most of the best performing nets today are trained in supervised fashion [3], [4], [9]. Despite the state of the art classification accuracy achieved by this approach, there are a number of disconcerting features: a huge number of parameters to be trained, which leads to long training times [3] and the requirement of large amounts of labeled data [10]; lack of a systematic framework for understanding commonly used "tricks" such as DropOut/DropConnect [9]; the requirement for manual tuning of parameters such as learning rate, weight decay and momentum [3]; and the difficulty in interpreting the information being extracted at various hidden layers of the network [11].

This research was supported in part by the Institute for Collaborative Biotechnologies through the grant W911NF-09-0001 from the U.S. Army Research Office, and in part by the Systems on Nanoscale Information fabriCs (SONIC), one of six centers supported by the STARnet phase of the Focus Center Research Program (FCRP), a Semiconductor Research Corporation program sponsored by MARCO and DARPA. The content of the information does not necessarily reflect the position or the policy of the Government, and no official endorsement should be inferred.

In this paper, we ask whether we can simplify both implementation and understanding of convolutional architectures, based on combining several key observations. First, while we have at best a coarse understanding of the higher layers of the visual cortex, we should be able to leverage the fairly detailed picture available for the *front end* of the visual system, including retinal ganglion cells (RGCs) and the lateral geniculate nucleus (LGN), along with the simple cells in V1. Thus, it should be possible to engineer machine learning front ends to be *faithfully neuro-mimetic rather than merely neuro-inspired.* Second, we would like to build on the intuition that our visual system extracts a set of "universal" features for any object being viewed, irrespective of whether a classification task is to be performed. Research in the field of *transfer learning* [12], where parameters of a neural net trained with a dataset have been found to work reasonably well with other datasets, seems to support this assumption. This implies that a system which focuses most of its effort on unsupervised learning for feature extraction, and takes on supervised classification at the end, should have a reasonable chance of success. Indeed, such an approach has been shown to work reasonably well by a few researchers, but further effort is needed to provide classification performance competitive with supervised nets tuned for the purpose of classification. Third, if we shift the focus to unsupervised learning, then the task becomes one of clustering, for which there are simple, well-established algorithms with little need for parameter tuning.

Based on the preceding concepts, we propose and evaluate a convolutional architecture that attains classification performance comparable to the state of the art (beating the state of the art for the NORB image database, and coming close to it for the MNIST handwritten digit database), while lending itself to relatively straightforward interpretation. Our design approach and contributions are summarized as follows:
1) As the first part of our neuro-mimetic front end, we build retinal ganglion cells (RGCs) with center-surround characteristics, with center-on cells responding when the center is brighter than the surround, and the center-off cells responding in the reverse situation. The number of such cells and the receptive cell size are matched to the resolution of the images being processed based on the known parameters of the fovea, the center of the retinal field with the greatest

concentration of RGCs. The RGC outputs can be viewed as being directly transported to the lateral geniculate nucleus (LGN), with a one-to-one mapping between RGCs and LGN neurons. Thus, we may view this part of the model as applying to the cascade of the RGC and LGN. We perform local contrast normalization on the RGC/LGN outputs, with the neighborhood used determined by reported experimental parameters. We then rectify these outputs before feeding them to the next layer.

2) Our second front end stage is a model for V1 simple cells layered on top of RGC/LGN. These are edge detectors constructed using the rough parameters determined by the classical experiments of Hubel and Wiesel [13], [14]. We quantize the edge orientations into bins of width $\pi/8$ (the actual binning in visual cortex may be finer-grained, but we choose a relatively coarse bin size to limit complexity). We use several different kinds of edge detectors, so that there are 48 edge detectors centered at each spatial location. We perform local contrast normalization and rectification on the simple cell outputs. The front end is fixed, with the only tunable parameter being the "viewing distance," as discussed in Section II-C.

3) Beyond simple cells, neuroscientific guidance sufficient for constructing a complete model of the next layer is no longer available. We therefore use clustering based on k-means for unsupervised learning henceforth. We first use k-means clustering of outputs from simple cells to obtain centroids (each of which can be interpreted as a neuron). Feature vectors are given by soft assignments to these centroids (which can be viewed as thresholded neuron outputs), and feature vectors from adjacent regions are pooled to obtain the final feature vector. A similar procedure (k-means, soft assignments, and pooling) can be used to build successive layers on top of this. Note that the structure remains convolutional (the same set of centroids slides across the image), but we are zooming out (creating feature vectors for larger segments of the image) as we go up in the hierarchy.

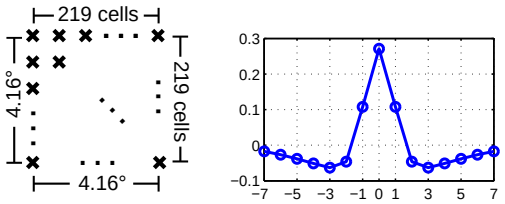
4) After the fixed front end and the unsupervised learning we finally perform classification via supervised learning of a standard support vector machine (SVM) [15] with a radial basis function (RBF) kernel. The best error rates we achieve are: 0.66% on MNIST [1], which is comparable to the best rates reported on this dataset without data augmentation and 2.52% on NORB (uniform-normalized [16]), which improves on the state of the art for this dataset.

Related work: The relevant papers in experimental and computational neuroscience which our front end model is based on are mentioned in Section II. The importance of carefully designing the pre-processing layer has been noted in the machine learning literature. It was shown in [17] that optimizing the various parameters of a single layer convolutional architecture, followed by simple non-linear clustering using k-means, results in performance even better than several deep architectures. In [4], it was found that adding a pre-processing *contrast-extraction* layer to the deep CNN architecture improves recognition performance with the NORB dataset.

There are several references [18], [19], [20], [21], [22] that have employed layers of unsupervised feature extraction prior to supervised classification, an approach adopted in this work as well. Most of these papers use some form of reconstruction error combined with a sparsity constraint as the cost function for training the unsupervised layers. This differs from our use of k-means clustering to learn the weights of the unsupervised layers, an approach which is much simpler to implement computationally. A few references that have used k-means clustering for vision include [17], [23]. In these papers the clustering step is used directly on the raw images and their implementation of k-means differs significantly from ours, especially for the higher layers. We use much fewer number of centroids and get better error performance on the dataset common amongst their work and ours (NORB, [17]).

II. The Front End Model

Our model consists of two layers of neurons, the first corresponding to the RGC/LGN cascade, and the second to V1 simple cells, along the primate visual pathway. We model the fovea, the small part of the visual field around the center of gaze where the visual acuity is highest [24]. The fovea is responsible for tasks that require high-resolution spatial detail such as reading. The diameter of the fovea is reported to be between 4.17° and 5.21° [25], [24]. The average of these estimates is 4.69°, and we model our "digital fovea" as a 4.16°-by-4.16° square patch having the same area as a disk with 4.69° diameter.



(a) (b)

Fig. 1. **(a)** Cross marks show cell centers which are arranged on the vertices of a regular grid. In each row (or column) there are 219 RGCs. Each RGC cell applies a difference-of-Gaussian (DoG) filter, which defines the receptive field of the cell. Receptive fields of neighboring cells heavily overlap. **(b)** Difference of Gaussian filter along a single dimension. X-axis indices correspond to number of RGC cells.

A. RGC/LGN processing

The number of RGCs in the fovea is estimated around $120,000$ [26], [27]. Among many types of RGCs [28], midget RGCs (sustained response cells or P-cells) carry the high-acuity information [25] and comprise 80% of all the RGCs in the retina [29]. About half of these cells are ON-center-OFF-surround and the other half are OFF-center-ON-surround [24]. Based on this evidence, we create two parallel visual

pathways, one for ON-center cells and the other for OFF-center cells. Each pathway contains approximately 48000 cells. The cell centers are located on the vertices of a square regular grid (Fig. 1(a)). The front end also includes two mechanisms that are critical for operation over the wide dynamic range exhibited by natural stimuli: local luminance gain control (LGC) and contrast gain control (CGC) [30], [5].

We first apply LGC as described by Carandini and Heeger [5]. Denoting by $\boldsymbol{x}$ the input image, the luminance normalized image $\boldsymbol{c}$ is given as

$$\boldsymbol{c}_{i,j} = \frac{\boldsymbol{x}_{i,j} - \overline{\boldsymbol{x}_{i,j}}}{\overline{\boldsymbol{x}_{i,j}}} \tag{1}$$

where i, j denote a pixel and $\overline{\boldsymbol{x}_{i,j}}$ is a weighted average around pixel i, j,

$$\overline{\boldsymbol{x}_{i,j}} = \sum_p \sum_q \boldsymbol{w}_{p,q} \boldsymbol{x}_{i-p,j-q}. \tag{2}$$

where the weights $\boldsymbol{w}$ are given by the Gaussian surround filter suggested in [31], normalized to sum to 1.

Computation of center-surround contrast is classically modeled using the difference-of-Gaussian (DoG) model [32], [33], [34] consisting of two components, center and surround, each of which is a 2D Gaussian function. We set the parameters of the center and surround Gaussian filters based on the values given for the macaque retina [34] (details in the appendix). Taking the difference between these gives a DoG filter (Fig. 1(b)) whose radius covers about 7 cell centers along a row. Convolving the luminance-normalized image with the DoG filter, the ON-center cell responses are governed by the positive part of the output, and the OFF-center by the negative part (Fig. 2). We apply CGC as follows. The output (spike rate) of a cell whose center is at i, j set to [5] is given by

$$\boldsymbol{r}_{i,j} = \frac{\sum_p \sum_q \boldsymbol{v}_{p,q} \boldsymbol{c}_{i-p,j-q}}{\beta + \sqrt{\sum_p \sum_q \boldsymbol{w}_{p,q} \boldsymbol{c}^2_{i-p,j-q}}} \tag{3}$$

where $\boldsymbol{v}$ are the difference-of-Gaussian weights. The square-root term in the denominator, called the local contrast, is the weighted root mean square of the luminance normalized intensity values within the whole receptive field. The area defined by $\boldsymbol{w}$ is called the suppressive field. The parameter β has been fit to neural data by Bonin *et al* [31], but this value is for cells outside of the fovea, and hence is not directly usable for our model. We therefore choose a value of β $(= 0.1)$ so that the cells in our model qualitatively match various effects (step change in luminance, step change in contrast, size and contrast tuning) described by Bonin et al. [31] (we skip details due to lack of space).

Finally, the non-negative spike rate of the cell is obtained via a rectification non-linearity [30]:

$$\boldsymbol{y}^{ON}_{i,j} = \max(0, \boldsymbol{r}_{i,j} - T_{RGC}) \tag{4}$$

$$\boldsymbol{y}^{OFF}_{i,j} = \max(0, -\boldsymbol{r}_{i,j} - T_{RGC}) \tag{5}$$

where T_{RGC} is the rectification threshold: we set $T_{RGC} = 0$, which corresponds to simply splitting responses into positive and negative components. Such "polarity splitting" has been used in several machine learning algorithms (e.g., [21]), and preserves more information than absolute value rectification. The overall flow of RGC processing is illustrated for a single cell in Fig. 2.

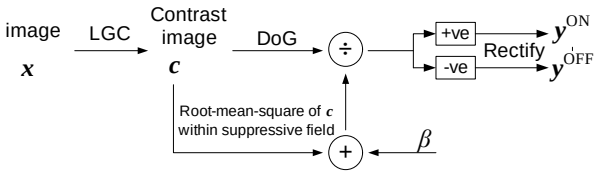


Fig. 2. RGC processing pipeline for a single RGC cell

While both luminance and contrast gain control are thought to start at the retina, lateral geniculate nucleus (LGN) cells strengthen CGC [30]. For this reason, we refer to this layer as the RGC/LGN layer.

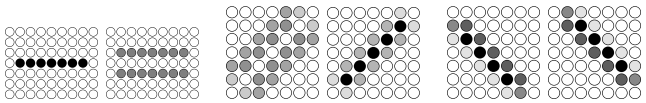


Fig. 3. A simple cell sums the output of RGC/LGN cells according to its incoming weights, these are represented here in terms of the colors of the circles. The darker the color of a cell, the more weight it has. Transparent cells have zero weight. Weights of each simple cell are normalized to sum to 1. For each simple cell, the weight connections to the midget-ON and OFF RGCs are shown on the left and right sides respectively. (a) orientation $0°$, OFF-ON-OFF type connection to midget ON. (b) orientation $45°$, ON-OFF-ON type connection to midget ON. (c) orientation $135°$, ON-OFF type connection to midget ON.

B. V1 simple cells

The V1 layer consists of two populations of neurons: simple cells and complex cells. While there is a strong consensus on the computation performed by V1 simple cells – they extract oriented edges – the picture is less clear about the complex and hypercomplex cells. Hubel and Wiesel [13] suggest that some complex cells are implementing an OR-like (or MAX-like) operation, while there are recent studies [35], [36] which suggest significant computational diversity among complex cells. We therefore only include simple cells in our front end model.

Simple cells have incoming connections from the RGC/LGN layer. We create simple cell receptive fields based on the size ($0.25°$x$0.25°$ [14]) and the shapes ([13, Fig. 2]) reported by Hubel and Wiesel for foveal simple cells. While this seminal work that we draw upon is almost five decades old, there are only a few other studies [37], [38] of primate foveal V1 cells, and the detail they present are insufficient to implement a complete simple cell population. Other models for parafoveal neurons ($5° - 6°$ degrees off-center) [39], [40] are similar in concept, but different in size, from the Hubel/Wiesel foveal model.

There are a total of 48 different types of simple cells in our model. There are 8 orientations, starting at $0°$ (horizontal

edge) and increasing in increments of 22.5°. For each orientation, there are 6 kinds of simple cells: two ON-OFF-ON, two OFF-ON-OFF and one each of the type ON-OFF and OFF-ON. To understand the differences between these types we illustrate three different simple cells in Figure 3. Each simple cell is connected to both midget-ON and midget-OFF RGCs (and thus obtains information from both the positive and negative parts of the DoG outputs), and its shape is characterized by the set of nonzero weights. Each simple cell has a receptive field size of 7x7 RGC cells, but depending on its shape and type (equivalently, the set of nonzero weights), the number of incoming connections vary from 14 to 39 RGC/LGN cells. The unnormalized output of the simple cell at location (i, j) with orientation θ and shape γ is the sum of its afferent inputs:

$$\boldsymbol{s}^{(raw)}_{i,j,\theta,\gamma} = \sum_{p,q} \boldsymbol{\ell}^{ON}_{p,q} \boldsymbol{y}^{ON}_{i-p,j-q} + \sum_{p,q} \boldsymbol{\ell}^{OFF}_{p,q} \boldsymbol{y}^{OFF}_{i-p,j-q} \quad (6)$$

where $\boldsymbol{\ell}$ are the weights (e.g. as shown in Fig. 3) of the incoming RGC/LGN cells. The superscripts ON and OFF refer to the midget-ON and midget-OFF pathways. Similar to the contrast gain control occurring at the previous layer, cortical neurons are also locally normalized [30]. Carandini and Heeger [5] propose several variations of the normalization model. (Normalization has also been successfully used in bio-inspired methods [6], [41], [42].) In our experiments, we use a normalization similar to (3) used at the RGC/LGN layer: local demeaning, followed by a divisive normalization with root-mean-square of nearby outputs, a measure of local contrast.

$$\boldsymbol{s}^{(norm)}_{i,j,\theta,\gamma} = \frac{\boldsymbol{s}^{(raw)}_{i,j,\theta,\gamma} - \overline{\boldsymbol{s}^{(raw)}_{i,j,\theta,\gamma}}}{\max\left(\epsilon, \sqrt{\sum_{p,q,\theta,\gamma} \boldsymbol{w}_{p,q} \left(\boldsymbol{s}^{(raw)}_{i-p,j-q,\theta,\gamma} - \overline{\boldsymbol{s}^{(raw)}_{i,j,\theta,\gamma}}\right)^2}\right)} \quad (7)$$

where the summation is taken over the suppressive field $\boldsymbol{w}$ across orientations and shapes, $\overline{\boldsymbol{s}^{(raw)}_{i,j,\theta,\gamma}}$ is a weighted local average (using $\boldsymbol{w}$ as weights) of unnormalized V1 outputs for θ, γ around i, j, and ϵ is a small positive constant to prevent division by zero (we set it to 0.001). Finally, the normalized simple cell output is rectified to yield a non-negative spike rate

$$\boldsymbol{s}_{i,j,\theta,\gamma} = \max(0, \boldsymbol{s}^{(norm)}_{i,j,\theta,\gamma}). \quad (8)$$

C. Viewing distance and foveal image resolution

Our model has a 4.16°x4.16° visual field. For a typical viewing distance of 50 cm, this field corresponds to a 3.6x3.6 cm^2 patch. The smaller the viewing distance, the smaller the image patch covered by the fovea, and vice versa.

In order to implement our model digitally, one has to assume a size for the foveal image. One possibility is to assume that the resolution is limited by the number of photoreceptor cells. In the fovea, there are almost exclusively cone photoreceptors. Based on the cone density at the fovea [25], there are about $3 \cdot 10^5$ cells which would mean a 550x550 pixel resolution. Considering the typical viewing distance example given above, 3.6 cm would correspond to 550 pixels resulting in a 152.8 pixels/cm density which is too high compared to pixel densities of available displays ($\approx 40 - 100$ pixels/cm). To close this gap, one either has to increase the resolution of the input image or scale down the foveal image size. We choose the latter for simplicity and assume that the foveal image resolution is equivalent to the RGC resolution, i.e. 219x219 (61 pixel/cm). That is, at every pixel there is a RGC cell center. With these settings, the radius of the center component for a midget-ON cell is 1.27 pixels and the radius of the surround component is 5.53 pixels. An image from the MNIST dataset [1], which is 28x28 pixels, would be seen by 28x28 midget-ON RGC cells (and by the same number of midget-OFF cells); and would cover approximately 0.5x0.5 cm^2 area on a display with 60 pixel/cm viewed at 50 cm distance. An image from the NORB dataset [16] (96x96 pixels) would cover 1.6x1.6 cm^2.

While one RGC center per pixel is a sensible design choice, it is possible to tune the viewing distance parameter in our model. For example, larger values would increase the number of RGC centers per pixel, and require sub-pixel computations. We do not experiment with the viewing distance parameter in this paper, but note that it could be of interest, for example, when comparing the performance of our model with human performance on the same task in psychophysics experiments.

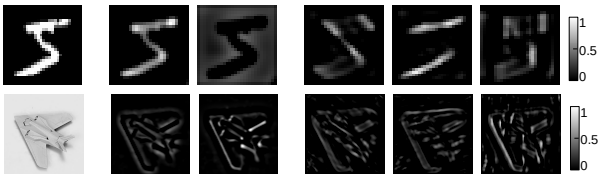


Fig. 4. Sample RGC and V1 output. First row is for an image from MNIST, the second row is for NORB. The first column has the original images. The second and third columns are midget-ON and midget-OFF outputs. The last three columns are outputs of 4 simple cells at different orientations. The midget-ON and OFF responses seem to light up the *relevant* regions containing activity.

III. Higher Layer Processing

Our front end implements 48 types of simple cells centered around each input pixel, so that our front end outputs, for each pixel, an f-dimensional feature ($f = 48$ for monocular images as in MNIST, and $f = 96$ for NORB, which consists of a set of binocular images). We employ k-means clustering on this f-dimensional data, as a natural proxy for complex cell modeling. Thus, the feature map for an $N \times N$ image at our front end output is $N \times N \times f$, while that after the first layer of clustering is $N \times N \times k$ (to be cut down by pooling). We consider two implementations: a single layer of clustering followed by pooling and supervised classification, or two layers of clustering (and pooling), and then using a

concatenation of layers 1 and 2 features for classification. The second implementation is consistent with visual models for higher layers, which predict connections from both layers V1 and V2 into V3.

A. Layer 1 of clustering

We have denoted by $\boldsymbol{s}_{i,j}$ the activations of simple cells centered at a particular spatial location i, j. To represent a response in general, we drop spatial coordinates from the notation and denote the activations by $\boldsymbol{a} = \boldsymbol{s}_{i,j}$, an f-dimensional vector. We implement spherical k-means clustering [43] using an *inner product* similarity metric $\boldsymbol{a}^T\boldsymbol{c}$, where $\boldsymbol{c}$ denotes a cluster center. This is equivalent to clustering using a standard Euclidean distance metric with a unit norm constraint on the cluster centers. In our implementation, we use the online clustering algorithm in [43], which has the advantage of being less sensitive to initialization. We speed up the algorithm by using mini-batches instead of iterating over single data points.

Note that computation of the inner product of a data vector with a cluster center is identical to weighted summations in classical neural networks, hence we may interpret each cluster center as a neuron. The subsequent nonlinearity, however, is different from the sigmoidal nonlinearity in standard neural networks. As described shortly, we use soft assignments, which may be interpreted in terms of local competition between the neurons.

In addition to using the *standard* inner product as a similarity metric, we also consider a modified version that takes into account the correlations in simple cell activations. Given the weights connecting LGNs to the simple cells, represented by $L = [\boldsymbol{\ell}_1, .., \boldsymbol{\ell}_{48}]$, we compute the 48×48 correlation matrix as $C_l = L^T L$ and use the metric $\boldsymbol{a}^T C_l^{-1} \boldsymbol{c}$ or $(C_l^{-\frac{1}{2}}\boldsymbol{a})^T(C_l^{-\frac{1}{2}}\boldsymbol{c})$ for k-means. This can be viewed as doing *whitening* before clustering. For NORB, where $f = 96$ and simple cell outputs are concatenation of the left and right channels, we do not have prior information about the correlations among the two channels, and model them as independent.

Given the centroids, the soft activations are evaluated by $f\left([\boldsymbol{a}^T C^{-1}\boldsymbol{c_1}, ..., \boldsymbol{a}^T C^{-1}\boldsymbol{c_{K_1}}]^T\right)$, where $C = C_l$ or $C = I$ and K_1 are the number of layer 1 cluster centers learned. We use the *soft threshold* as the encoding function, i.e. $f(x) := \max(0, x - T)$. It is known that neurons fire only when active above a certain threshold hence rectification for the non-linearity is a natural choice. For choosing the value of T we take the simple approach of setting it to maintain a certain level of *sparsity* on average. For instance, we can choose T for 80% sparsity (i.e., only 20% of the neurons have non-zero activations on average). This design rule gives us a direct and intuitive handle on controlling the level of sparsity, as opposed to the regularization parameter generally used in cost functions containing a sparsity term [22], [18], [19]. The resulting design conforms to the intuition that neural activity on average is expected to be low. The final activation vector generated is of length $K_1 + 1$: the last coordinate is set to a non-zero value when all the K_1 responses corresponding to the centroids turn out to be zero after thresholding. This typically corresponds to patches with no or negligible activity.

Features extracted by layer 1, as expected, correspond to different kinds of edges, blobs etc. In order to visualize a centroid, we backproject its receptive fields to the raw image level and plot the patches closest to it. Since layer 1 centroids are connected directly to the simple cell responses, their receptive field size is same as that of the simple cells: 7×7 RGCs or pixels in the image domain. In Figure 5, for the MNIST dataset, we show visualizations for four centroids.

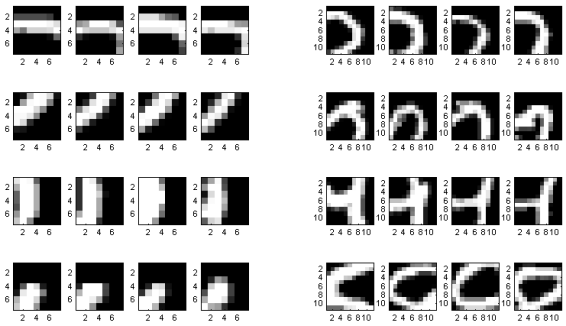


Fig. 5. Left side: layer 1 centroids. Right side: layer 2 centroids. Each row plots patches closest to that centroid.

B. Layer 2 of clustering

The idea with the second layer of clustering is to extract more complex features: combination of simple edges like corners, L-junctions etc. The expansion of receptive field size or zooming out is achieved via local spatial pooling and concatenation. Max-pooling over a small neighborhood also results in local translation invariance. Pooling is generally followed by subsampling, hence it results in reducing the resolution of the feature maps. Denoting the max-pooled activations at the spatial location i, j by $\boldsymbol{b}_{i,j}$, these are then concatenated over a 2×2 neighborhood to generate $4(K_1+1)$-dimensional input for the second layer of clustering, given by $[\boldsymbol{b}_{i,j}; \boldsymbol{b}_{i,j+1}; \boldsymbol{b}_{i+1,j}, \boldsymbol{b}_{i+1,j+1}]$. These activations now correspond to larger patches of the raw image. Clustering is performed using the similarity metric:

$$\sum_{ii=0}^{1}\sum_{jj=0}^{1} \frac{\boldsymbol{b}_{i+ii,j+jj}^T \boldsymbol{w}_{ii,jj}}{\|\boldsymbol{b}_{i+ii,j+jj}\| \, \|\boldsymbol{w}_{ii,jj}\|} \tag{9}$$

where a second layer centroid is represented by $\boldsymbol{c}^{(2)} = [\boldsymbol{w}_{0,0}; \boldsymbol{w}_{0,1}; \boldsymbol{w}_{1,0}; \boldsymbol{w}_{1,1}]$. Using this metric can be interpreted as individually comparing the four quadrants of the larger patch and computing an averaged matching score. This is expected to group together shapes with similar arrangement of edges, with the metric interpreted as *stitching* the edges

	Sparsity level= 80%			Sparsity level= 95%		
	(Layer 1) (K_1=200)	(Layer 1) (K_1=600)	(Layer 1+2) (K_1=200) (K_2=600)	(Layer 1) (K_1=200)	(Layer 1) (K_1=600)	(Layer 1+2) (K_1=200) (K_2=600)
MNIST	0.73	0.72	**0.66**	0.78	0.78	0.68
NORB	3.96	3.71	2.94	2.58	**2.52**	2.90

TABLE I
MNIST AND NORB RESULTS: ERROR RATE (%) ON THE TEST SET.

together. The soft assignment encoding function is as in layer 1.

In order to understand how pooling, subsampling and concatenation enlarges the receptive field size, consider a simple 1D example. Suppose that layer 1 centroids/neurons have a receptive field of size 7 (i.e. a neuron at location i in layer 1 gets its inputs from layer 0 neurons indexed at $[i-3, i+3]$). Now suppose we do pooling and subsampling, both by a factor of 2. For pooling by a factor of 2, layer 1 neurons at i and $i+1$ are pooled together to generate a layer 2 neuron, so that the effective receptive field (with respect to layer 0) for this new neuron is 8: $[i-3, i+4]$. Since we subsample by a factor of 2, the neighbor of this new neuron is based on pooling layer 1 neurons at $i+2$ and $i+3$. Now, when these two neighboring layer 2 neurons are concatenated, their resulting receptive field size is 10 in terms of layer 0: $[i-3, i+4]+[i-1, i+6] = [i-3, i+6]$.

In our experiments with MNIST, after layer 1 clustering, we perform 2×2 pooling, subsampling by 3 and 2×2 concatenation, followed by layer 2 clustering: hence layer 2 centroids correspond to 11×11 sized raw image patches. Figure 5 shows visualizations of a few layer 2 centroids using these 11×11 patches.

IV. EXPERIMENTS

In this section, we evaluate our model on two standard image classification benchmarks, MNIST [1] and NORB [16]. The only free parameter for the neuro-mimetic front end is the viewing distance which we set to 50cm. For the higher layers we experiment with number of centers $K_1 = 200$ or 600 for layer 1, and $K_1 = 200$ and $K_2 = 600$ when employing both layers 1 and 2. Thresholds are chosen to keep the sparsity level at either 80% or 95% for both layers. We use non-linear SVM with the radial basis function (RBF) kernel [44] for supervised classification. RBF SVM has two parameters: the cost parameter, which we fix to 100 as that seemed to be a robust choice in our experiments, and the scale parameter for the kernel, γ, which is set via a grid search using cross-validation on a subset of the training set. Several references have used data augmentation (via affine distortions) to enlarge the training set in order to boost classification performance, but we do not employ it here.

MNIST: MNIST consists of 28×28 images of handwritten digits. The dataset contains 60K training and 10K testing images. The front end produces feature maps of size $28 \times 28 \times (K_1{+}1)$. If only layer 1 is used for classification, spatial average pooling over a 4×4 grid followed by concatenation provides a 1D vector of dimension $4^2 \times (K_1 + 1)$ to be fed into the RBF SVM. When layer 2 is also used, we fix $K_1 = 200$ and max-pool layer 1 activations over a 2×2 local neighborhood. This is subsampled by a factor of 3, and edges are cropped, giving feature maps of size $8 \times 8 \times 201$. We then concatenate neighboring responses over a 2×2 grid, which leads to a feature map size $7 \times 7 \times 804$. The 804-length feature vectors are clustered in layer 2 using $K_2 = 600$ centroids, producing feature maps of size $7 \times 7 \times 601$. Finally, layer 2 features for classification are generated by pooling over a 3×3 grid, coarser than layer 1 since the activations now correspond to larger image patches (11×11, layer 1 centroids represent 7×7). Concatenating layer 1 and 2 features results in a total of $4^2 \cdot 201 + 3^2 \cdot 601 = 8625$ features per image, which is comparable to the length of layer 1 features alone with $K_1 = 600$ (9616). For MNIST, we find that using *whitening* prior to layer 1 clustering, as discussed in section III-A, yields better results, hence we only report those error rates (Table I). We see that the best error rate 0.66% is achieved using both layer 1 and 2 features and a sparsity level of 80%. Increasing the sparsity appears to degrade the performance, especially when using just layer 1. The state of the art on MNIST (without distortions) is 0.39% [45], which is achieved using a purely supervised net. Although the error rate we get is higher than that, it is comparable to the rates reported by several other references, 0.64% [18], 0.82% [20], 0.59% [19], that use a combination of unsupervised and supervised learning.

NORB: We use the normalized-uniform variant [16] of the NORB dataset. Each of the training and test sets have 24300 binocular images of 5 classes of toys placed on a uniform background. Each monocular image is 96×96. We pre-process the images by cropping 8 pixels from all sides reducing the image size to 80×80, in order to speed up the processing of the dataset. This cropping discards some of the uniform background and it does not affect the final performance. The operations are mostly identical to those for MNIST, hence we only mention the differences here. Due to the larger image sizes, the final spatial pooling before classification is done over a finer grid: 5×5 for layer 1 and 4×4 for layer 2. Another difference is that max pooling is performed over 3×3 neighborhoods after layer 1 clustering, the layer 2 centroids represent 12×12 patches. As with MNIST, the size of concatenated layer 1 and 2 features is comparable to layer 1 features with $K_1 = 600$ centers. For NORB, unlike MNIST, omitting whitening at layer 1 clustering results in better performance. We believe this could be due to the inability of the correlation matrix (C_l) to model correlations between the left and right channels. The current best result on the normalized-uniform NORB, to the best of our knowledge, is the one reported in [4] and is 2.87% without data augmentation and 2.53% with translation distortions. The best result obtained by us of 2.52% thus improves upon the state of the art; it is even marginally better

than the previous best with distortions, even though we do not employ distortions.

Discussion: While these classification results are encouraging, there are several unanswered questions. Design choices such as whitening and sparsity level appear to be dataset dependent for optimizing the classification performance. It might be the case that the optimal sparsity levels depend on the noisiness of the dataset or hierarchy of the layer. The impact of whitening before clustering is also not clear. In [17], whitening using the empirical covariance matrix has been found to improve performance, but it did not improve our results. We generally expect higher layer features to improve recognition performance, but in the NORB experiments with 95% sparsity, we were surprised to find performance degrading with the inclusion of layer 2 features. Clearly, our understanding of how best to combine information generated from different layers is far from complete. While our focus has been on feature design via clustering, it is important to explore multiple options for the supervised classification layered on top of it (e.g., comparing multilayer neural nets to the nonlinear SVM used here).

V. Conclusion

We have shown that an architecture based on neuromimetic front end processing and clustering offers a promising approach for "universal" feature extraction for machine vision. Layering a generic (but powerful) supervised classifier on top is shown to provide performance close to, or exceeding, the state of the art for two well known image databases. Key advantages of our approach are its simplicity, the small number of tunable parameters, and the ability to easily interpret the features being extracted at each layer.

We view this work as a first step towards bridging the gap between computational neuroscience and machine learning: machine vision algorithms are often neuro-inspired but rarely implement computations that strictly follow neuro-scientific findings, while psychophysical models that try to follow physiological visual processing more closely are typically applied to restricted problems with artificial inputs[46], [47]. The results in this paper show that leveraging neuro-scientific findings more carefully can pay off in terms of machine vision performance.

An obvious disadvantage of our approach, from the point of view of machine learning, is that we are limited in our front end design by the state of knowledge in neuroscience, instead of learning purely from data. For example, our model here is restricted here to grayscale images, because more work is needed to put together the available experimental evidence regarding color processing at the RGC/LGN layers, which exhibits features such as red-green and blue-yellow opponency [28]. However, we believe that this additional effort in faithful modeling is well worth it because of the potential benefits from leveraging evolution. In particular, we would like to extend our approach (both in terms of neuromimetic front end and layered clustering) to other kinds of data, such as audio and video.

A fundamental challenge, as we aim to build additional layers using clustering, is to develop a quantitative understanding of whether all of the relevant information is being captured by our feature extractor. The only available metric at present to evaluate the efficacy of our architecture is classification performance after inserting a supervised layer, which is sensitive to the dataset and perhaps to the complexity of the supervised layer. An important open question, therefore, is if there are alternative metrics for evaluating the quality of information being extracted by unsupervised learning models such as ours. Of course, in parallel with this line of inquiry, we would like to continue optimizing our architecture so that it meets or surpasses classification performance on standard databases.

Appendix A
Difference of Gaussian parameters

We use the classical difference-of-Gaussians (DoG) model ([32], [33], [34]):

$$R(x,y) = K_c e^{-\frac{(x^2+y^2)}{r_c^2}} - K_s e^{-\frac{(x^2+y^2)}{r_s^2}} \quad (10)$$

where K_c and r_c are the contrast gain and radius of the center component, respectively, and K_s, r_s are the same for the surround component. DoG parameter values for the foveal RGCs are not directly available in published data. Croner and Kaplan [34] report

- median values of $r_c = 0.03^\circ$ and $r_s = 0.18^\circ$, for cells at $0^\circ - 5^\circ$ eccentricity[1]; and
- median values of $r_c = 0.05^\circ$ and $r_s = 0.43^\circ$ for cells at $5^\circ - 10^\circ$ eccentricity.

r_c, r_s increase linearly with eccentricity [34]. Hence, we fit a line to the values above (e.g. for r_c, two points on the line are $(2.5^\circ, 0.03)$ and $(7.5^\circ, 0.05)$ where we took 2.5° as the representative eccentricity for the $0^\circ - 5^\circ$ interval, and 7.5° for the $5^\circ - 10^\circ$). We choose 1° as the representative eccentricity for foveal RGCs, where the lines yield $r_c = 0.024^\circ$ and $r_s = 0.105^\circ$. The degree/pixel ratio for our model is $4.16^\circ/219$ pixels $= 0.019$ degree/pixel. Therefore, $r_c = 0.024/0.019 = 1.27$ pixels and $r_s = 0.105/0.019 = 5.53$ pixels. The values of K_c and K_s are inversely proportional to the center and surround areas, respectively [34].

References

[1] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[2] P. Simard, D. Steinkraus, and J. C. Platt, "Best practices for convolutional neural networks applied to visual document analysis," in *ICDAR*, vol. 3, pp. 958–962, 2003.

[1]The eccentricity of a point A on the retina is the angle between the center of the fovea and A.

[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks.," in *NIPS*, vol. 1, p. 4, 2012.

[4] D. C. Cireşan, U. Meier, J. Masci, L. M. Gambardella, and J. Schmidhuber, "High-performance neural networks for visual object classification," *arXiv preprint arXiv:1102.0183*, 2011.

[5] M. Carandini and D. J. Heeger, "Normalization as a canonical neural computation.," *Nature reviews. Neuroscience*, vol. 13, pp. 51–62, Jan. 2012.

[6] K. Jarrett, K. Kavukcuoglu, M. A. Ranzato, and Y. LeCun, "What is the best multi-stage architecture for object recognition?," in *International Conference on Computer Vision (ICCV)*, pp. 2146–2153, 2009.

[7] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pp. 807–814, 2010.

[8] B. A. Olshausen and D. J. Field, "Sparse coding with an overcomplete basis set: a strategy employed by V1?," *Vision research*, vol. 37, pp. 3311–25, Dec. 1997.

[9] L. Wan, M. Zeiler, S. Zhang, Y. L. Cun, and R. Fergus, "Regularization of neural networks using dropconnect," in *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, pp. 1058–1066, 2013.

[10] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *arXiv preprint arXiv:1207.0580*, 2012.

[11] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Computer Vision–ECCV 2014*, pp. 818–833, Springer, 2014.

[12] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell, "Decaf: A deep convolutional activation feature for generic visual recognition," *arXiv preprint arXiv:1310.1531*, 2013.

[13] D. H. Hubel and T. N. Wiesel, "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex," *The Journal of physiology*, pp. 106–154, 1962.

[14] D. H. Hubel and T. N. Wiesel, "Receptive fields and functional architecture of monkey striate cortex," *The Journal of physiology*, pp. 215–243, 1968.

[15] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[16] Y. LeCun, F. J. Huang, and L. Bottou, "Learning methods for generic object recognition with invariance to pose and lighting," in *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, vol. 2, pp. II–97, IEEE, 2004.

[17] A. Coates, A. Y. Ng, and H. Lee, "An analysis of single-layer networks in unsupervised feature learning," in *International Conference on Artificial Intelligence and Statistics*, pp. 215–223, 2011.

[18] M. Ranzato, F. J. Huang, Y.-L. Boureau, and Y. Lecun, "Unsupervised learning of invariant feature hierarchies with applications to object recognition," in *CVPR*, 2007.

[19] K. Labusch, E. Barth, and T. Martinetz, "Simple method for high-performance digit recognition based on sparse coding.," *IEEE transactions on neural networks / a publication of the IEEE Neural Networks Council*, vol. 19, pp. 1985–9, Nov. 2008.

[20] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng, "Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations," in *Proceedings of the 26th Annual International Conference on Machine Learning*, pp. 609–616, ACM, 2009.

[21] A. Coates and A. Y. Ng, "The importance of encoding versus training with sparse coding and vector quantization," in *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pp. 921–928, 2011.

[22] M. D. Zeiler, G. W. Taylor, and R. Fergus, "Adaptive deconvolutional networks for mid and high level feature learning," in *Computer Vision (ICCV), 2011 IEEE International Conference on*, pp. 2018–2025, IEEE, 2011.

[23] A. Coates and A. Y. Ng, "Learning feature representations with k-means," in *Neural Networks: Tricks of the Trade*, pp. 561–580, Springer, 2012.

[24] B. A. Wandell, *Foundations of vision*. Sinauer Associates, 1995. Available from https://foundationsofvision.stanford.edu/.

[25] H. Kolb, E. Fernandez, and R. Nelson, eds., *Webvision: The Organization of the Retina and Visual System*. Salt Lake City (UT): University of Utah Health Sciences Center, 1995. Available from http://www.ncbi.nlm.nih.gov/books/NBK11530/.

[26] S. Filipe and L. a. Alexandre, "From the human visual system to the computational models of visual attention: a survey," *Artificial Intelligence Review*, Jan. 2013.

[27] J. Sjöstrand, N. Conradi, and L. Klarén, "How many ganglion cells are there to a foveal cone?," *Graefe's Archive for Clinical and Experimental Ophthalmology*, vol. 232, no. 7, p. 432437, 1994.

[28] G. D. Field and E. J. Chichilnisky, "Information processing in the primate retina: circuitry and coding.," *Annual review of neuroscience*, vol. 30, pp. 1–30, Jan. 2007.

[29] D. M. Dacey and M. R. Petersen, "Dendritic field size and morphology of midget and parasol ganglion cells of the human retina.," *Proc. of the National Academy of Sciences (PNAS)*, vol. 89, pp. 9666–70, Oct. 1992.

[30] M. Carandini, J. B. Demb, V. Mante, D. J. Tolhurst, Y. Dan, B. a. Olshausen, J. L. Gallant, and N. C. Rust, "Do we know what the early visual system does?," *The Journal of Neuroscience*, vol. 25, no. 46, pp. 10577–97, 2005.

[31] V. Bonin, V. Mante, and M. Carandini, "The suppressive field of neurons in lateral geniculate nucleus," *The Journal of neuroscience*, vol. 25, pp. 10844–56, Nov. 2005.

[32] R. Rodieck, "Quantitative analysis of cat retinal ganglion cell response to visual stimuli," *Vision Research*, vol. 5, pp. 583–601, Dec. 1965.

[33] C. Enroth-Cugell and J. G. Robson, "The contrast sensitivity of retinal ganglion cells of the cat.," *The Journal of physiology*, vol. 187, pp. 517–52, Dec. 1966.

[34] L. J. Croner and E. Kaplan, "Receptive fields of P and M ganglion cells across the primate retina.," *Vision Research*, vol. 35, pp. 7–24, Jan. 1995.

[35] I. M. Finn and D. Ferster, "Computational diversity in complex cells of cat primary visual cortex.," *The Journal of Neuroscience*, vol. 27, pp. 9638–48, Sept. 2007.

[36] I. Kagan, M. Gur, and M. Snodderly, "Modeling V1 complex cells in alert monkeys," in *CSH meeting on Computational and Systems Neuroscience (COSYNE)*, 2004.

[37] M. J. Hawken and A. J. Parker, "Spatial properties of neurons in the monkey striate cortex.," *Proceedings of the Royal Society of London. Series B, Containing papers of a Biological character. Royal Society (Great Britain)*, vol. 231, pp. 251–88, July 1987.

[38] A. J. Parker and M. J. Hawken, "Two-dimenisonal spatial structure of receptive fields in monkey striate cortex," *Journal of the Optical Society of America. A*, vol. 5, pp. 598–605, 1988.

[39] S. Marčelja, "Mathematical description of the responses of simple cortical cells," *JOSA*, vol. 70, no. 11, pp. 1297–1300, 1980.

[40] D. L. Ringach, "Spatial structure and symmetry of simple-cell receptive fields in macaque primary visual cortex.," *Journal of neurophysiology*, vol. 88, pp. 455–63, July 2002.

[41] S. Lyu and E. Simoncelli, "Nonlinear image representation using divisive normalization," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pp. 1–8, June 2008.

[42] N. Pinto, D. D. Cox, and J. J. DiCarlo, "Why is real-world visual object recognition hard?," *PLoS computational biology*, vol. 4, p. e27, Jan. 2008.

[43] S. Zhong, "Efficient online spherical k-means clustering," in *Neural Networks, 2005. IJCNN'05. Proceedings. 2005 IEEE International Joint Conference on*, vol. 5, pp. 3180–3185, IEEE, 2005.

[44] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[45] C.-Y. Lee, S. Xie, P. Gallagher, Z. Zhang, and Z. Tu, "Deeply-supervised nets," *arXiv preprint arXiv:1409.5185*, 2014.

[46] W. S. Geisler, "Sequential ideal-observer analysis of visual discriminations.," *Psychological review*, vol. 96, no. 2, p. 267, 1989.

[47] A. B. Watson and A. J. Ahumada, "A standard model for foveal detection of spatial contrast," *Journal of Vision*, vol. 5, no. 9, p. 6, 2005.