Learning long-range spatial dependencies with horizontal gated recurrent units

Drew Linsley, Junkyung Kim, Vijay Veerabadran, Charlie Windolf, Thomas Serre
Carney Institute for Brain Science
Department of Cognitive Linguistic & Psychological Sciences
Brown University
Providence, RI 02912
`{drew_linsley,junkyung_kim,vijay_veerabadran,`
`charles_windolf,thomas_serre}@brown.edu`

Abstract

Progress in deep learning has spawned great successes in many engineering applications. As a prime example, convolutional neural networks, a type of feedforward neural networks, are now approaching – and sometimes even surpassing – human accuracy on a variety of visual recognition tasks. Here, however, we show that these neural networks and their recent extensions struggle in recognition tasks where co-dependent visual features must be detected over long spatial ranges. We introduce a visual challenge, Pathfinder, and describe a novel recurrent neural network architecture called the horizontal gated recurrent unit (hGRU) to learn intrinsic horizontal connections – both within and across feature columns. We demonstrate that a single hGRU layer matches or outperforms all tested feedforward hierarchical baselines including state-of-the-art architectures with orders of magnitude more parameters.

1   Introduction

Consider Fig. 1a which shows a sample image from a representative segmentation dataset [1] (left) and the corresponding contour map produced by a state-of-the-art deep convolutional neural network (CNN) [2] (right). Although this task has long been considered challenging because of the need to integrate global contextual information with inherently ambiguous local edge information, modern CNNs are capable to detect contours in natural scenes at a level that rivals that of human observers [2–6]. Now, consider Fig. 1b which depicts a variant of a visual psychology task referred to as "Pathfinder" [7]. Reminiscent of the everyday task of reading a subway map to plan a commute (Fig. 1c), the goal in Pathfinder is to determine if two white circles in an image are connected by a path. These images are visually simple compared to natural images like the one shown in Fig. 1a, and the task is indeed easy for human observers [7]. Nonetheless, we will demonstrate that modern CNNs struggle to solve this task.

Why is it that a CNN can accurately detect contours in a natural scene like Fig. 1a but also struggle to integrate paths in the stimuli shown in Fig. 1b? In principle, the ability of CNNs to learn such long-range spatial dependencies is limited by their localized receptive fields (RFs) – hence the need to consider deeper networks because they allow the buildup of larger and more complex RFs. Here, we use a large-scale analysis of CNN performance on the Pathfinder challenge to demonstrate that simply increasing depth in feedforward networks constitutes an inefficient solution to learning the long-range spatial dependencies needed to solve the Pathfinder challenge.

An alternative solution to problems that stress long-range spatial dependencies is provided by biology. The visual cortex contains abundant horizontal connections which mediate non-linear interactions
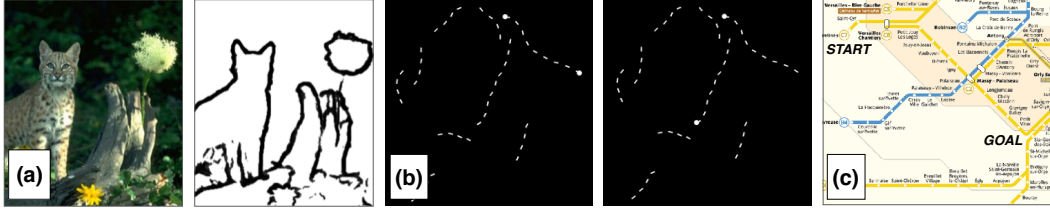
Figure 1: State-of-the-art CNNs excel at detecting contours in natural scenes, but they are strained by a task that requires the detection of long-range spatial dependencies. (a) Representative contour detection performance of a leading neural network architecture [23]. (b) Exemplars from the Pathfinder challenge: a task consisting of synthetic images which are parametrically controlled for long-range dependencies. (c) Long-range dependencies similar to those in the Pathfinder challenge are critical for everyday behaviors, such as reading a subway map to navigate a city.

between neurons across distal regions of the visual field [8, 9]. These intrinsic connections, popularly called "association fields", are thought to form the main substrate for mechanisms of contour grouping according to Gestalt principles, by mutually exciting colinear elements while also suppressing clutter elements that do not form extended contours [10–15]. Such "extra-classical receptive field" mechanisms, mediated by horizontal connections, allow receptive fields to adaptively "grow" without additional processing depth. Building on previous computational neuroscience work [e.g., 10, 16–19], our group has recently developed a recurrent network model of classical and extra-classical receptive fields that is constrained by the anatomy and physiology of the visual cortex [20]. The model was shown to account for diverse visual illusions providing computational evidence for a novel canonical circuit that is shared across visual modalities.

Here, we show how this computational neuroscience model can be turned into a modern end-to-end trainable neural network module. We describe an extension of the popular gated recurrent unit (GRU) [21], which we call the horizontal GRU (hGRU). Unlike CNNs, which exhibit a sharp decrease in accuracy for increasingly long paths, we show that the hGRU is highly effective at solving the Pathfinder challenge with just *one layer* and a fraction of the number of parameters and training samples needed by CNNs. We further find that, when trained on natural scenes, the hGRU learns connection patterns that coarsely resemble anatomical patterns of horizontal connectivity found in the visual cortex, and exhibits a detection profile that strongly correlates with human behavior on a classic contour detection task [22].

Related work Much previous work on recurrent neural networks (RNNs) has focused on modeling sequences with learnable gates in the form of long-short term memory (LSTM) units [24] or gated recurrent units (GRUs) [21]. RNNs have also been extended to learning spatial dependencies in static images with broad applications [25–29]. In this approach, images are transformed into one-dimensional sequences that are used to train an RNN. In recent years, several approaches have introduced convolutions into RNNs, using the recursive application of convolutional filters as a method for increasing the depth of processing through time on tasks like object recognition and super-resolution without additional parameters [30–32]. Other groups have constrained these convolutional-RNNs with insights from neuroscience and cognitive science, engineering specific patterns of connectivity between processing layers [33–36]. The proposed hGRU builds on this line of biologically-inspired implementations of RNNs, adding connectivity patterns and circuit mechanisms that are typically found in computational neuroscience models of neural circuits [e.g., 10, 16–20].

Another class of models related to our proposed approach is Conditional Random Fields (CRFs), probabilistic models aimed at explicitly capturing associations between nearby features. The connectivity implemented in CRFs is similar to the horizontal connections used in the hGRU, and has been successfully applied as a post-processing stage in visual tasks such as segmentation [37, 38] to smooth out and increase the spatial resolution of prediction maps. Recently, such probabilistic methods have been successfully incorporated in a generative vision model shown to break text-based CAPTCHAs [39]. Originally formulated as probabilistic models, CRFs can also be cast as RNNs [40].

2 Horizontal gated recurrent units (hGRUs)

Original contextual neural circuit model We begin by referencing the recurrent neural model of contextual interactions developed by Mély et al. [20]. Below we adapted the model notations to a computer vision audience. Model units are indexed by their 2D positions (x, y) and feature channel k. Neural activity is governed by the following differential equations (see Supp. Material for the full treatment):

$$\eta \dot{H}^{(1)}_{xyk} + \epsilon^2 H^{(1)}_{xyk} = \left[\xi X_{xyk} - (\alpha H^{(1)}_{xyk} + \mu)\, C^{(1)}_{xyk} \right]_+$$
$$\tau \dot{H}^{(2)}_{xyk} + \sigma^2 H^{(2)}_{xyk} = \left[\gamma C^{(2)}_{xyk} \right]_+ . \tag{1}$$

where

$$C^{(1)}_{xyk} = (\mathbf{W} * \mathbf{H}^{(2)})_{xyk}$$
$$C^{(2)}_{xyk} = (\mathbf{W} * \mathbf{H}^{(1)})_{xyk},$$

Here, $\mathbf{X} \in \mathbb{R}^{W \times H \times K}$ is the feedforward drive (i.e., neural responses to a stimulus), $\mathbf{H}^{(1)} \in \mathbb{R}^{W \times H \times K}$ is the recurrent circuit input, and $\mathbf{H}^{(2)} \in \mathbb{R}^{W \times H \times K}$ the recurrent circuit output. Modeling input and output states separately allows for the implementation of a particular form of inhibition known as "shunting" (or divisive) inhibition. Unlike the excitation in the model which acts linearly on a unit's input, inhibition acts on a unit's output and hence, regulates the unit response non-linearly (i.e., given a fixed amount of inhibition and excitation, inhibition will increase with the unit's activity unlike excitation which will remained constant).

The convolutional kernel $\mathbf{W} \in \mathbb{R}^{S \times S \times K \times K}$ describes hypercolumn connectivity (constrained by anatomical data[1]). The scalar parameters μ and α control linear and quadratic (i.e., shunting) inhibition by $\mathbf{C}^{(1)} \in \mathbb{R}^{W \times H \times K}$, γ scales excitation by $\mathbf{C}^{(2)} \in \mathbb{R}^{W \times H \times K}$, and ξ scales the feedforward drive. Activity at each stage is linearly rectified (ReLU) $[\cdot]_+ = \max(\cdot, 0)$. Finally, η, ϵ, τ and σ are time constants. To make this model amenable to modern computer vision applications, we set out to develop a version where all parameters could be trained from data. If we let $\eta = \tau$ and $\sigma = \epsilon$ for symmetry and apply Euler's method to Eq. 1 with a time step of $\Delta t = \eta / \epsilon^2$, then we obtain the discrete-time equations:

$$H^{(1)}_{xyk}[t] = \epsilon^{-2} \left[\xi X_{xyk} - (\alpha H^{(1)}_{xyk}[t-1] + \mu) C^{(1)}_{xyk}[t] \right]_+$$
$$H^{(2)}_{xyk}[t] = \epsilon^{-2} \left[\gamma C^{(2)}_{xyk}[t] \right]_+ . \tag{2}$$

Here, $\cdot[t]$ denotes the approximation at the t-th discrete timestep. This results in a trainable convolutional recurrent neural network (RNN) which performs Euler integration of a dynamical system similar to the neural model of [20].

hGRU formulation We build on Eq. 2 to introduce the hGRU – a model with the ability to learn complex interactions between units via horizontal connections within a single processing layer (Fig. 2). The hGRU extends the derivation from Eq. 2 with three modifications that improve the training of the model with gradient descent and its expressiveness[2]. (i) We introduce learnable gates, borrowed from the gated recurrent unit (GRU) framework (see Supp. Material for the full derivation from Eq. 2). (ii) The hGRU makes the operations for computing $\mathbf{H}^{(2)}$ (excitation) symmetric with those of $\mathbf{H}^{(1)}$ (inhibition), providing the circuit the ability to learn how to implement linear and quadratic interactions at each of these processing stages. (iii) To control unstable gradients, the hGRU uses a squashing pointwise non-linearity and a learned parameter to globally scale activity at every processing timestep (akin to a constrained version of the recurrent batchnorm [41]).

---

[1]There are four separate connectivity patterns in [20] to describe excitatory vs. inhibitory and near vs. far interactions between units. We combine these into a single kernel to simplify notation.

[2]These modifications involved relaxing several constraints from the original neuroscience model that are less useful for solving the tasks investigated here (see Supp. Material for performance of an hGRU with constrained inhibition and excitation.)
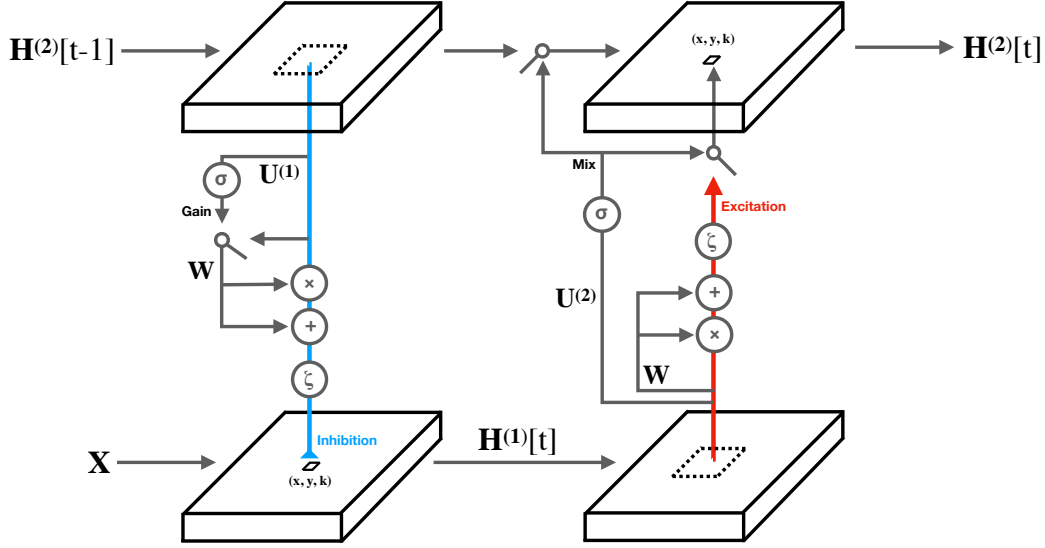
Figure 2: The hGRU circuit. The hGRU can learn highly non-linear interactions between spatially neighboring units in the feedforward drive $\mathbf{X}$, which are encoded in its hidden state $\mathbf{H}^{(2)}$. This computation involves two stages, which are inspired by a recurrent neural circuit of horizontal connections [20]. First, the horizontal inhibition (blue) is calculated by applying a gain to $\mathbf{H}^{(2)}[t-1]$, and convolving the resulting activity with the kernel W which characterizes these interactions. Linear (+ symbol) and quadratic (× symbol) operations control the convergence of this inhibition onto $\mathbf{X}$. Second, the horizontal excitation (red) is computed by convolving $\mathbf{H}^{(1)}[t]$ with W. Another set of linear and quadratic operations modulate this activity, before it is mixed with the persistent hidden state $\mathbf{H}^{(2)}[t-1]$. Note that the excitation computation involves an additional "peephole" connection, not depicted here. Small solid-line squares within the hypothetical activities that the circuit operates on denote the unit indexed by 2D position (x, y) and feature channel k, whereas dotted-line squares depict the unit's receptive field (a union of both classical and extra-classical definitions) in the previous activity.

In our hGRU implementation, the feedforward drive $\mathbf{X}$ corresponds to activity from a preceding convolutional layer. The hGRU encodes spatial dependencies between feedforward units via its (time-varying) hidden states $\mathbf{H}^{(1)}$ and $\mathbf{H}^{(2)}$. Updates to the hidden states are managed using two activities, referred to as the reset and update "gates": $\mathbf{G}^{(1)}$ and $\mathbf{G}^{(2)}$. These activities are derived from convolutions, denoted by $*$, between the kernels $\mathbf{U}^{(1)}, \mathbf{U}^{(2)} \in \mathbb{R}^{1 \times 1 \times K \times K}$ and hidden states $\mathbf{H}^{(1)}$ and $\mathbf{H}^{(2)}$, shifted by biases $\mathbf{b}^{(1)}, \mathbf{b}^{(2)} \in \mathbb{R}^{1 \times 1 \times K}$, respectively. The pointwise non-linearity σ is applied to each activity, normalizing them in the range $[0, 1]$. Because these activities are real-valued, we hereafter refer to the reset gate as the "gain", and the update gate as the "mix".

Horizontal interactions between units are calculated by the kernel $\mathbf{W} \in \mathbb{R}^{S \times S \times K \times K}$, where S describes the spatial extent of these connections in a single timestep (Fig. 2). Consistent with computational models of neural circuits (e.g., [10, 16–20]), $\mathbf{W}$ is constrained to have symmetric weights between channels. This means that the weight $W_{x_0+\Delta x, y_0+\Delta y, k_1, k_2}$ will have the same value as the weight $W_{x_0+\Delta x, y_0+\Delta y, k_2, k_1}$ where x_0 and y_0 denote the center of the kernel. This constraint reduces the number of learnable parameters by nearly half compared to a normal convolutional kernel. The hidden states $\mathbf{H}^{(1)}$ and $\mathbf{H}^{(2)}$ are recomputed via horizontal interactions at every timestep $t \in [0, T]$. We begin by describing the computation of $\mathbf{H}^{(1)}[t]$ which is given by the following equations:

$$\mathbf{G}^{(1)}[t] = \sigma(\mathbf{U}^{(1)} * \mathbf{H}^{(2)}[t-1] + \mathbf{b}^{(1)}) \qquad (3)$$

$$C^{(1)}_{xyk}[t] = (\mathbf{W} * (\mathbf{G}^{(1)}[t] \odot \mathbf{H}^{(2)}[t-1]))_{xyk} \qquad (4)$$

$$H^{(1)}_{xyk}[t] = \zeta(X_{xyk} - C^{(1)}_{xyk}[t](\alpha_k H^{(2)}_{xyk}[t-1] + \mu_k)) \qquad (5)$$

Channels in $\mathbf{H}^{(2)}[t-1]$ are first modulated by the gain[3] $\mathbf{G}^{(1)}[t]$. The resulting activity is convolved with $\mathbf{W}$ to compute $\mathbf{C}^{(1)}[t]$, which is the horizontal inhibition of the hGRU at this timestep. This inhibition is applied to $\mathbf{X}$ via the parameters $\boldsymbol{\mu}$ and $\boldsymbol{\alpha}$, which are k-dimensional vectors that respectively scale linear and quadratic (akin to shunting inhibition described in Eq. 1) terms of the horizontal interaction with $\mathbf{X}$. The pointwise ζ is a hyperbolic tangent that squashes activity into the range $[-1, 1]$ (but see Supp. Material for a hGRU with a rectified linearity). Importantly, in contrast to the original circuit, in this formulation the update to $\mathbf{H}^{(1)}[t]$ (Eq. 5) is calculated by combining horizontal connection contributions of $\mathbf{C}^{(1)}[t]$ with $\mathbf{H}^{(2)}[t-1]$ rather than $\mathbf{H}^{(1)}[t-1]$, which we found improved learning on the visual tasks explored here.

The updated $\mathbf{H}^{(1)}[t]$ is next used to calculate $\mathbf{H}^{(2)}[t]$.

$$G^{(2)}_{xyk}[t] = \sigma((\mathbf{U}^{(2)} * \mathbf{H}^{(1)}[t])_{xyk} + b^{(2)}_k) \tag{6}$$

$$C^{(2)}_{xyk}[t] = (\mathbf{W} * \mathbf{H}^{(1)}[t])_{xyk} \tag{7}$$

$$\tilde{H}^{(2)}_{xyk}[t] = \zeta(\kappa_k H^{(1)}_{xyk}[t] + \beta_k C^{(2)}_{xyk}[t] + \omega_k H^{(1)}_{xyk}[t] C^{(2)}_{xyk}[t]) \tag{8}$$

$$H^{(2)}_{xyk}[t] = \eta_t(H^{(2)}_{xyk}[t-1](1 - G^{(2)}_{xyk}[t]) + \tilde{H}^{(2)}_{xyk}[t] G^{(2)}_{xyk}[t]) \tag{9}$$

The mix $\mathbf{G}^{(2)}[t]$ is calculated by convolving $\mathbf{U}^{(2)}[t]$ with $\mathbf{H}^{(1)}[t]$, followed by the addition of $\mathbf{b}^{(2)}$. The activity $\mathbf{C}^{(2)}[t]$ represents the excitation of horizontal connections onto the newly-computed $\mathbf{H}^{(1)}[t]$. Linear and quadratic contributions of horizontal interactions at this stage are controlled by the k-dimensional parameters $\boldsymbol{\kappa}$, $\boldsymbol{\omega}$, and $\boldsymbol{\beta}$. The parameters $\boldsymbol{\kappa}$ and $\boldsymbol{\omega}$ control the linear and quadratic contributions of horizontal connections to $\tilde{\mathbf{H}}^{(2)}[t]$. The parameter $\boldsymbol{\beta}$ is a gain applied to $\mathbf{C}^{(2)}[t]$, giving $\mathbf{W}$ an additional degree of freedom in expressing this excitation. With this full suite of interactions, the hGRU can in principle implement both a linear and a quadratic form of excitation (i.e., to assess self-similarity), each of which play specific computational roles in perception [42]. Note that the inclusion of $\mathbf{H}^{(1)}[t]$ in Eq. 8 functions as a "peephole" connection between it and $\tilde{\mathbf{H}}^{(2)}[t]$. Finally, the mix $\mathbf{G}^{(2)}$ integrates the candidate $\tilde{\mathbf{H}}^{(2)}_t$ with $\mathbf{H}^{(2)}_t$. The learnable T-dimensional parameter $\boldsymbol{\eta}$, which we refer to as a time-gain, helps control unstable gradients during training. This time-gain modulates $\mathbf{H}^{(2)}_t$ with the scalar, η_t, which as we show in our experiments below improves model performance.

3 The Pathfinder challenge

We evaluated the limits of feedforward and recurrent architectures on the "Pathfinder challenge", a synthetic visual task inspired by cognitive psychology [7]. The task, depicted in Fig. 1b, involves detecting whether two circles are connected by a path. This is made more difficult by allowing target paths to curve and introducing multiple shorter unconnected "distractor" paths. The Pathfinder challenge involves three separate datasets, for which the length of paths and distractors are parametrically increased. This challenge therefore screens models for their effectiveness in detecting complex long-range spatial relationships in cluttered scenes.

Stimulus design Pathfinder images were generated by placing oriented "paddles" on a canvas to form dashed paths. Each image contained two paths made of a fixed number of paddles and multiple distractors made of one third as many paddles. Positive examples were generated by placing two circles at the ends of a single path (Fig. 1b, left) and negative examples by placing one circle at the end of each of the paths (Fig. 1b, right). The paths were curved and variably shaped, with the possible number of shapes exponential to the path length. The Pathfinder challenge consisted of three datasets, in which path and distractor length was successively increased, and with them, the overall task difficulty. These datasets had path lengths of 6, 9 and 14 paddles, and each contained 1,000,000 unique images of 150×150 pixels. See Supp. Material for a detailed description of the stimulus generation procedure.

---

[3]GRU gate activities are often a function of a hidden state and $\mathbf{X}[t]$. Because the feedforward drive here is constant w.r.t. time, we omit it from these calculations. In practice, its inclusion did not affect performance.

Model implementation   We performed a large-scale analysis of the effectiveness of feedforward and recurrent computations on the Pathfinder challenge. We controlled for the effects of idiosyncratic model specifications by using a standard architecture, consisting of "input", "feature extraction", and "readout" processing stages. Swapping different feedforward or recurrent layers into the feature extraction stage let us measure the relative effectiveness of each on the challenge. All models except for state-of-the-art "residual networks" (ResNets) [43] and per-pixel prediction architectures were embedded in this architecture, and these exceptions are detailed below. See Supp. Material for a detailed description of the input and readout stage. Models were trained on each Pathfinder challenge dataset (Fig. 3d), with 90% of the images used for training (900,000) and the remainder for testing (100,000). We measured model performance in two ways. First, as the accuracy on test images. Second, as the "area under the learning curve" (ALC), or mean accuracy on the test set evaluated after every 1000 batches of training, which summarized the rate at which a model learned the task. Accuracy and ALC were taken from the model that achieved the highest accuracy across 5 separate runs of model training. All models were trained for two epochs except for the ResNets, which were trained for four. Model training procedures are detailed in Supp. Material.

Recurrent models   We tested 6 different recurrent layers in the feature extraction stage of the standard architecture: hGRUs with 8, 6, and 4-timesteps of processing; a GRU; and hGRUs with lesions applied to parameters controlling linear or quadratic horizontal interactions. Both the GRU and lesioned versions of the hGRU ran for 8 timesteps. These layers had 15×15 horizontal connection kernels (W) with an equal number of channels as their input layer (25 channels).

We observed 3 overarching trends: First, each model's performance monotonically decreased, or "strained", as path length increased. Increasing path length reduced model accuracy (Fig. 3a), and increased the number of batches it took to learn a task (Fig. 3b). Second, the 8-timestep hGRU was more effective than any other recurrent model, and it outperformed each of its lesioned variants as well as a standard GRU. Notably, this hGRU was strained the *least* by the Pathfinder challenge out of all tested models, with a negligible drop in accuracy as path length increased. This finding highlights the effectiveness of the hGRU for processing long-range spatial dependencies, and how the dynamics implemented by its linear and quadratic horizontal interactions are important. Third, hGRU performance monotonically decreased with processing time. This revealed a minimum number of timesteps that the hGRU needed to solve each Pathfinder dataset: 4 for the length-6 condition, 6 for the length-9 condition, and 8 for the length-14 condition (first vs. second columns in Fig. 3a). Such time-dependency in the Pathfinder task is consistent with the accuracy-reaction-time tradeoff found in humans as the distance between endpoints of a curve increases [7].

Feedforward models   We screened an array of feedforward models on the Pathfinder challenge. Model performance revealed the importance of kernel size vs. kernel width, model depth, and feedforward operations for incorporating additional scene context for solving Pathfinder. Model construction began by embedding the feature extraction stage of the standard model with kernels of one of three different sizes: 10×10, 15×15, or 20×20. These are referred to as small, medium, and large kernel models (Fig. 3). To control for the effect of network capacity on performance, the number of kernels given to each model was varied so that the number of parameters in each model configuration was equal to each other and the hGRU (36, 16, and 9 kernels). We also tested two other feedforward models that featured candidate operations for incorporating contextual information into local convolutional activities. One version used (2-pixel) dilated convolutions, which involves applying a stride to the kernel before convolving the input [44, 45], and has been found useful for many computer vision problems [38, 46, 47]. The other version applied a non-local operation to convolutional activities [48], which can introduce (non-recurrent) interactions between units in a layer. These operations were incorporated into the first feature extraction layer of the medium kernel (15×15 filter) model described above. We also considered deeper versions of each of the above "1-layer" models (referring to the depth of the feature extraction stage), stacking them to build 3- and 5-layer versions. This yielded a total of 15 different feedforward models.

Without exception, the performance of each feedforward model was significantly strained by the Pathfinder challenge. The magnitude of this straining was well predicted by model depth and size, and operations for incorporating additional contextual information made no discernible difference to the overall pattern of results. The 1-layer models were most effective on the 6-length Pathfinder dataset, but were unable to do better than chance on the remaining conditions. Increasing model capacity to 3 layers rescued the performance of all but the small kernel model on the 9-length Pathfinder dataset,
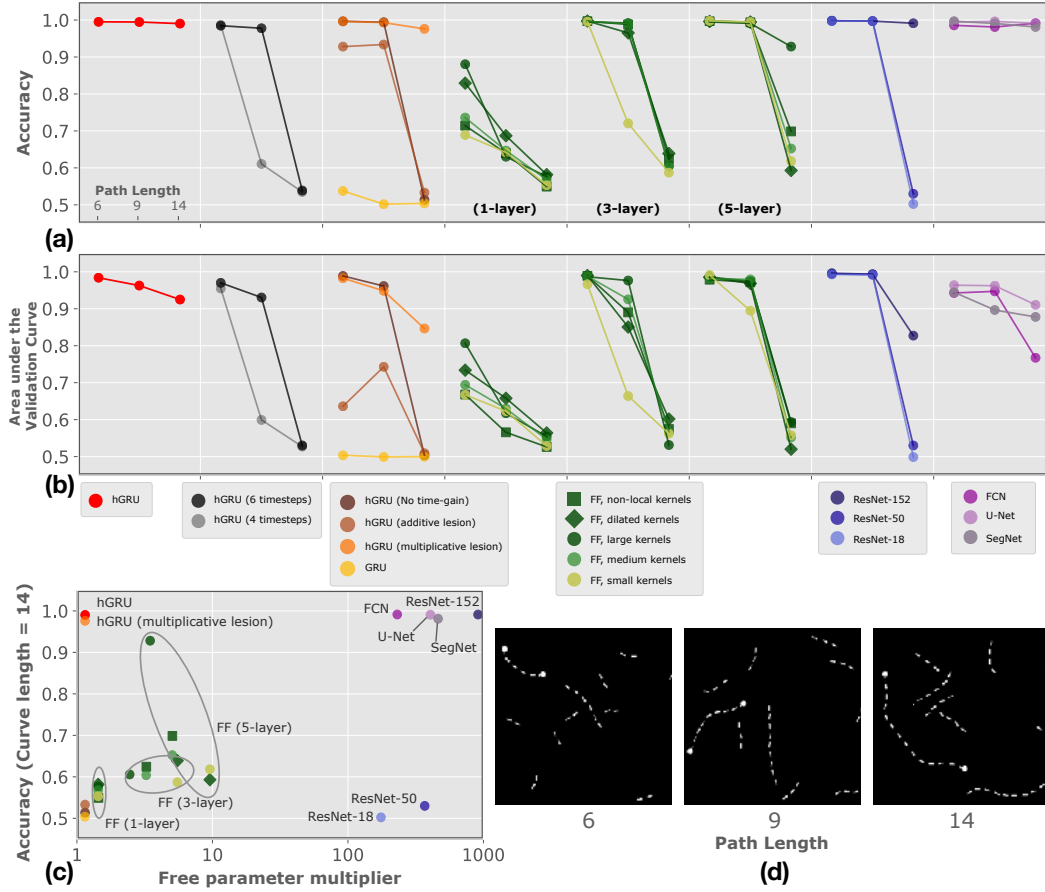
Figure 3: The hGRU efficiently learns long-range spatial dependencies that otherwise strain feedforward architectures. (a) Model accuracy is plotted for the three Pathfinder challenge datasets, which featured paths of 6- 9- and 14-paddles. Each panel depicts the accuracy of a different model class after training on for each pathfinder dataset (see Supp. Material for additional models). Only the hGRU and state-of-the-art models for classification (the two right-most panels) approached perfect accuracy on each dataset. (b) Measuring the area under the learning curve (ALC) of each model (mean accuracy) demonstrates that the rate of learning achieved by the hGRU across the Pathfinder challenge is only rivaled by the U-Net architecture (far right). (c) The hGRU is significantly more parameter-efficient than feedforward models at the Pathfinder challenge, with its closest competitors needing at least 200× the number of parameters to match its performance. The x-axis shows the number of parameters in each model versus the hGRU, as a multiple of the latter. The y-axis depicts model accuracy on the 14-length Pathfinder dataset. (d) Pathfinder challenge exemplars of different path lengths (all are positive examples).

but even then did little to improve performance on the 14-length dataset. Of the 5-layer models, only the large kernel configuration came close to solving the 14-length dataset. The ALC of this model, however, demonstrates that its rate of learning was slow, especially compared to the hGRU (Fig. 3b). The failures of these feedforward models is all the more striking when considering that each had between 1× and 10× the number of parameters as the hGRU (Fig. 3c, compare the red and green markers).

Residual networks We reasoned that if the performance of feedforward models on the Pathfinder challenge is a function of model depth, then state-of-the-art networks for object recognition with many times the number of layers should easily solve the challenge. We tested this possibility by

training ResNets with 18, 50, and 152 layers on the Pathfinder challenge. Each model was trained "from scratch" with standard weight initialization [49], and given additional epochs of training (4) to learn the task because of their large number of parameters. However, even with this additional training, only the deepest 152-layer ResNet was able to solve the challenge (Fig. 3a). Even so, the 152-layer ResNet was less efficient at learning the 14-length dataset than the hGRU (Fig. 3b), and achieved its performance with nearly $1000\times$ as many parameters (Fig. 3c; see Supp. Material for additional ResNet experiments).

Per-pixel prediction models   We considered the possibility that CNN architectures for per-pixel prediction tasks, such as contour detection and segmentation, might be better suited to the Pathfinder challenge than those designed for classification. We therefore tested three representative per-pixel prediction models: the fully-convolutional network (FCN), the skip-connection U-Net, and the unpooling SegNet. These models used an encoder/decoder style architecture, which was followed by the readout processing stage of the standard architecture described above to make them suitable for Pathfinder classification. Encoders were the VGG16 [50], and each model was trained from scratch with Xavier initialized weights. Like the ResNets, these models were given 4 epochs of training to accommodate their large number of parameters.

The fully-convolutional network (FCN) architecture is one of the first successful uses of CNNs for per-pixel prediction [3, 44, 51, 52]. Decoders in these models use "1×1" convolutions to combine upsampled activity maps from several layers of the encoder. We created an FCN model which applied this procedure to the last layer of each of the 5 VGG16-convolution blocks. These activity maps were upsampled by learnable kernels, which were initialized with weights for bilinear interpolation. In contrast to the feedforward models discussed above, the FCN successfully learned all conditions in the Pathfinder challenge (Fig. 3a, purple circle). It did so less efficiently than the hGRU, however, with a lower ALC score on the 14-length dataset and $200\times$ as many free parameters (Fig. 3b).

Another approach to per-pixel prediction uses "skip connections" to connect specific layers of a model's encoder to its decoder. This approach was first described in [44] as a method for more effectively merging coarse-layer information into a model's decoder, and later extended to the U-Net [53]. We implemented a version of the U-Net architecture that had a VGG16 encoder and a decoder. The decoder consisted of 5 randomly initialized and learned upsampling layers, which had additive connections to the final convolutional layer in each of the encoder's VGG16 blocks. Using standard VGG16 nomenclature to define one of these connections, this meant that "conv 4_3" activity from the encoder was added to the second upsampled activity map in the decoder. The U-Net was on par with the hGRU and the FCN at solving the Pathfinder challenge. It was also nearly as efficient as the hGRU in doing so (Fig. 3b), but used over $350\times$ as many parameters as the hGRU (Fig. 3c; see Supp. Materials for additional U-Net experiments).

Unpooling models eliminate the need for feature map upsampling by routing decoded activities to the locations of the winning max-pooling units derived from the encoder. Unpooling is also a leading approach for a variety of dense per-pixel prediction tasks, including segmentation, which is exemplified by SegNet [54]. We tested a SegNet on the Pathfinder challenge. This model has a decoder that mirrors its encoder, with unpooling operations replacing its pooling. The SegNet achieved high accuracy on each of the Pathfinder datasets, but was less efficient at learning them than the hGRU, with worse ALC scores across the challenge (Fig. 3b). The SegNet also featured the second-most parameters of any model tested, which was $400\times$ more than the hGRU.

4   Explaining biological horizontal connections with the hGRU

Statistical image analysis studies have suggested that cortical patterns of horizontal connections, commonly referred to as "association fields", may reflect the geometric regularities of oriented elements present in natural scenes [55]. Because the hGRU is designed to capture such spatial regularities, we investigated whether it learned patterns of horizontal connections that resemble these association fields. We visualized the horizontal kernels learned by the hGRU to solve tasks (Fig. S5). When trained on the the Pathfinder challenge, hGRU kernels resembled the dominant patterns of horizontal connectivity in visual cortex. Prominent among these patterns are (1) the antagonistic near-excitatory vs. far-inhibitory surround organization also found in the visual cortex [56]; (2) the association field, with collinear excitation and orthogonal inhibition [8, 9]; and (3) other higher-order surround computations [57]. We also visualized these patterns after training the hGRU to detect

contours in the naturalistic BSDS500 image dataset [1]. These horizontal kernels took on similar patterns of connectivity, but with far more definition and regularity, suggesting that the hGRU learns best from natural scene statistics.

How well does the hGRU explain human psychophysics data? We tested this by recreating the synthetic contour detection dataset used in [22]. This task had human participants detect a contour formed by co-linearly aligned paddles in an array of randomly oriented distractors. Multiple versions of the task were created by varying the distance between paddles in the contour (5 conditions). Contour detection accuracy of the hGRU was recorded on each of dataset for comparison with participants in [22], whose responses were digitally extracted with WebPlotDigitizer from [22] and averaged (N=2). Plotting hGRU accuracy against the reported "detection score" revealed that increasing inter-paddle distance caused similar performance straining for both (Fig. S6).

5 Discussion

The present study demonstrates that long-range spatial dependencies generally strain CNNs, with only very deep and state-of-the-art networks overcoming the visual variability introduced by long paths in the Pathfinder challenge. Although feedforward networks are generally effective at learning and detecting relatively rigid objects shown in well-defined poses, these models tend towards a brute-force solution when tasked with the recognition of less constrained structures, such as a path connecting two distant locations. This study adds to a body of work highlighting examples of routine visual tasks where CNNs fall short of human performance [58–62].

We demonstrate a solution to the Pathfinder challenge inspired by neuroscience. The hGRU leverages computational principles of visual cortical circuits to learn complex spatial interactions between units. For the Pathfinder challenge, this translates into an ability to represent the elements forming an extended path while ignoring surrounding clutter. We find that the hGRU can reliably detect paths of any tested length or form using just a single layer. This contrasts sharply with the successful state-of-the-art feedforward alternatives, which used much deeper architectures and orders of magnitude more parameters to achieve similar success. The key mechanisms underlying the hGRU's performance are well known in computational neuroscience [10, 16–20]. However, these mechanisms have been typically overlooked in computer vision (but see [39] for a successful vision model using horizontal connections and shown to break text-based CAPTCHAs).

We also found that hGRU performance on the Pathfinder challenge is a function of the *amount of time* it was given for processing. This finding suggests that it concurrently expands the facilitative influence of one end of a target curve to the other while suppressing the influence of distractors. The performance of the hGRU on the Pathfinder challenge captures the iterative nature of computations used by our visual system during similar tasks [63] – exhibiting a comparable tradeoff between performance and processing-time [7]. Visual cortex is replete with association fields that are thought to underlie perceptual grouping [11, 64]. Theoretical models suggest that patterns of horizontal connections reflect the statistics of natural scenes, and here too we find that horizontal kernels in the hGRU learned from natural scenes resemble cortical patterns of horizontal connectivity, including association fields and the paired near-excitatory / far-inhibitory surrounds that may be responsible for many contextual illusions [20, 56]. The horizontal connections learned by the hGRU reproduce another aspect of human behavior, in which the saliency of a straight contour decreases as the distance between its paddles increases. This sheds light on a possible relationship between horizontal connections and saliency computation.

In summary, this work diagnoses a computational deficiency of feedforward networks, and introduces a biologically-inspired solution that can be easily incorporated into existing deep learning architectures. The weights and patterns of behavior learned by the hGRU appear consistent with those associated with the visual cortex, demonstrating its potential for establishing novel connections between machine learning, cognitive science, and neuroscience.

References

[1] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 33(5):898–916, May 2011.

[2] C.-Y. Lee, S. Xie, P. Gallagher, Z. Zhang, and Z. Tu. Deeply-Supervised nets. In *Artificial Intelligence and Statistics*, pages 562–570, February 2015.

[3] S. Xie and Z. Tu. Holistically-Nested edge detection. *International journal of computer vision*, 125(1):3–18, December 2017.

[4] Y. Liu, M. M. Cheng, X. Hu, K. Wang, and X. Bai. Richer convolutional features for edge detection. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5872–5881, July 2017.

[5] K.-K. Maninis, J. Pont-Tuset, P. Arbelaez, and L. Van Gool. Convolutional oriented boundaries: From image segmentation to High-Level tasks. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):819–833, April 2018.

[6] Y. Wang, X. Zhao, and K. Huang. Deep crisp boundaries. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3892–3900, 2017.

[7] R. Houtkamp and P. R. Roelfsema. Parallel and serial grouping of image elements in visual perception. *Journal of experimental psychology. Human perception and performance*, 36(6): 1443–1459, December 2010.

[8] D. D. Stettler, A. Das, J. Bennett, and C. D. Gilbert. Lateral connectivity and contextual interactions in macaque primary visual cortex. *Neuron*, 36(4):739–750, November 2002.

[9] K. S. Rockland and J. S. Lund. Intrinsic laminar lattice connections in primate visual cortex. *The Journal of comparative neurology*, 216(3):303–318, May 1983.

[10] S. Grossberg and E. Mingolla. Neural dynamics of perceptual grouping: textures, boundaries, and emergent segmentations. *Perception & psychophysics*, 38(2):141–171, August 1985.

[11] D. J. Field, A. Hayes, and R. F. Hess. Contour integration by the human visual system: Evidence for a local "association field". *Vision research*, 33(2):173–193, 1993.

[12] G. W. Lesher and E. Mingolla. The role of edges and line-ends in illusory contour formation. *Vision research*, 33(16):2253–2270, November 1993.

[13] W. Li, V. Piëch, and C. D. Gilbert. Contour saliency in primary visual cortex. *Neuron*, 50(6): 951–962, June 2006.

[14] W. Li, V. Piëch, and C. D. Gilbert. Learning to link visual contours. *Neuron*, 57(3):442–451, February 2008.

[15] S. W. Zucker. Stereo, shading, and surfaces: Curvature constraints couple neural computations. *Proceedings of the IEEE*, 102(5):812–829, May 2014.

[16] P. Series, J. Lorenceau, and Y. Frégnac. The "silent" surround of V1 receptive fields: theory and experiments. *Journal of physiology-Paris*, 97:453–474, 2003.

[17] L. Zhaoping. Neural circuit models for computations in early visual cortex. *Current opinion in neurobiology*, 21(5):808–815, October 2011.

[18] S. Shushruth, P. Mangapathy, J. M. Ichida, P. C. Bressloff, L. Schwabe, and A. Angelucci. Strong recurrent networks compute the orientation tuning of surround modulation in the primate primary visual cortex. *Journal of Neuroscience*, 32(1):308–321, 2012.

[19] D. B. Rubin, S. D. Van Hooser, and K. D. Miller. The stabilized supralinear network: a unifying circuit motif underlying multi-input integration in sensory cortex. *Neuron*, 85(2):402–417, January 2015.

[20] D. A. Mély, D. Linsley, and T. Serre. Complementary surrounds explain diverse contextual phenomena across visual modalities. *Psychological review*, 125(5):769–784, October 2018.

[21] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using RNN Encoder–Decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Stroudsburg, PA, USA, 2014. Association for Computational Linguistics.

[22] W. Li and C. D. Gilbert. Global contour saliency and local colinear interactions. *Journal of neurophysiology*, 88(5):2846–2856, November 2002.

[23] S. Xie and Z. Tu. Holistically-Nested edge detection. *International journal of computer vision*, 125(1-3):3–18, December 2017.

[24] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8): 1735–1780, November 1997.

[25] A. Graves, S. Fernández, and J. Schmidhuber. Multi-dimensional recurrent neural networks. In *Proceedings of the 17th International Conference on Artificial Neural Networks*, ICANN'07, pages 549–558, Berlin, Heidelberg, 2007. Springer-Verlag.

[26] A. Graves and J. Schmidhuber. Offline handwriting recognition with multidimensional recurrent neural networks. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems 21*, pages 545–552. Curran Associates, Inc., 2009.

[27] S. Bell, C. Lawrence Zitnick, K. Bala, and R. Girshick. Inside-outside net: Detecting objects in context with skip pooling and recurrent neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2874–2883, 2016.

[28] L. Theis and M. Bethge. Generative image modeling using spatial LSTMs. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 1927–1935. Curran Associates, Inc., 2015.

[29] A. Van Den Oord, N. Kalchbrenner, and K. Kavukcuoglu. Pixel recurrent neural networks. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, pages 1747–1756, New York, NY, USA, 2016. JMLR.org.

[30] M. Liang and X. Hu. Recurrent convolutional neural network for object recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3367–3375, 2015.

[31] Q. Liao and T. Poggio. Bridging the gaps between residual learning, recurrent neural networks and visual cortex. *arXiv preprint arXiv:1604.03640*, April 2016.

[32] J. Kim, J. K. Lee, and K. M. Lee. Deeply-Recursive convolutional network for image Super-Resolution. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1637–1645. IEEE, June 2016.

[33] C. J. Spoerer, P. McClure, and N. Kriegeskorte. Recurrent convolutional neural networks: A better model of biological object recognition. *Frontiers in psychology*, 8:1551, September 2017.

[34] W. Lotter, G. Kreiman, and D. Cox. Deep predictive coding networks for video prediction and unsupervised learning. In *International Conference on Learning Representations (ICLR)*, 2017.

[35] A. R. Zamir, T.-L. Wu, L. Sun, W. Shen, B. E. Shi, J. Malik, and S. Savarese. Feedback Networks. *arXiv preprint arxiv: 1612.09508*, December 2016.

[36] A. Nayebi, D. Bear, J. Kubilius, K. Kar, S. Ganguli, D. Sussillo, J. J. DiCarlo, and D. L. K. Yamins. Task-Driven convolutional recurrent models of the visual system. *arXiv preprint arXiv:1807.00053*, June 2018.

[37] I. Kokkinos. Pushing the boundaries of boundary detection using deep learning. In *International Conference on Learning Representations (ICLR)*, 2016.

[38] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, April 2018.

[39] D. George, W. Lehrach, K. Kansky, M. Lázaro-Gredilla, C. Laan, B. Marthi, X. Lou, Z. Meng, Y. Liu, H. Wang, A. Lavin, and D. S. Phoenix. A generative vision model that trains with high data efficiency and breaks text-based CAPTCHAs. *Science*, 358(6368), December 2017.

[40] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. H. S. Torr. Conditional random fields as recurrent neural networks. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1529–1537. IEEE, December 2015.

[41] T. Cooijmans, N. Ballas, C. Laurent, Ç. Gülçehre, and A. Courville. Recurrent batch normalization. In *International Conference on Learning Representations (ICLR)*, 2017.

[42] D. R. Martin, C. C. Fowlkes, and J. Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE transactions on pattern analysis and machine intelligence*, 26(5):530–549, May 2004.

[43] K. He, X. Zhang, S. Ren, and J. Sun. Identity mappings in deep residual networks. In *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part IV*, pages 630–645, 2016.

[44] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[45] F. Yu and V. Koltun. Multi-Scale context aggregation by dilated convolutions. In *International Conference on Learning Representations (ICLR)*, 2016.

[46] T. Wang, M. Sun, and K. Hu. Dilated deep residual network for image denoising. In *29th IEEE International Conference on Tools with Artificial Intelligence, ICTAI 2017, Boston, MA, USA, November 6-8, 2017*, pages 1272–1279, 2017. doi: 10.1109/ICTAI.2017.00192.

[47] R. Hamaguchi, A. Fujita, K. Nemoto, T. Imaizumi, and S. Hikosaka. Effective use of dilated convolutions for segmenting small object instances in remote sensing imagery. *CoRR*, abs/1709.00179, 2017.

[48] X. Wang, R. Girshick, A. Gupta, and K. He. Non-Local Neural Networks. In *2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[49] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, ICCV '15, pages 1026–1034, Washington, DC, USA, 2015. IEEE Computer Society. ISBN 978-1-4673-8391-2. doi: 10.1109/ICCV.2015.123. URL http://dx.doi.org/10.1109/ICCV.2015.123.

[50] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[51] G. Papandreou, I. Kokkinos, and P.-A. Savalle. Modeling local and global deformations in deep learning: Epitomic convolution, multiple instance learning, and sliding window detection. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 390–399. IEEE, June 2015.

[52] K.-K. Maninis, J. Pont-Tuset, P. Arbelaez, and L. Van Gool. Convolutional oriented boundaries: From image segmentation to High-Level tasks. *IEEE transactions on pattern analysis and machine intelligence*, May 2017.

[53] O. Ronneberger, P. Fischer, and T. Brox. U-Net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241. Springer International Publishing, 2015.

[54] V. Badrinarayanan, A. Kendall, and R. Cipolla. SegNet: A deep convolutional Encoder-Decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(12):2481–2495, December 2017.

[55] O. Ben-Shahar and S. Zucker. Geometrical computations explain projection patterns of long-range horizontal connections in visual cortex. *Neural computation*, 16(3):445–476, March 2004.

[56] S. Shushruth, L. Nurminen, M. Bijanzadeh, J. M. Ichida, S. Vanni, and A. Angelucci. Different orientation tuning of near- and far-surround suppression in macaque primary visual cortex mirrors their tuning in human perception. *The Journal of neuroscience: the official journal of the Society for Neuroscience*, 33(1):106–119, January 2013.

[57] H. Tanaka and I. Ohzawa. Surround suppression of V1 neurons mediates orientation-based representation of high-order visual features. *Journal of neurophysiology*, 101(3):1444–1462, March 2009.

[58] J. Kim, M. Ricci, and T. Serre. Not-So-CLEVR: learning same-different relations strains feedforward neural networks. *Interface focus*, 8(4):20180011, August 2018.

[59] C. Szegedy, Google Inc, W. Zaremba, I. Sutskever, Google Inc, J. Bruna, D. Erhan, Google Inc, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations (ICLR)*, 2014.

[60] A. Nguyen, J. Yosinski, and J. Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 427–436, 2015.

[61] A. Volokitin, G. Roig, and T. A. Poggio. Do deep neural networks suffer from crowding? In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5628–5638. Curran Associates, Inc., 2017.

[62] K. Ellis, A. Solar-Lezama, and J. Tenenbaum. Unsupervised learning by program synthesis. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 973–981. Curran Associates, Inc., 2015.

[63] P. R. Roelfsema and R. Houtkamp. Incremental grouping of image elements in vision. *Attention, perception & psychophysics*, 73(8):2542–2572, November 2011.

[64] C. D. Gilbert and W. Li. Top-down influences on visual processing. *Nature reviews. Neuroscience*, 14(5):350–363, May 2013.