Cite as: D. George *et al*., *Science* 10.1126/science.aag2612 (2017).

A generative vision model that trains with high data efficiency and breaks text-based CAPTCHAs

D. George,* W. Lehrach, K. Kansky, M. Lázaro-Gredilla,* C. Laan, B. Marthi, X. Lou, Z. Meng, Y. Liu, H. Wang, A. Lavin, D. S. Phoenix

Vicarious AI, 2 Union Square, Union City, CA 94587, USA.

*Corresponding author. Email: dileep@vicarious.com (D.G.); miguel@vicarious.com (M.L.-G.)

Learning from few examples and generalizing to dramatically different situations are capabilities of human visual intelligence that are yet to be matched by leading machine learning models. By drawing inspiration from systems neuroscience, we introduce a probabilistic generative model for vision in which message-passing based inference handles recognition, segmentation and reasoning in a unified way. The model demonstrates excellent generalization and occlusion-reasoning capabilities, and outperforms deep neural networks on a challenging scene text recognition benchmark while being 300-fold more data efficient. In addition, the model fundamentally breaks the defense of modern text-based CAPTCHAs by generatively segmenting characters without CAPTCHA-specific heuristics. Our model emphasizes aspects like data efficiency and compositionality that may be important in the path toward general artificial intelligence.

The ability to learn and generalize from a few examples is a hallmark of human intelligence (*1*). CAPTCHAs, images used by websites to block automated interactions, are examples of problems that are easy for humans but difficult for computers. CAPTCHAs are hard for algorithms because they add clutter and crowd letters together to create a chicken-and-egg problem for character classifiers — the classifiers work well for characters that have been segmented out, but segmenting the individual characters requires an understanding of the characters, each of which might be rendered in a combinatorial number of ways (*2*–*5*). A recent deep-learning approach for parsing one specific CAPTCHA style required millions of labeled examples from it (*6*), and earlier approaches mostly relied on hand-crafted style-specific heuristics to segment out the character (*3*, *7*); whereas humans can solve new styles without explicit training (Fig. 1A). The wide variety of ways in which letterforms could be rendered and still be understood by people is illustrated in Fig. 1.

Building models that generalize well beyond their training distribution is an important step toward the flexibility Douglas Hofstadter envisioned when he said that "for any program to handle letterforms with the flexibility that human beings do, it would have to possess full-scale artificial intelligence" (*8*). Many researchers have conjectured that this could be achieved by incorporating the inductive biases of the visual cortex (*9*–*12*), utilizing the wealth of data generated by neuroscience and cognitive science research. In the mammalian brain, feedback connections in the visual cortex play roles in figure-ground-segmentation, and in object-based top-down attention that isolates the contours of an object even when partially transparent objects occupy the same spatial locations (*13*–*16*). Lateral connections in the visual cortex are implicated in enforcing contour continuity (*17*, *18*). Contours and surfaces are represented using separate mechanisms that interact (*19*–*21*), enabling the recognition and imagination of objects with unusual appearance – for example a chair made of ice. The timing and topography of cortical activations give clues about contour-surface representations and inference algorithms (*22*, *23*). These insights based on cortical function are yet to be incorporated into leading machine learning models.

We introduce a hierarchical model called the Recursive Cortical Network (RCN) that incorporates these neuroscience insights in a structured probabilistic generative model framework (*5*, *24*–*27*).

In addition to developing RCN and its learning and inference algorithms, we applied the model to a variety of visual cognition tasks that required generalizing from one or a few training examples: parsing of CAPTCHAs, one-shot and few-shot recognition and generation of handwritten digits, occlusion reasoning, and scene text recognition. We then compared its performance to state of the art models.

Recursive cortical network

RCN builds on existing compositional models (*24*, *28*–*32*) in important ways [section 6 of (*33*)]. Although grammar based models (*24*) have the advantage of being based on well-known ideas from linguistics, they either limit interpretations to single trees or are computationally infeasible when using attributed relations (*32*). The seminal work on AND-OR

templates and tree-structured compositional models (*34*) has the advantage of simplified inference, but is lacking in selectivity owing to the absence of lateral constraints (*35*). Models from another important class (*25*, *29*) use lateral constraints, but rather than gradually building invariance through a pooling structure (*36*), they use parametric transformations for complete scale, rotation and translation invariance at each level. Custom inference algorithms are required, but those are not effective in propagating the effect of lateral constraints beyond local interactions. The representation of contours and surfaces in (*37*) do not model their interactions, choosing instead to model these as independent mechanisms. RCNs and Composition Machines (CM) (*32*) share the motivation of placing compositional model ideas in a graphical model formulation. However, CM's representational choice of "composed distributions" — using a single layer of random variables to collapse feature-detection, pooling and lateral coordination — leads to an expanded state space, which in turn constrains the model to a greedy inference and parsing process. In general, because of the varied and conflicting representational choices, inference in compositional models has relied on custom-crafted methods for different model instantiations, including solving stochastic partial differential equations (*30*), sampling based algorithms (*24*), and pruned dynamic programming (*29*).

RCN integrates and builds upon various ideas from compositional models — hierarchical composition, gradual building of invariances, lateral connections for selectivity, contour-surface factorization and joint-explanation based parsing — into a structured probabilistic graphical model such that Belief Propagation (*38*) can be used as the primary approximate inference engine [section 6 of (*33*)]. Experimental neuroscience data provided important guidance on the representational choices [section 7 of (*33*)], which were then confirmed to be beneficial using experimental studies. We now discuss the representation of RCN and its inference and learning algorithms. Mathematical details are discussed in sections 2 to 5 of (*33*).

Representation

In RCN, objects are modeled as a combination of contours and surfaces (Fig. 2A). Contours appear at the boundaries of surfaces, both at the outline of objects and at the border between the surfaces that compose the object. Surfaces are modeled using a Conditional Random Field (CRF) which captures the smoothness of variations of surface properties. Contours are modeled using a compositional hierarchy of features (*28*, *39*). Factored representation of contours (shape) and surfaces (appearance) enables the model to recognize object shapes with dramatically different appearances without training exhaustively on every possible shape and appearance combination. We now describe the shape and appearance representations in detail.

Figure 2B shows two subnetworks (black and blue) within a level of the RCN contour hierarchy. The filled and empty circular nodes in the graph are binary random variables that correspond to features and pools respectively. Each feature node encodes an AND relation of its child pools, and each pool variable encodes the OR of its child features, similar to AND-OR graphs (*34*). Lateral constraints, represented as rectangular "factor nodes", coordinate the choices between the pools they connect to. The two subnetworks, which can correspond to two objects or object parts, share lower level-features.

Figure 2C shows a three-level network that represents the contours of a square. The features at the lowest, intermediate and top levels respectively represent line segments, corners and the entire square. Each pool variable pools over different deformations, small translations, scale changes etc., of a "centered" feature, thus introducing the corresponding invariances. Without the lateral connections between the pools (the gray squares in Fig. 2C), generating from a feature node representing a corner can create misaligned line segments, as shown in Fig. 3A. Lateral connections between the pools provide selectivity (*35*) by ensuring that the choice of a feature in one pool affects the choice of features in pools it is connected to, creating samples where the contours vary more smoothly. The flexibility of lateral constraints is controlled through perturb-factor, a hyperparameter that is specified per level. Through multiple layers of feature pooling, lateral connections, and compositions, a feature node at the top level comes to represent an object that can be recognized with some level of translation, scale and deformation invariance.

Multiple objects are represented in the same shape hierarchy by sharing their parts (Fig. 2B). When multiple parents converge on a single child feature (feature node "e" in Fig. 2B), this will be active when any parent is active (OR-gate in the graphical model), and the child feature is allowed to be part of both parents if evidence allows, unlike the exclusive sharing in AND-OR graph grammars (*24*). Even when two higher-level features share some of the same lower-level features and pools, the higher-level features' lateral networks are kept separate by making copies of the lower-level feature for each specific higher-level feature it participates in, as shown in Fig. 2B. Parent-specific copies of lateral networks serve to achieve higher-order interactions compared to pairwise connections, similar to the state copying mechanism used in higher-order networks (*40*). This was also found to be important for message-passing to achieve accurate results and is reminiscent of techniques used in dual decomposition (*41*). Hierarchy in the RCN network plays two roles. First, it enables the representation of deformations gradually through multiple levels, spreading the amount of variation across layers (Fig. 3B). Second, hierarchy provides efficiency through

the sharing of features between different objects (*42*). Both of these result in efficient learning and inference through shared computations.

Surfaces are modeled using a pairwise CRF (Fig. 3C). Local surface patch properties like color, texture, or surface normal are represented by categorical variables, whose smoothness of variation is enforced by the lateral factors (gray squares in Fig. 2). Contours generated by the contour-hierarchy interact with the surface CRF in a specific way: contours signal the breaks in continuity of surfaces that occur both within an object and between the object and its background, a representational choice inspired by neurobiology (*19*). Figure 3, B and D, shows samples generated from an RCN.

Inference

In order to parse a scene, RCN maintains hierarchical graphs for multiple object instances at multiple locations tiling the scene. The parse of a scene can be obtained via maximum a posteriori (MAP) inference on this complex graph, which recovers the best joint configuration including object identities and their segmentations [section 4 of (*33*)]. Although the RCN network is extremely loopy, we found that message-passing (*38*), with a schedule that is inspired by the timing of activations in the visual cortex (*9*, *20*), resulted in fast and accurate inference. An input image is first passed through PreProc, which converts pixel values to edge likelihoods using a bank of Gabor-like filters. Partial assignments that correspond to object hypotheses are then identified using a forward and backward message passing in the network, and a complete approximate MAP solution is found by solving the scene-parsing problem on the graph of object hypotheses (Fig. 4). The forward pass gives an upper-bound on the log-probability of the nodes at the top level. The backward pass visits the high-scoring forward-pass hypotheses one by one, in a manner similar to a top-down attention process (*43*, *44*), running a conditional inference that assumes that all other nodes are off to find an approximate MAP configuration for the object (Fig. 4A). The backward pass can reject many object hypotheses that were falsely identified in the forward pass.

The global MAP configuration is a subset of all the object hypotheses generated from the forward and backward passes. The number of objects in the scene is inferred as part of this MAP solution. In addition to searching over an exponentially large number of subsets, finding the global MAP requires reasoning about high-order interactions between different hypotheses. We developed an approximate dynamic programming (DP) method that solves this in linear time. The DP algorithm exploits the fact that each object hypothesis occupies a contiguous region that can be represented as a 2d mask on the input image. By considering combinations of object hypotheses, i.e., parses, that produce spatially contiguous masks when their 2d-masks overlap, we create a topological ordering of the parses by sorting them according to masks that are contained in other masks. This results in a recursive computation of the score where only a linear number of candidate parses need to be evaluated in searching for the best parse. See section 4.7 of (*33*) for more details.

Learning

Features and lateral connections up to the penultimate level of the network are trained unsupervised using a generic 3D object data set that is task agnostic and rendered only as contour images. The resulting learned features vary from simple line segments at the lower levels to curves and corners at the higher levels.

Consider a partially learned model, where new features are being learned at level *k*, where features up to level *k-1* have already been learned and finalized, and a few features have been learned at level *k* (Fig. 4B). When a training image is presented, the first step is to find a MAP explanation for the contours of that image using the existing features at level *k*. This is identical to the inference problem described earlier of finding the MAP solution for a scene. The contours that remain unexplained are parsed using features at level *k-1*, and new features are proposed from their contour-continuous conjunctions. Repeating this process for all the training images accumulates counts on the usage of different features at level *k*, and the final features for this level are selected by optimizing an objective function that balances compression and reconstruction error (*31*). The same process is repeated level-by-level [see section 5.1 of (*33*)].

The lateral graph structure, which specifies the connectivity between pool pairs, is learned from the contour connectivity of input images. At the first pooling level, pools with features that are adjacent in the input contours are connected with each other. This process is repeated recursively in the hierarchy where lateral connections at the higher levels are inferred from adjacency in the lower-level graphs.

Features at the topmost level represent whole objects. These are obtained by finding the MAP configuration of a new object up to the penultimate level of the network, connecting pool pairs at the penultimate level according to the contour continuity of the input object, and then storing the conjunction of activations at the penultimate level as a feature in the top-most level. See section 5 of (*33*) for details.

Once the set of lower-level features and lateral connections are trained, they can be used for different domains by tuning a few hyper-parameters [section 8.3 of (*33*)]. The filter scales in the PreProc are chosen depending on the image and object size, and the flexibility of the lateral connections is set to match the distortions in the data. In addition, the features

at the lowest level have a "smoothing parameter" that sets an estimate on the probability that an edge pixel is ON owing to noise. This parameter can be set according to the noise levels in a domain.

Results

A CAPTCHA is considered broken if it can be automatically solved at a rate above 1% (*3*). RCN was effective in breaking a wide variety of text-based CAPTCHAs with very little training data, and without using CAPTCHA-specific heuristics (Fig. 5). It was able to solve reCAPTCHAs at an accuracy rate of 66.6% (character level accuracy of 94.3%), BotDetect at 64.4%, Yahoo at 57.4% and PayPal at 57.1%, significantly above the 1% rate at which CAPTCHAs are considered ineffective (*3*). The only differences in architecture across different CAPTCHA tasks are the sets of clean fonts used for training and the different choices of a few hyper-parameters, which depend on the size of the CAPTCHA image and the amount of clutter and deformations. These parameters are straightforward to set by hand, or can be tuned automatically via cross validation on an annotated CAPTCHA set. Noisy, cluttered and deformed examples from the CAPTCHAs were not used for training, yet RCN was effective in generalizing to those variations.

For reCAPTCHA parsing at 66.6% accuracy, RCN required only five clean training examples per character. The model uses three parameters that affect how single characters are combined together to read out a string of characters, and these parameters were both independent of the length of the CAPTCHAs and were robust to the spacing of the characters [Fig. 5B and section 8.4 of (*33*)]. In addition to obtaining a transcription of the CAPTCHA, the model also provides a highly accurate segmentation into individual characters, as shown in Fig. 5A. To compare, human accuracy on reCAPTCHA is 87.4%. Because many input images have multiple valid interpretations (Fig. 5A), parses from two humans agree only 81% of the time.

In comparison to RCNs, a state-of-the-art CNN (*6*) required a 50,000-fold larger training set of actual CAPTCHA strings, and it was less robust to perturbations to the input. Because the CNN required a large number of labeled examples, this control study used a CAPTCHA-generator that we created to emulate the appearance of reCAPTCHAs [see section 8.4.3 of (*33*)]. The approach used a bank of position-specific CNNs, each trained to discriminate the letter at a particular position. Training the CNNs to achieve a word-accuracy rate of 89.9% required over 2.3 million unique training images, created using translated crops for data augmentation, from 79,000 distinct CAPTCHA words. The resulting network fails on string lengths not present during training, and more importantly, the recognition accuracy of the network deteriorates rapidly with even minor perturbations to the spacing of characters that are barely perceptible to humans – 15% more spacing reduced accuracy to 38.4%, and 25% more spacing reduced accuracy to just 7%. This suggests that the deep-learning method learned to exploit the specifics of a particular CAPTCHA rather than learning models of characters that are then used for parsing the scene. For RCN, increasing the spacing of the characters results in an improvement in the recognition accuracy (Fig. 5B).

The wide variety of character appearances in BotDetect (Fig. 5C) demonstrates why the factorization of contours and surfaces is important: models without this factorization could latch on to the specific appearance details of a font, thereby limiting their generalization. The RCN results are based on testing on 10 different styles of CAPTCHAs from BotDetect, all parsed based on a single network trained on 24 training example per character, and using the same parsing parameters across all styles. Although BotDetect CAPTCHAs can be parsed using contour information alone, using the appearance information boosted the accuracy from 61.8% to 64.4%, using the same appearance model across all data sets. See section 8.4.6 of (*33*) for more details.

RCN outperformed other models on one-shot and few-shot classification tasks on the standard MNIST handwritten digit data set [section 8.7 of (*33*)]. We compared RCN's classification performance on MNIST as we varied the number of training examples from 1 to 100 per category. CNN comparisons were made with two state-of-the art models, a LeNet-5 (*45*) and the VGG-fc6 CNN (*46*) with its levels pre-trained for ImageNet (*47*) classification using millions of images. The fully-connected-layer fc6 of VGG-CNN was chosen for comparison because it gave the best results for this task compared to other pre-trained levels of the VGG-CNN, and compared to other pre-trained CNNs that used the same data set and edge pre-processing as RCN [section 5.1 of (*33*)]. In addition, we compared against the Compositional Patch Model (*48*) that recently reported state-of-the-art performance on this task. RCN outperformed the CNNs and the CPM (Fig. 6A). The one-shot recognition performance of RCN was 76.6% vs 68.9% for CPM and 54.2% for VGG-fc6. RCN was also robust to different forms of clutter that were introduced during testing, without having to expose the network to those transformations during training. In comparison, such out-of-sample test examples had a large detrimental effect on the generalization performance of CNNs (Fig. 6B). To isolate the contributions of lateral connections, forward pass, and backward pass to RCN's accuracy, we conducted lesion studies that selectively turned off these mechanisms. The results, summarized in Fig. 6C, show that all these mechanisms contribute significantly toward the performance of RCNs. RCN networks with two levels of feature detection and pooling were sufficient to get the best accuracy performance on character parsing tasks.

The effect of increasing the number of levels in the hierarchy is to reduce the inference time as detailed in section 8.11 of (*33*).

As a generative model, RCN outperformed Variational Auto Encoders (VAE) (*49*) and DRAW (*50*) on reconstructing corrupted MNIST images (Fig. 7, A and B). DRAW's advantage over RCN for the clean test set is not surprising because DRAW is learning an overly flexible model that almost copies the input image in the reconstruction, which hurts its performance on more cluttered data sets [section 8.9 of (*33*)]. On the Omniglot data (*1*), examples generated from RCN after one-shot training showed significant variations, while still being identifiable as the original category [Fig. 7D and section 8.6 of (*33*)].

To test occlusion reasoning (*51*–*53*) we created a variant of the MNIST data set by adding a rectangle to each validation/test image such that some parts of the digit were occluded by the rectangle and some parts of the rectangle were occluded by the digit [Fig. 7C and section 8.8 of (*33*)]. Occlusion relationships in these images cannot be deduced as a simple layering of one object in front of the other. Classification on this data set is challenging because many parts of the digit are occluded by the rectangle, and because the rectangle acts as clutter. If the rectangle is detected and segmented out, its effect on the evidence for a particular digit can be explained away using the RCN generative model, thereby improving the accuracy of classification and segmentation. RCN was tested for classification accuracy and for occlusion reasoning on this challenging data set. Classification accuracy without explaining away was 47.0%. Explaining away the rectangle boosts the classification accuracy to 80.7%. In addition, RCN was used to parse the scene by reasoning about the occlusion relation between the rectangle and the digit. The model was successful at predicting the precise occlusion relations of the test image as shown in Fig. 7C, obtaining a mean intersection over union (IOU) of 0.353 measured over the occluded regions.

Last, RCN was tested on the ICDAR-13 Robust Reading data set (*54*), a benchmark for text recognition in real world images (Fig. 7E). For this test, we enhanced the parsing algorithm to include prior knowledge about n-gram and word statistics, and about geometric priors related to the layout of letters in a scene, which includes spacing, relative sizes and appearance consistency [see section 8.5 of (*33*)]. We compared our result against top participants of the ICDAR competition, and against a recent deep learning approach (*55*) (Table 1). The RCN model outperformed the top contender, PhotoOCR, by 1.9%, despite PhotoOCR using 7.9 million training images, whereas RCN used 1,406 training images selected using model-based clustering from 25,584 font images. RCN achieved better accuracy on this task while being 300 times more data efficient, in addition to providing a detailed segmentation of the characters (Fig. 7E) that the competing methods do not provide.

Discussion

Segmentation resistance, the primary defense of text-based CAPTCHAs, has been a general principle that enabled their automated generation (*2*, *3*). Although specific CAPTCHAs have been broken before using style-specific segmentation heuristics (*3*, *7*), those attacks could be foiled easily by minor alterations to CAPTCHAs. RCN breaks the segmentation defense in a fundamental way and with very little training data, which suggests that websites should move to more robust mechanisms for blocking bots.

Compositional models have been successfully used in the past for generic object recognition and scene parsing, and our preliminary experiments [section 8.12 of (*33*)] indicate that RCN could be applicable in those domains as well (Fig. 8). The RCN formulation opens up compositional models to a wider array of advanced inference and learning algorithms developed in graphical models, potentially leading to improvements that build on their prior successes in real-world scene parsing (*56*, *57*). Despite the advantage of being a generative model, RCN needs several improvements to achieve superior performance on ImageNet-scale (*47*) data sets. Flexible merging of multiple instances, the use of surface appearance at all levels of the hierarchy during forward and backward inference, more sophisticated pooling structures that learn to pool over 3D transformations, and generative modeling of scene context and background are problems that need to be investigated and integrated with RCN [section 8.13 of (*33*)].

The high data efficiency of RCN, compared to whole-image models like CNNs and VAEs, derives from the fact that RCN encodes strong assumptions in its structure. Recent neural networks models incorporate ideas of compositionality using a spatial attention window (*58*), but their current instantiations need good separation between the objects in an uncluttered setting because each attention window is modeled using a whole-image VAE. Incorporation of RCN's object and part-based compositionality into neural network models would be an interesting research direction. Unlike neural networks, the current version of RCN learning algorithms need clean training data, a drawback we intend to address using gradient based learning as well as message-passing based approaches (*59*).

Combining RCN with Bayesian Program Learning (BPL) (*1*) is another avenue for future investigations. BPL has the advantage of precisely modeling the sequential causal mechanisms, e.g., the stroke generation in the Omniglot data set, but its inference depends on the contours being separated from the background – something RCN can easily provide. More generally, BPL and RCN-like graphical models could be

combined to obtain the expressive power and efficient inference required to model the parallel and sequential processes (*60*) involved in perception and cognition.

Of course, Douglas Hofstadter's challenge – understanding letterforms with the same efficiency and flexibility of humans – still stands as a grand goal for artificial intelligence. People use a lot more commonsense knowledge, in context-sensitive and dynamic ways, when they identify letterforms (Fig. 1C, iii). Our works suggests that incorporating inductive biases from systems neuroscience can lead to robust, generalizable machine-learning models that demonstrate high data efficiency. We hope that this work inspires improved models of cortical circuits (*61*, *62*) and investigations that combine the power of neural networks and structured probabilistic models toward general artificial intelligence systems.

Methods summary

For reCAPTCHA experiments, we downloaded 5500 reCAPTCHA images from google.com reCAPTCHA page, of which 500 were used as validation set for parameter tuning, and accuracy numbers are reported on the remaining 5000. The images were scaled up by a factor of 2. A similar-looking font to those used in reCAPTCHA, Georgia, was identified by visual comparison from the fonts available on the local system. RCN was trained on a few rotations of the lowercase and uppercase characters from this font. Hyperparameters were optimized using the validation set. Human accuracy on the reCAPTCHA data set was estimated using Amazon Mechanical Turk (AMT) using U.S. based workers.

Emulated reCAPTCHA data sets, used to train the neural network for control experiments, were created using ImageMagick to produce distortions that are qualitatively similar to the original reCAPTCHA. The emulated data generator is used as an unlimited source to generate random batches for training the neural network. Neural network optimization was run for 80 epochs, where data are permuted at the start of every epoch; data were also augmented by random translations of up to 5 pixels in each cardinal direction per epoch.

Similar methods were used for BotDetect, PayPal and Yahoo CAPTCHAs. For BotDetect, we downloaded a data set of 50-100 images per CAPTCHA style for determining the parsing parameters and training setup, and another 100 images as a testing data set on which the network is not tuned. As training images for the system, we selected a series of fonts and scales from those available on the system by visually comparing a few examples of the BotDetect CAPTCHAs. The BotDetect test images were rescaled by a factor of 1.45. Parsing parameters were optimized using the validation set, and the transferability of the parsing parameters were tested by adapting the parameters for each style separately and then testing those parameters on the other styles.

For training RCN to parse ICDAR, we obtained 492 fonts from Google Fonts, which resulted in 25584 character training images. From this we selected a set of training images using an automated greedy font selection approach. We rendered binary images for all fonts and then used the resulting images of the same letter to train an RCN. This RCN is then used to recognize the exact images it was trained on, providing a compatibility score (between 0.0 and 1.0) for all pairs of fonts of the same letter. Finally, using a threshold (=0.8) as the stopping criterion, we greedily select the most representative fonts until 90% of all fonts are represented, which resulted in 776 unique training images. The parser is trained using 630 word images and the character ngrams are trained using words from the Wikipedia.

RCN classification experiments on the MNIST data set are done by up-sampling the images by a factor of 4. For each training setting, two pooling hyperparameters of the model were adapted using an independent validation set of rotated MNIST digits. Several ways of pre-training the CNN are explored as part of the baselines. To understand the performance of the networks on noisy MNIST data, we created six variants of noise, each one with three levels of severity. For occlusion reasoning, the RCN network was trained with 11 categories: ten MNIST digit categories with 20 examples for category and the rectangular ring category with one example. Reconstruction experiments on the MNIST data set used networks that were trained only on clean MNIST images which were then tested for mean squared reconstruction error on 6 different noise variants, each with 3 levels of severity. Full methods are available in supplemental materials.

REFERENCES AND NOTES

1. B. M. Lake, R. Salakhutdinov, J. B. Tenenbaum, Human-level concept learning through probabilistic program induction. *Science* **350**, 1332–1338 (2015). doi:10.1126/science.aab3050 Medline
2. K. Chellapilla, P. Simard, "Using machine learning to break visual human interaction proofs (HIPs)," in *Advances in Neural Information Processing Systems 17)* (2004) pp. 265–272.
3. E. Bursztein, M. Martin, J. C. Mitchell, "Text-based CAPTCHA strengths and weaknesses," in *Proceedings of the 18th ACM Conference on Computer and Communications Security* (ACM, 2011), pp. 125–138.
4. G. Mori, J. Malik, "Recognizing objects in adversarial clutter: Breaking a visual CAPTCHA," in *2003 IEEE Conference on Computer Vision and Pattern Recognition* (IEEE Computer Society, 2003), pp. I-134–I-141.
5. V. Mansinghka, T. D. Kulkarni, Y. N. Perov, J. Tenenbaum, "Approximate bayesian image interpretation using generative probabilistic graphics programs," in *Advances in Neural Information Processing Systems 26* (2013), pages 1520–1528.
6. I. Goodfellow, Y. Bulatov, J. Ibarz, S. Arnoud, V. Shet, "Multi-digit number recognition from street view imagery using deep convolutional neural networks," paper presented at the International Conference on Learning Representations (ICLR) 2014, Banff, Canada, 14 to 16 April 2014.
7. E. Bursztein, J. Aigrain, A. Moscicki, J. C. Mitchell, "The End is nigh: Generic Solving of text-based CAPTCHAs," paper presented at the 8th USENIX Workshop on Offensive Technologies (WOOT '14), San Diego, CA, 19 August 2014.
8. D. R. Hofstadter, *Metamagical Themas: Questing for the Essence of Mind and Pattern* (Basic Books, 1985).
9. T. S. Lee, D. Mumford, Hierarchical Bayesian inference in the visual cortex. *JOSA A* **20**, 1434–1448 (2003). doi:10.1364/JOSAA.20.001434 Medline
10. T. L. Griffiths, N. Chater, C. Kemp, A. Perfors, J. B. Tenenbaum, Probabilistic

models of cognition: Exploring representations and inductive biases. *Trends Cogn. Sci.* **14**, 357–364 (2010). doi:10.1016/j.tics.2010.05.004 Medline
11. T. S. Lee, "The visual system's internal model of the world," in *Proceedings of the IEEE*, vol. 103 (IEEE, 2015), pp. 1359–1378
12. D. Kersten, A. Yuille, Bayesian models of object perception. *Curr. Opin. Neurobiol.* **13**, 150–158 (2003). doi:10.1016/S0959-4388(03)00042-4 Medline
13. C. D. Gilbert, W. Li, Top-down influences on visual processing. *Nat. Rev. Neurosci.* **14**, 350–363 (2013). doi:10.1038/nrn3476 Medline
14. V. A. F. Lamme, P. R. Roelfsema, The distinct modes of vision offered by feedforward and recurrent processing. *Trends Neurosci.* **23**, 571–579 (2000). doi:10.1016/S0166-2236(00)01657-X Medline
15. P. R. Roelfsema, V. A. F. Lamme, H. Spekreijse, Object-based attention in the primary visual cortex of the macaque monkey. *Nature* **395**, 376–381 (1998). doi:10.1038/26475 Medline
16. E. H. Cohen, F. Tong, Neural mechanisms of object-based attention. *Cereb. Cortex* **25**, 1080–1092 (2015). doi:10.1093/cercor/bht303 Medline
17. D. J. Field, A. Hayes, R. F. Hess, Contour integration by the human visual system: Evidence for a local "association field". *Vision Res.* **33**, 173–193 (1993). doi:10.1016/0042-6989(93)90156-Q Medline
18. C. D. Gilbert, T. N. Wiesel, Columnar specificity of intrinsic horizontal and corticocortical connections in cat visual cortex. *J. Neurosci.* **9**, 2432–2442 (1989). Medline
19. E. Craft, H. Schütze, E. Niebur, R. von der Heydt, A neural model of figure-ground organization. *J. Neurophysiol.* **97**, 4310–4326 (2007). doi:10.1152/jn.00203.2007 Medline
20. V. A. F. Lamme, V. Rodriguez-Rodriguez, H. Spekreijse, Separate processing dynamics for texture elements, boundaries and surfaces in primary visual cortex of the macaque monkey. *Cereb. Cortex* **9**, 406–413 (1999). doi:10.1093/cercor/9.4.406 Medline
21. E. A. DeYoe, D. C. Van Essen, Concurrent processing streams in monkey visual cortex. *Trends Neurosci.* **11**, 219–226 (1988). doi:10.1016/0166-2236(88)90130-0 Medline
22. X. Huang, M. A. Paradiso, V1 response timing and surface filling-in. *J. Neurophysiol.* **100**, 539–547 (2008). doi:10.1152/jn.00997.2007 Medline
23. H. Zhou, H. S. Friedman, R. von der Heydt, Coding of border ownership in monkey visual cortex. *J. Neurosci.* **20**, 6594–6611 (2000). Medline
24. S.-C. Zhu, D. Mumford, *A Stochastic Grammar of Images* (Now Publishers, 2007).
25. L. L. Zhu, Y. Chen, A. Yuille, Recursive compositional models for vision: Description and review of recent work. *J. Math. Imaging Vis.* **41**, 122–146 (2011). doi:10.1007/s10851-011-0282-2
26. A. L. Yuille, "Towards a theory of compositional learning and encoding of objects," in *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)* (IEEE, 2011), pp. 1448–1455.
27. R. Salakhutdinov, J. B. Tenenbaum, A. Torralba, "Learning to learn with compound HD models," in *Advances in Neural Information Processing Systems 24* (2012), pp. 1–9.
28. Y. Chen, L. Zhu, C. Lin, A. Yuille, H. Zhang, "Rapid inference on a novel AND/OR graph for object detection, segmentation and parsing," in *Advances in Neural Information Processing Systems 20* (2007), pp. 289–296.
29. L. Zhu, A. Yuille, "Recursive compositional models: Representation, learning, and inference," In *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (IEEE, 2009), p. 5.
30. Z. Tu, X. Chen, A. L. Yuille, S.-C. Zhu, Image parsing: Unifying segmentation, detection, and recognition. *Int. J. Comput. Vis.* **63**, 113–140 (2005). doi:10.1007/s11263-005-6642-x
31. S. Fidler, A. Leonardis, "Towards scalable representations of object categories: Learning a hierarchy of parts," in *2007 IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (IEEE, 2007).
32. Y. Jin, S. Geman, "Context and hierarchy in a probabilistic image model," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2 (IEEE, 2006), pp. 2145–2152.
33. Supplementary materials.
34. Z. Si, S.-C. Zhu, Learning AND-OR templates for object recognition and detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **35**, 2189–2205 (2013).
35. S. Geman, Invariance and selectivity in the ventral visual pathway. *J. Physiol. Paris* **100**, 212–224 (2006). doi:10.1016/j.jphysparis.2007.01.001 Medline
36. T. Serre, L. Wolf, S. Bileschi, M. Riesenhuber, T. Poggio, Robust object recognition with cortex-like mechanisms. *IEEE Trans. Pattern Anal. Mach. Intell.* **29**, 411–426 (2007). doi:10.1109/TPAMI.2007.56 Medline
37. C. Guo, S.-C. Zhu, Y. N. Wu, Primal sketch: Integrating structure and texture. *Comput. Vis. Image Underst.* **106**, 5–19 (2007). doi:10.1016/j.cviu.2005.09.004
38. J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference* (Morgan Kaufmann, 1988).
39. T. Wu, S.-C. Zhu, A numerical study of the bottom-up and top-down inference processes in and-or graphs. *Int. J. Comput. Vis.* **93**, 226–252 (2011). doi:10.1007/s11263-010-0346-6
40. J. Xu, T. L. Wickramarathne, N. V. Chawla, Representing higher-order dependencies in networks. *Sci. Adv.* **2**, e1600028 (2016). doi:10.1126/sciadv.1600028 Medline
41. D. Sontag, A. Globerson, T. Jaakkola, "Introduction to dual decomposition for inference," in *Optimization for Machine Learning* (MIT Press, 2010), pp. 1–37.
42. E. Bienenstock, S. Geman, D. Potter, "Compositionality, MDL priors, and object recognition," in *Advances in Neural Information Processing Systems 10* (1997), pp. 838–844.
43. J. Tsotsos, A. Rothenstein, Computational models of visual attention. *Scholarpedia* **6**, 6201 (2011). doi:10.4249/scholarpedia.6201
44. B. A. Olshausen, C. H. Anderson, D. C. Van Essen, A neurobiological model of visual attention and invariant pattern recognition based on dynamic routing of information. *J. Neurosci.* **13**, 4700–4719 (1993). Medline
45. Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition. *Proc. IEEE* **86**, 2278–2324 (1998). doi:10.1109/5.726791
46. K. Simonyan, A. Zisserman, "Very deep convolutional networks for large-scale image recognition," paper presented at the International Conference on Learning Representations (ICLR) 2015, San Diego, CA, 7 to 9 May 2015.
47. J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, F.-F. Li, "Imagenet: A large-scale hierarchical image database," in *IEEE Conference on Computer Vision and Pattern Recognition* (IEEE, 2009), pp. 248–255.
48. A. Wong, A. L. Yuille, "One shot learning via compositions of meaningful patches," in *2015 IEEE International Conference on Computer Vision* (IEEE, 2015), pp. 1197–1205.
49. D. P. Kingma, M. Welling, "Stochastic gradient VB and the variational auto-encoder," paper presented at the International Conference on Learning Representations (ICLR) 2014, Banff, Canada, 14 to 16 April 2014.
50. K. Gregor, I. Danihelka, A. Graves, D. J. Rezende, D. Wierstra, "DRAW: A recurrent neural network for image generation," in *Proceedings of the 32nd International Conference on Machine Learning* (Proceedings of Machine Learning Research, 2015), pp. 1462–1471.
51. C. K. Williams, M. K. Titsias, Greedy learning of multiple objects in images using robust statistics and factorial learning. *Neural Comput.* **16**, 1039–1062 (2004). doi:10.1162/089976604773135096 Medline
52. T. Gao, B. Packer, D. Koller, "A segmentation-aware object detection model with occlusion handling," in *2011 IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (IEEE, 2011), pp. 1361–1368.
53. R. Fransens, C. Strecha, L. Van Gool, "A mean field EM-algorithm for coherent occlusion handling in MAP-estimation problems," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1 (IEEE, 2006) pp. 300–307.
54. D. Karatzas, F. Shafait, S. Uchida, M. Iwamura, L. Gomez i Bigorda, S. R. Mestre, J. Mas, D. F. Mota, J. A. Almazan, L.-P. de las Heras, "ICDAR 2013 Robust Reading Competition," in *2013 12th International Conference on Document Analysis and Recognition* (IEEE, 2013), pp. 1484–1493.
55. M. Jaderberg, K. Simonyan, A. Vedaldi, A. Zisserman, "Deep structured output learning for unconstrained text recognition," paper presented at the International Conference on Learning Representations (ICLR) 2015, San Diego, CA, 7 to 9 May 2015.
56. J. Wang, A. Yuille, "Semantic Part segmentation using compositional model combining shape and appearance," in *2015 IEEE Conference on Computer Vision and Pattern Recognition* (IEEE, 2015), pp. 1788–1797.
57. D. Tabernik, A. Leonardis, M. Boben, D. Skočaj, M. Kristan, Adding discriminative power to a generative hierarchical compositional model using histograms of

compositions. *Comput. Vis. Image Underst.* **138**, 102–113 (2015). doi:10.1016/j.cviu.2015.04.006
58. S. M. Ali Eslami, N. Heess, T. Weber, Y. Tassa, K. Kavukcuoglu, "Attend, infer, repeat: Fast scene understanding with generative models," paper presented at the 30th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain, 5 to 10 December 2016.
59. M. L'azaro-Gredilla, Y. Liu, D. S. Phoenix, D. George, "Hierarchical compositional feature learning," arXiv:1611.02252 [cs.LG] (7 November 2016).
60. S. Ullman, Visual routines. *Cognition* **18**, 97–159 (1984). doi:10.1016/0010-0277(84)90023-4 Medline
61. S. Litvak, S. Ullman, Cortical circuitry implementing graphical models. *Neural Comput.* **21**, 3010–3056 (2009). doi:10.1162/neco.2009.05-08-783 Medline
62. D. George, J. Hawkins, Towards a mathematical theory of cortical micro-circuits. *PLOS Comput. Biol.* **5**, e1000532 (2009). doi:10.1371/journal.pcbi.1000532 Medline
63. N. Le Roux, N. Heess, J. Shotton, J. Winn, Learning a generative model of images by factoring appearance and shape. *Neural Comput.* **23**, 593–650 (2011). doi:10.1162/NECO_a_00086 Medline
64. S. Eslami, C. Williams, "A generative model for parts-based object segmentation," in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, K. Q. Weinberger, Eds. (Curran Associates, 2012), pp. 100–107.
65. A. Krizhevsky, I. Sutskever, G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25* (2012), pp. 1097–1105.
66. A. Globerson, T. S. Jaakkola, "Fixing max-product: Convergent message passing algorithms for MAP LP-relaxations," in *Advances in Neural Information Processing Systems 20* (2008), pp. 553–560.
67. V. Kolmogorov, Convergent tree-reweighted message passing for energy minimization. *IEEE Trans. Pattern Anal. Mach. Intell.* **28**, 1568–1583 (2006).
68. H. Wang, K. Daphne, Subproblem-tree calibration: A unified approach to max-product message passing. In *Proceedings of the 30th International Conference on Machine Learning* (Proceedings of Machine Learning Research, 2013), pp. 190–198).
69. S. E. Shimony, Finding MAPs for belief networks is NP-hard. *Artif. Intell.* **68**, 399–410 (1994). doi:10.1016/0004-3702(94)90072-8
70. T. Werner, A linear programming approach to max-sum problem: A review. *IEEE Trans. Pattern Anal. Mach. Intell.* **29**, 1165–1179 (2007). doi:10.1109/TPAMI.2007.1036 Medline
71. N. Komodakis, N. Paragios, G. Tziritas, MRF energy minimization and beyond via dual decomposition. *IEEE Trans. Pattern Anal. Mach. Intell.* **33**, 531–552 (2011).
72. T. Meltzer, A. Globerson, Y. Weiss, "Convergent message passing algorithms – a unifying view," In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, J. A. Bilmes, A. Y. Ng, Eds. (AUAI Press, 2009), pp. 393–401
73. S. G. Mallat, Zhifeng Zhang, Matching pursuits with time-frequency dictionaries. *IEEE Trans. Signal Process.* **41**, 3397–3415 (1993). doi:10.1109/78.258082
74. S. Fidler, M. Boben, A. Leonardis, "Optimization framework for learning a hierarchical shape vocabulary for object class detection," in *BMVC 2009* (2009).
75. B. Li *et al*., "Shrec'12 track: Generic 3d shape retrieval," paper presented at the Eurographics Workshop on 3D Object Retrieval, Cagliari, Italy, 13 May 2012.
76. A. Yuille, R. Mottaghi, Complexity of representation and inference in compositional models with part sharing. *J. Mach. Learn. Res.* **17**, 1–28 (2016).
77. M. A. Fischler, R. A. Elschlager, The representation and matching of pictorial structures. *IEEE Trans. Comput.* **22**, 67–92 (1973).
78. P. F. Felzenszwalb, D. P. Huttenlocher, Pictorial Structures for Object Recognition. *Int. J. Comput. Vis.* **61**, 55–79 (2005). doi:10.1023/B:VISI.0000042934.15159.49
79. L. L. Zhu, Y. Chen, A. Torralba, W. Freeman, A. Yuille, "Part and appearance sharing: Recursive compositional models for multi-view multi-object detection," in *2010 IEEE Conference on Computer Vision and Pattern Recognition* (IEEE, 2010).
80. C. K. I. Williams, N. J. Adams, "DTs: Dynamic trees," in *Advances in Neural Information Processing Systems 12* (1999), pp. 634–640.
81. L. L. Zhu, C. Lin, H. Huang, Y. Chen, A. Yuille, "Unsupervised structure learning: Hierarchical recursive composition, suspicious coincidence and competitive exclusion," in *10th European Conference on Computer Vision* (Springer, 2008), pp. 759–773.
82. S. Ullman, Object recognition and segmentation by a fragment-based hierarchy. *Trends Cogn. Sci.* **11**, 58–64 (2007). doi:10.1016/j.tics.2006.11.009 Medline
83. L. Bottou, Y. Bengio, Y. Le Cun, "Global training of document processing systems using graph transformer networks," in *1997 IEEE Conference on Computer Vision and Pattern Recognition* (IEEE, 1997), pp. 489–494.
84. Z. Kourtzi, N. Kanwisher, Representation of perceived object shape by the human lateral occipital complex. *Science* **293**, 1506–1509 (2001). doi:10.1126/science.1061133 Medline
85. J. J. DiCarlo, D. Zoccolan, N. C. Rust, How does the brain solve visual object recognition? *Neuron* **73**, 415–434 (2012). doi:10.1016/j.neuron.2012.01.010 Medline
86. D. H. Hubel, T. N. Wiesel, Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *J. Physiol.* **160**, 106–154 (1962). doi:10.1113/jphysiol.1962.sp006837 Medline
87. L. Wiskott, T. J. Sejnowski, Slow feature analysis: Unsupervised learning of invariances. *Neural Comput.* **14**, 715–770 (2002). doi:10.1162/089976602317318938 Medline
88. R. D. S. Raizada, S. Grossberg, Towards a theory of the laminar architecture of cerebral cortex: Computational clues from the visual system. *Cereb. Cortex* **13**, 100–113 (2003). doi:10.1093/cercor/13.1.100 Medline
89. O. Ben-Shahar, S. Zucker, Geometrical computations explain projection patterns of long-range horizontal connections in visual cortex. *Neural Comput.* **16**, 445–476 (2004). doi:10.1162/089976604772744866 Medline
90. J. Hawkins, D. George, J. Niemasik, Sequence memory for prediction, inference and behaviour. *Philos. Trans. R. Soc. Lond. B Biol. Sci.* **364**, 1203–1209 (2009). doi:10.1098/rstb.2008.0322 Medline
91. K. Moutoussis, The physiology and psychophysics of the color-form relationship: A review. *Front. Psychol.* **6**, 1407 (2015). Medline
92. E. A. Lachica, P. D. Beck, V. A. Casagrande, Parallel pathways in macaque monkey striate cortex: Anatomically defined columns in layer III. *Proc. Natl. Acad. Sci. U.S.A.* **89**, 3566–3570 (1992). doi:10.1073/pnas.89.8.3566 Medline
93. F. Federer, J. M. Ichida, J. Jeffs, I. Schiessl, N. McLoughlin, A. Angelucci, Four projection streams from primate V1 to the cytochrome oxidase stripes of V2. *J. Neurosci.* **29**, 15455–15471 (2009). doi:10.1523/JNEUROSCI.1648-09.2009 Medline
94. C. W. Tyler, R. von der Heydt, "Contour-, surface-, and object-related coding in the visual cortex," in *Computer Vision: From Surfaces to 3D Objects* (Chapman and Hall/CRC, 2011), pp. 145–162.
95. F. T. Qiu, R. von der Heydt, Neural representation of transparent overlay. *Nat. Neurosci.* **10**, 283–284 (2007). doi:10.1038/nn1853 Medline
96. K. Friston, The free-energy principle: A unified brain theory? *Nat. Rev. Neurosci.* **11**, 127–138 (2010). doi:10.1038/nrn2787 Medline
97. D. J. Felleman, D. C. Van Essen, Distributed hierarchical processing in the primate cerebral cortex. *Cereb. Cortex* **1**, 1–47 (1991). doi:10.1093/cercor/1.1.1 Medline
98. A. M. Bastos, W. M. Usrey, R. A. Adams, G. R. Mangun, P. Fries, K. J. Friston, M. Andre, Canonical microcircuits for predictive coding. *Neuron* **76**, 695–711 (2013).
99. X. Lou, K. Kansky, W. Lehrach, C. C. Laan, B. Marthi, D. S. Phoenix, D. George, "Generative shape models: Joint text recognition and segmentation with very little training data," in *Advances in Neural Information Processing Systems 29* (2016).
100. L. von Ahn, B. Maurer, C. McMillen, D. Abraham, M. Blum, reCAPTCHA: Human-based character recognition via Web security measures. *Science* **321**, 1465–1468 (2008). doi:10.1126/science.1160379 Medline
101. K. He, X. Zhang, S. Ren, J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition* (IEEE, 2016), pp. 770–778.
102. M. Abadi *et al*., "TensorFlow: Large-scale machine learning on heterogeneous distributed systems." arXiv:1603.04467 [cs.DC] (14 March 2016).
103. I. Tsochantaridis, T. Hofmann, T. Joachims, Y. Altun, "Support vector machine learning for interdependent and structured output spaces," in *Proceedings of the Twenty-First International Conference on Machine Learning* (ACM, 2004), p. 104.
104. A. Bissacco, M. Cummins, Y. Netzer, H. Neven, "PhotoOCR: Reading text in uncontrolled conditions," in *2013 IEEE International Conference on Computer Vision*, (IEEE, 2013), pp. 785–792.
105. Z. Ghahramani, Probabilistic machine learning and artificial intelligence. *Nature* **521**, 452–459 (2015). doi:10.1038/nature14541 Medline
106. N. D. Goodman, J. B. Tenenbaum, T. Gerstenberg, "Concepts in a probabilistic

language of thought," in *The Conceptual Mind: New Directions in the Study of Concepts*, E. Margolis, S. Lawrence, Eds. (MIT Press, 2015), pp. 623–654.
107. H. Lee, R. Grosse, R. Ranganath, 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* (ACM, 2009), pp. 609–616.
108. D. Kingma, J. Ba, "Adam: A method for stochastic optimization," paper presented at the International Conference on Learning Representations (ICLR) 2015, San Diego, CA, 7 to 9 May 2015.

ACKNOWLEDGMENTS

We thank the anonymous reviewers for their careful review and helpful suggestions that greatly improved the manuscript. We thank B. Olshausen, T. Dean, B. Lake, and B. Jaros for suggesting improvements after reading early versions of this manuscript. We are grateful to A. Yuille and F.-F. Li for insightful discussions leading to this work. Data sets used in the paper are available for download at www.vicarious.com. The inventions described in this paper are protected by U.S. patents 9262698, 9373085, 9607262 and 9607263. As text-based CAPTCHAs are still widely used to protect websites, the scientific benefit of releasing the source code must be balanced with the potential for it to be used for circumventing protections that prevent automated interactions with websites. As a compromise, a simplified reference implementation of RCN algorithms for the MNIST data set is available at www.vicarious.com/science_rcn.

SUPPLEMENTARY MATERIALS

www.sciencemag.org/cgi/content/full/science.aag2612/DC1
Materials and Methods
Supplementary Text
Figs. S1 to S27
Tables S1 to S15
References (*63–108*)

2 June 2016; accepted 8 September 2017
Published online 26 October 2017
10.1126/science.aag2612

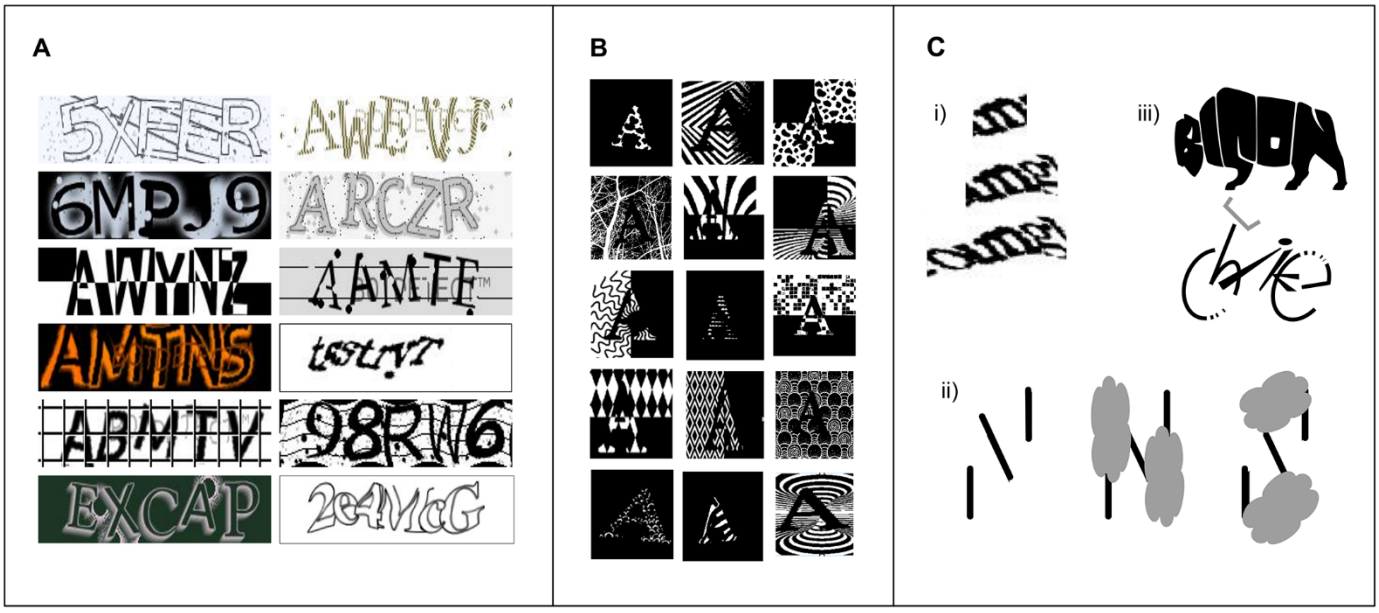


Fig. 1. Flexibility of letterform perception in humans. (**A**) Humans are good at parsing unfamiliar CAPTCHAs. (**B**) The same character shape can be rendered in a wide variety of appearances, and people can detect the "A" in these images regardless. (**C**) Common sense and context affect letterform perception: (i) m vs u and n. (ii) the same line segments are interpreted as N or S depending on occluder positions. (iii) perception of the shapes aids the recognition of "b,i,s,o,n" and "b,i,k,e". [Bison logo with permission from Seamus Leonard, http://www.steadynow.com]

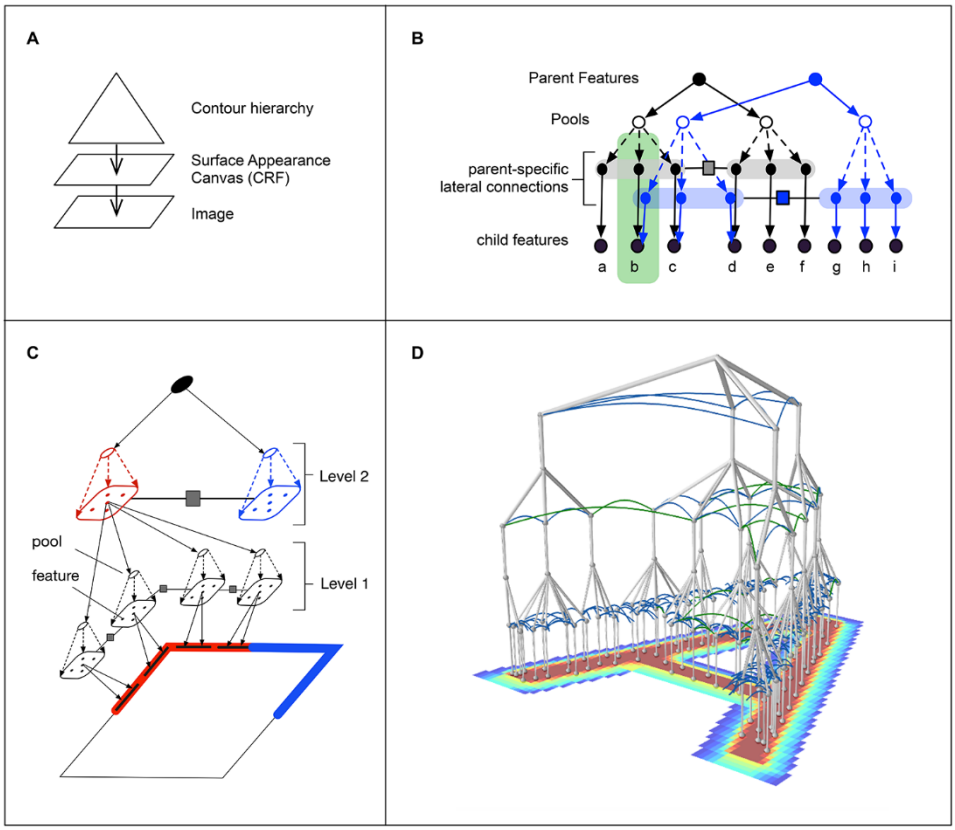


Fig. 2. Structure of the RCN. (**A**) A hierarchy generates the contours of an object, and a Conditional Random Field (CRF) generates its surface appearance. (**B**) Two subnetworks at the same level of the contour hierarchy keep separate lateral connections by making parent-specific copies of child features and connecting them with parent-specific laterals; nodes within the green rectangle are copies of the feature marked "e". (**C**) A three level RCN representing the contours of a square. Features at Level 2 represent the four corners, and each corner is represented as a conjunction of four line-segment features. (**D**) Four-level network representing an "A".

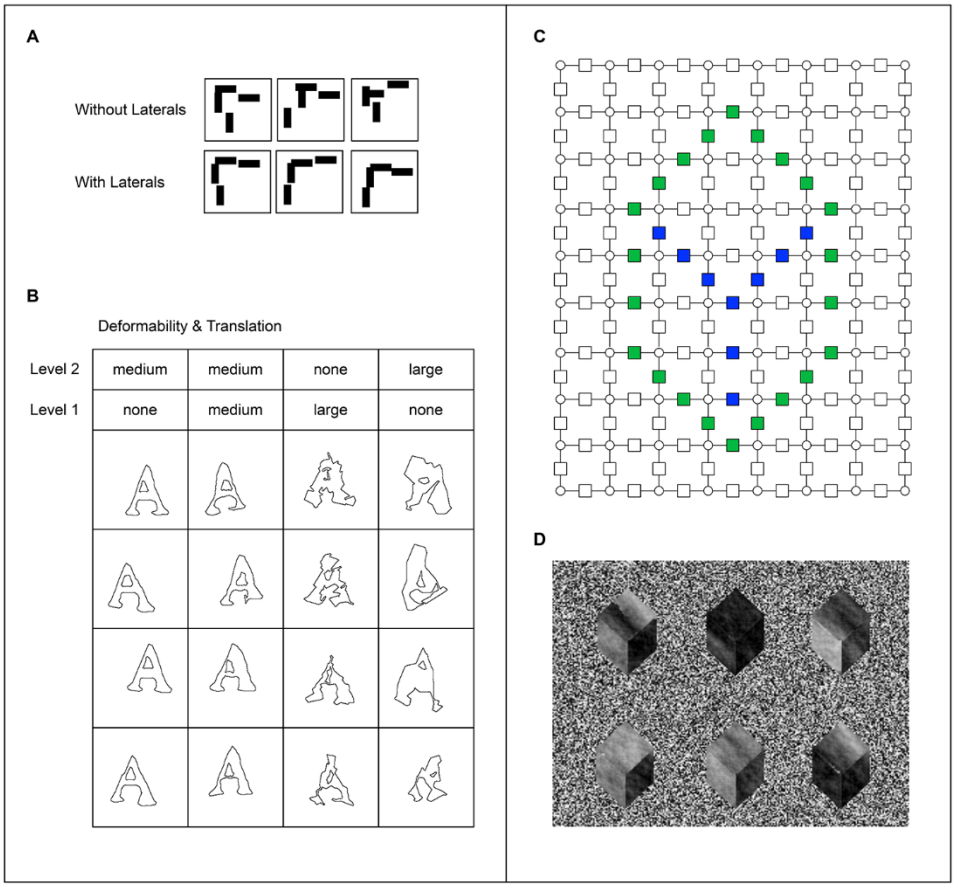


Fig. 3. Samples from RCN. (**A**) Samples from a corner feature with and without lateral connections. (**B**) Samples from character "A" for different deformability settings, determined by pooling and lateral perturb-factors, in a 3-level hierarchy similar to Fig. 2D, where the lowest level features are edges. Column 2 shows a balanced setting where deformability is distributed between the levels to produce local deformations and global translations. The other columns show some extreme configurations. (**C**) Contour to surface-CRF interaction for a cube. Green factors: foreground-to-background edges, blue: within-object edges. (**D**) Different surface-appearance samples for the cubical shape in C. [See section 3 of (*33*) for CRF parameters.]

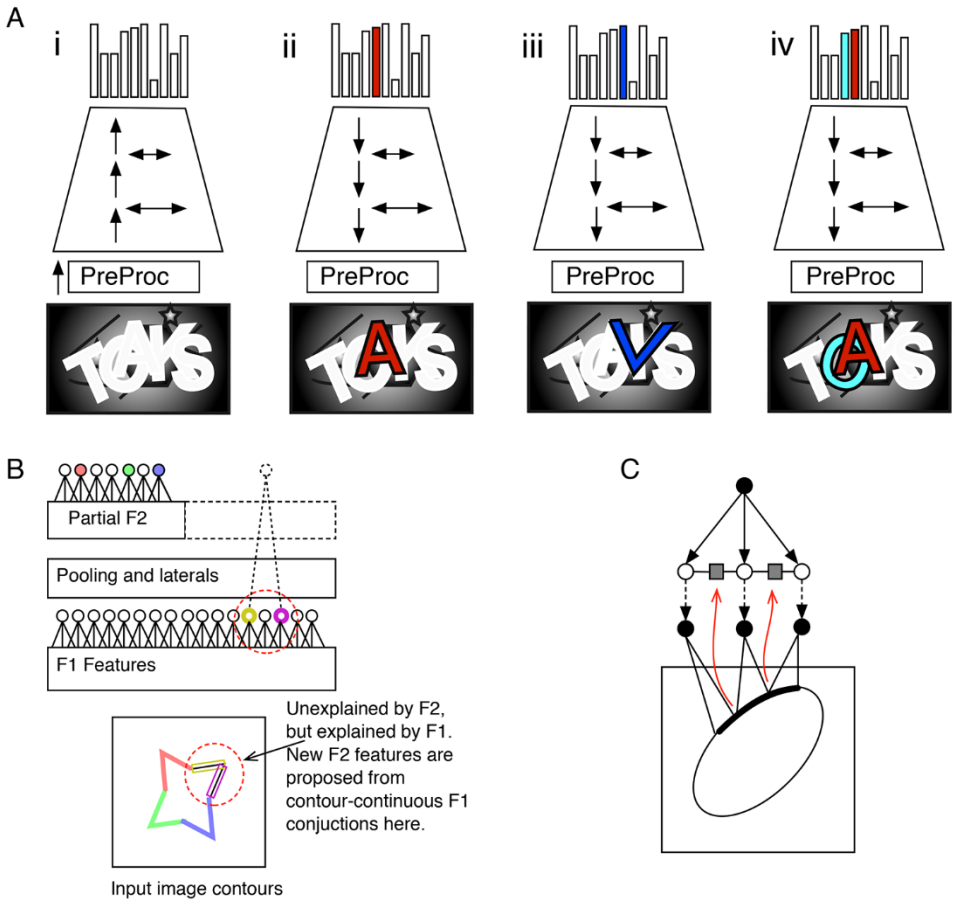
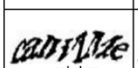
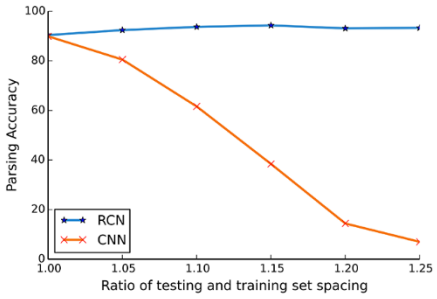


Fig. 4. (A) (i) Forward pass, including lateral propagation, produces hypotheses about the multiple letters present in the input image. PreProc is a bank of Gabor-like filters that convert from pixels to edge likelihoods [section 4.2 of (33)]. (ii) Backward pass and lateral propagation creates the segmentation mask for a selected forward-pass hypothesis, here the letter "A" [section 4.4 of (*33*)]. (iii) A false hypothesis "V" is hallucinated at the intersection of "A" and "K"; false hypotheses are resolved via parsing [section 4.7 of (*33*)]. (iv) Multiple hypotheses can be activated to produce a joint explanation that involves explaining away and occlusion reasoning. (B) Learning features at the second feature level. Colored circles represent feature activations. The dotted circle is a proposed feature [see text and section 5 of (*33*)]. (C) Learning of laterals from contour adjacency (see text).

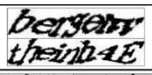
A

Input & Ground Truth	RCN Top Parse	RCN Second Parse	Human Labels
 calwlme	 calwime	 calwlme	canmme caiwime
 erldhhm	 erldbhm	 erldhhm	erldnhm erldhhm
 esterrow	 esterrow	 estenow	esterrow esterrow

B



C

CAPTCHA name	Examples	Word Accuracy	Character Accuracy
reCAPTCHA		66.6%	94.3%
Botdetect		64.4%	91.6%
Yahoo		57.4%	92.5%
PayPal		57.1%	89.3%

D

Fig. 5. Parsing CAPTCHAs with RCN. (**A**) Representative reCAPTCHA parses showing top two solutions, their segmentations, and labels by two different Amazon Mechanical Turk workers. (**B**) Word accuracy rates of RCN and CNN on the control CAPTCHA data set. CNN is brittle and RCN is robust when character-spacing is changed. (**C**) Accuracies for different CAPTCHA styles. (**D**) Representative BotDetect parses and segmentations (indicated by the different colors).

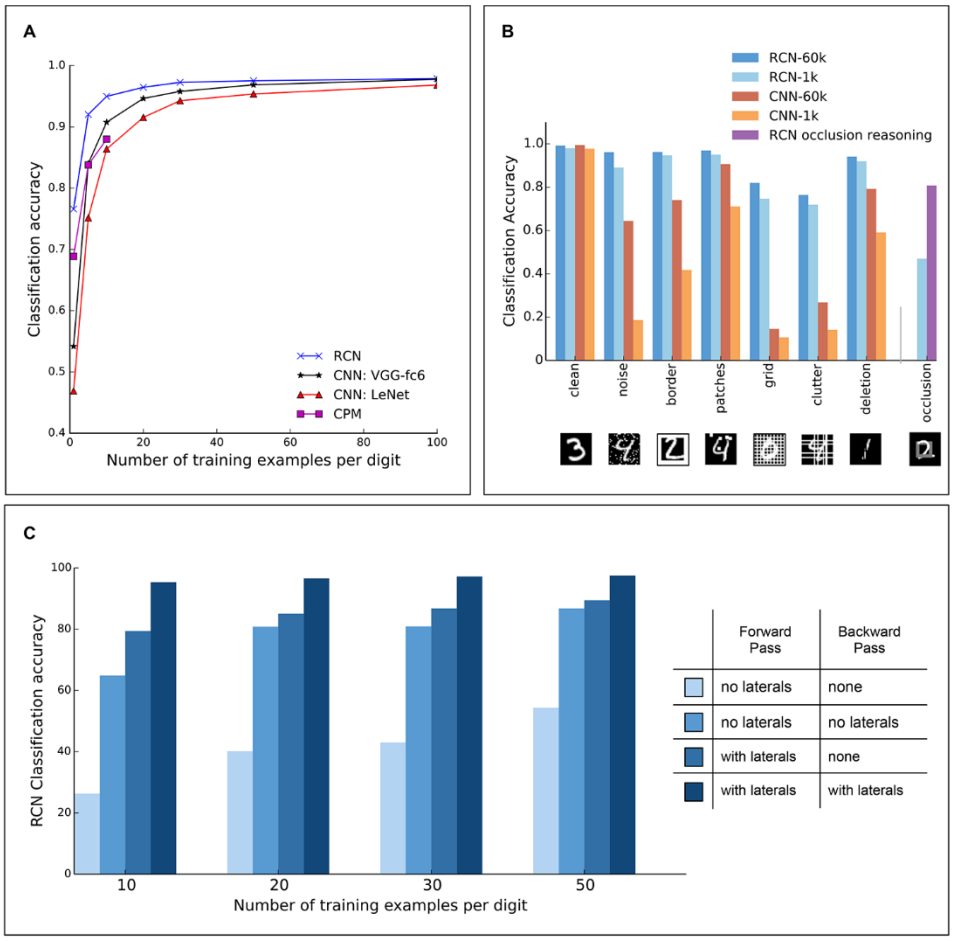


Fig. 6. MNIST classification results for training with few examples. (**A**) MNIST classification accuracy for RCN, CNN, and CPM. (**B**) Classification accuracy on corrupted MNIST tests. Legends show the total number of training examples. (**C**) MNIST classification accuracy for different RCN configurations.

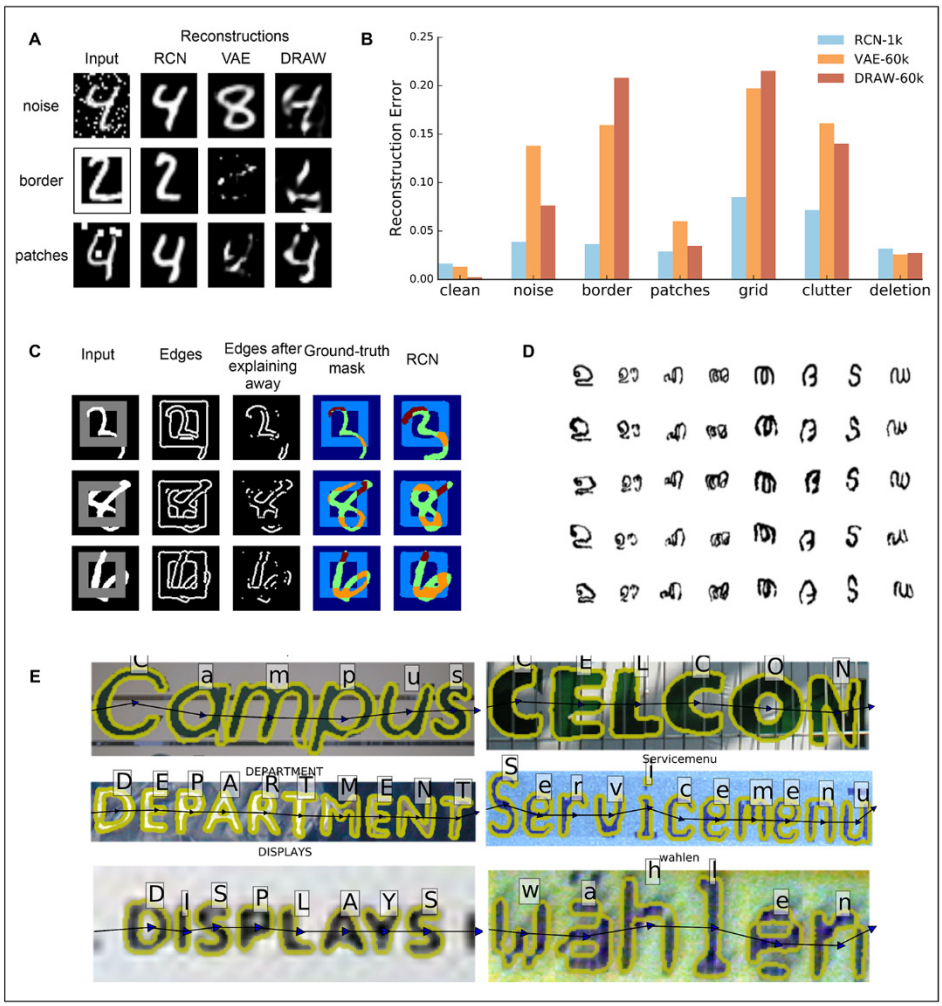


Fig. 7. Generation, occlusion reasoning, and scene-text parsing with RCN. Examples of reconstructions (**A**) and reconstruction error (**B**) from RCN, VAE and DRAW on corrupted MNIST. Legends show the number of training examples. (**C**) Occlusion reasoning. The third column shows edges remaining after RCN explains away the edges of the first detected object. Ground-truth masks reflect the occlusion relationships between the square and the digit. The portions of the digit that are in front of the square are indicated by brown color and the portions that are behind the square are indicated by orange color. The last column shows the predicted occlusion mask. (**D**) One-shot generation from Omniglot. In each column, row 1 shows the training example and the remaining rows show generated samples. (**E**) Examples of ICDAR images successfully parsed by RCN. The yellow outlines show segmentations.

Fig. 8. Application of RCN to parsing scenes with objects. Shown are the detections and instance segmentations obtained when RCN was applied to a scene parsing task with multiple real-world objects in cluttered scenes on random backgrounds. Our experiments suggest that RCN could be generalized beyond text parsing [see section 8.12 of (*33*) and Discussion].

Table 1. Accuracy and number of training images for different methods on the ICDAR-13 robust reading data set.

Method	Accuracy	Total no. of training images
PLT (*54*)	64.6%	Unknown
NSEP (*54*)	63.7%	Unknown
PicRead (*54*)	63.1%	Unknown
Deep Structured Output Learning (*55*)	81.8%	8,000,000
PhotoOCR (*54*)	84.3%	7,900,000
RCN	**86.2%**	**26,214 (reduced to 1406)**

Science

A generative vision model that trains with high data efficiency and breaks text-based CAPTCHAs

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

published online October 26, 2017