Guarding Against Adversarial Attacks using Biologically Inspired Contour Integration

Salman Khan — Systems Design Eng. Dept., University of Waterloo, ON, Canada
Alexander Wong — Systems Design Eng. Dept., University of Waterloo, ON, Canada
Bryan Tripp — Systems Design Eng. Dept., University of Waterloo, ON, Canada

Abstract

Artificial vision systems are susceptible to adversarial attacks. Small intentional changes to images can cause these systems to misclassify with high confidence. The brain has many mechanisms for strengthening weak or confusing inputs. One such technique, contour integration can separate objects from irrelevant background. We show that incorporating contour integration within artificial visual systems can increase their robustness to adversarial attacks.

1 Introduction

Deep neural networks have surpassed human-level performance [1] on visual perception tasks such as object classification. However, the performance of these networks degrades dramatically when presented with adversarial images [3]. In these adversarial images, small perturbations which are often imperceptible to humans are added to cause these models to misclassify with high confidence.

Under difficult viewing conditions, the brain uses a variety of contextual modulation techniques, whereby signals from outside the receptive field (RF) of neurons alter within RF responses, to augment weak and confusing feed-forward inputs. One such technique, contour integration [6] is employed with edge extraction in the primary visual (V1) cortex. Contour integration was first observed psycho-physically as the popping out of patterns of small line segments that followed smooth trajectories in the presence of distractors (see figure 2A and C). Next, it was found that V1 neurons whose RFs overlapped with co-aligned fragments showed elevated responses. [15, 6]. As the contours of natural objects are mostly smooth with few sharp edges, contour integration is thought to be a mechanism for separating out object contours from irrelevant background.

In this study, we investigate whether such a mechanism embedded within a large-scale artificial object classification can assist in combating adversarial attacks. This might be expected, because contour integration makes the brain more sensitive to features that often appear in natural scenes, and have significance for understanding these scenes, whereas adversarial attacks often involve changes changes with quite different statistics. The contributions of this study are:

1. A novel contour integration model is proposed. Different from previous stand-alone models [7, 8, 9], the primary focus of the model is on investigating the role of contour integration within an object classification network.

2. An evaluation of the proposed contour integration model within an object classification network on adversarial images to demonstrate its efficacy for providing defence against adversarial attacks.

2 Methodology

2.1 Contour Integration Layer

Contour integration works in conjunction with edge detection. Once pertinent edges are extracted, contour integration modulates the output of each edge-extracting neuron based on how many of its neighbors detected co-aligned edges.

We define contour integration as,

$$A_L(x,y,k) = A_F(x,y,k) + \sigma\left(A_F(x,y,k)\sum_{m,n,i} W_k \circledast A_F(m,n,i) + b_k\right), \quad (1)$$

where, $A_L(x,y,k)$ is the contour enhanced activation at position (x,y) on channel k, $A_F(m,n,i)$ is the activation of the neuron at position $(x-m, y-n)$ and channel i in the previous layer, W_k is the contour integration kernel for channel k, b is a bias term, $\sigma(.)$ is a nonlinear activation function and $\circledast$ is the convolutional operator.

The structure is similar to a convolutional feature extracting layer but includes constraints that replicate properties of lateral connections of V1 neurons. First, contour integration is a modulatory effect that is present only when there exists a signal within the RF. Second, if there is no contour enhancement, the feed-forward input passes through as is. Third, as the spatial extent of lateral connections is much larger than the RF of V1 neurons [4], contour integration kernels are larger than feature extracting ones. Forth, since contour integration kernels model lateral connections, there needs to be a one-to-one correspondence between the number of feature extracting and contour integration kernels. No other constraints are included and contour integration kernels need to learn which of their neighbors to connect with in order to enhance contours of their corresponding feature extracting kernels.

2.2 Training the Contour Integration Layer

Stimuli similar to those of [5] were used for training. They consisted of co-aligned 2D Gabor fragments (contour) embedded in a sea of similar but randomly oriented fragments. Each contour was defined by four parameters: (1) contour length c_{len}, (2) the spacing between fragments $c_{spacing}$, (3) contour curvature, β, and (4) the deviation of individual fragments from the orientation of the contour, α (see Table 1 for details and values used in the training set). In each training image, contours were centrally placed with the middle fragment aligned with the RF of the middle neuron of the target edge extraction kernel. A sample training image is shown in 1.

Table 1: Contour Parameters and their Ranges in the Training Set.

Name	Range	Definition
c_{len}	(1, 3, 5, 7, 9)	Number of co-aligned fragments.
$c_{spacing}$	(1, 1.2, 1.4, 1.6, 1.9)	Ratio of distance (pixels) between the centers of two consecutive fragments to fragment length.
β	(0, $\pm$15, $\pm$30, $\pm$45, $\pm$60)	Contour curvature. The rotation (degrees) of the orientation between consecutive contour fragments.
α	(0, $\pm$15, $\pm$30)	Orientation offset of a fragment with respect to contour curvature.

Each contour integration kernel was trained individually. First, the optimum 2D Gabor fragment that maximally activated the target feature extracting kernel was found. Next, 200 training images for each contour parameter combination (Table 1) were constructed. This resulted in a total of 30,000 training images for each contour integration kernel. For a given magnitude of β and α, positive and negative values were drawn randomly for each contour fragment, so the contours in each training image were unique. Further uniqueness was added by generating randomly oriented background fragments in each image.

Contour-integration kernels were learned using supervised training. Expected enhancement gains were derived from the results of [5, 6]. The results in [6] provide empirically measured enhancement gains at the level of individual neurons, but only for linear contours. [6] found a strong correlation between behavioral detectability and empirically measured enhancement gains in macaques. Therefore we extrapolated these results to curved contour based on the behavioral results of [5], which measured the ability of test subjects to detect curved contours. As contour detectability decreases with curvature, we equated behavioral detectability of 100 percent with enhancement gains of a linear contours with a similar configuration. Expected enhancement gains for curved contours were found by multiplying detectability results with measured linear gains.

A mean square error (MSE) cost function was used during training,

$$L(x,y,k) = \frac{1}{N}\left(G_{expected} - \frac{A_L(x,y,k)}{(A_F(x,y,k)+\varepsilon)}\right)^2 + \lambda|W_k|, \quad (2)$$

where $L(x,y,k)$ is the loss for kth contour integration kernel, N is the total number of images in the training set, $G_{expected}$ is the expected contour enhancement gain and λ is a scaling parameter. L1 weight regularization was used to limit the number of learned lateral connections.

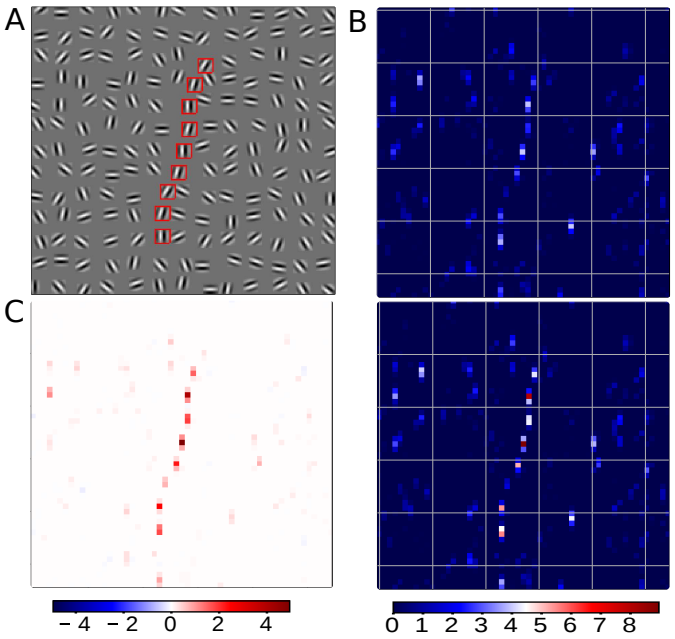

Fig. 1: Training contour-integration kernels. (A) Sample image for an edge extracting kernel with a vertical orientation. All fragments are identical except for their orientation. Centrally located fragments were co-aligned to form a contour ($c_{len} = 9$, $c_{spacing} = 1$, $\beta = \pm 15$, $\alpha = 0$). Contour fragments are bounded by red tiles (only for display) (B) (Top) activation map of the edge extracting kernel (Bottom) activation map of the corresponding contour integration kernel. (C) Difference between feature extracting and contour integration activations.

2.3 Training the Full Model

AlexNet [2] was selected as the parent classification system. The contour integration layer was inserted after the first convolutional layer. Many of the feature extracting kernels in this layer function as edge detectors. The large size (11x11) of these kernels facilitated finding optimal 2D Gabor fragments necessary for training contour integration kernels. Not all feature extracting kernels are edge extractors. Moreover, some fitted 2D Gabor fragments had a high spatial frequency and were unsuitable for the contour image generation process. For these feature extracting kernels, their contour integration kernels was set to zero, thereby allowing their activations to pass through unaltered. In summary, contour integration kernels were learned for 22 of the 96 possible feature extracting kernels.

Contour integration kernels are learned after edge extracting kernels are finalized. Starting from a pre-trained model [10], the contour integration layer was trained separately and inserted back into the larger object classifications system. The weights of the first feature extracting and contour integration layers were fixed and the rest of the model was retrained for object classification on ImageNet [12]. Training parameters for the contour integration and the full model are listed in Table 2.

2.4 Adversarial Attack

The adversarial attack of [13] is used to test the robustness of the full model. Starting with a random population of pixel perturbations (location and RGB values), differential evolution is used to iteratively search for a configurable set of perturbations that cause the model to misclassify an image. It is a black-box technique that only requires predicted class probabilities. Compared to other adversarial attacks, it is simpler to use, applicable to more models (does not require access to gradients or network structure) and the size of the induced perturbation can be controlled. Using an AlexNet model, [13] achieved a single pixel attack success rate of 41.22% over the ImageNet (ILSVRC 2012) validation set. We attack the fully trained model using 1,3,5 pixels attacks. Parameters for the differential evolution algorithm were similar to those of [13]. Source code for the adversarial attack was taken from [11].

Table 2: Training Parameters

Model	Contour Integration Layer	AlexNet + Contour Integration
epochs	100	35
batch size	32	64
loss function	Mean Square Error	Cross Entropy
kernel size	35x35	
λ	0.0005	
Activation function	Leaky ReLU ($\alpha = 0.9$)	

3 Results

Figure 2A and C show example contour images not seen during training. The model successfully detected the embedded contours even though they are of a unknown shape, length and position, Figure 2B and D respectively. Figure 3 shows that the model successfully learned many contour integration properties. Enhancement gains increase with contour length, but decrease as contour curvature or fragment spacing increase. Moreover, the learned connection pattern are similar to known connectivity profiles of V1 lateral connections [14]. Excitatory connections are formed in the preferred direction of the feature extracting neuron while mostly inhibitory connections are learned in the orthogonal to the preferred direction (results not shown).

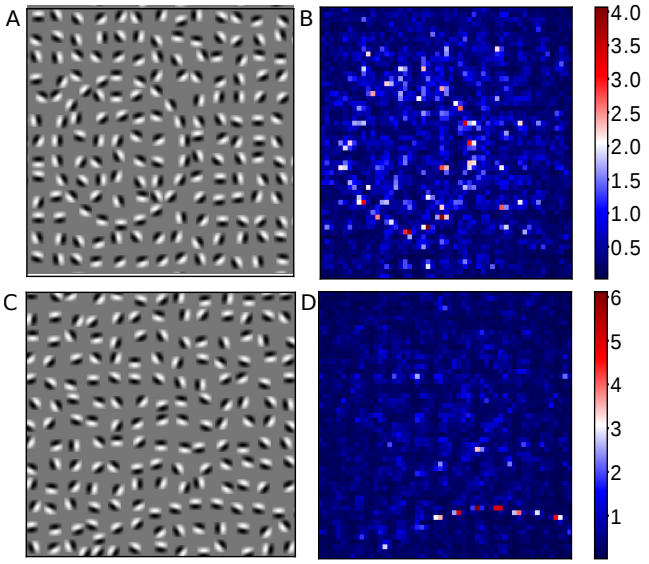

Fig. 2: Enhancing unseen contours. (A) A circular contour embedded in a sea of distractors. (B) Modifications made by the contour integration layer. Shown is the maximum difference between contour enhanced and edge extraction activations across all channels. Circular contours were not included in training. (C and D) are similar to figures (A) and (B) but for a position not included in training.

Results of the adversarial attacks on our contour integration model and a control AlexNet model are shown in Table 3. Both models were trained similarly. Results are averaged across 3 trials. In each trial, 1000 image were randomly chosen from the ILSVRC 2012 validation set to access model accuracy. From the set of images that the network correctly classified, 300 images were randomly selected for pixel level attacks. As can be seen, the inclusion of the contour integration layer provides marginally better results for all pixel-level attacks.

Our attack success rates for the 1 pixel attack are lower than those of [13]. We used different image pre-processing and trained our model for a shorter period of time. We additionally tested the

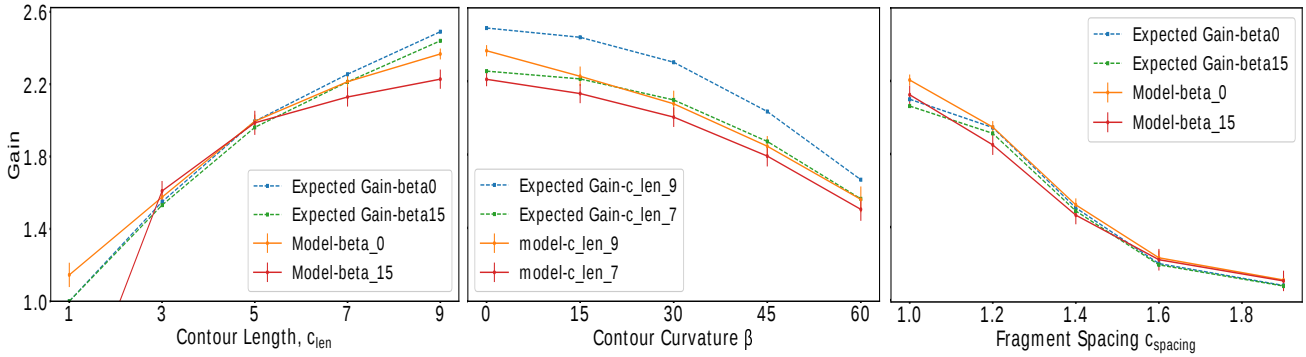

Fig. 3: Replicated neurophysiological properties. (A) Gain vs. contour length. (B) Gain vs. contour curvature. (C) Gain vs. fragment spacing. Each point is the average across 100 images. Vertical bars show ± 1SD. Expected gains are derived from [5, 6] as explained in section 2.2.

the 1 pixel attack on a pre-trained Mobilenet [16] that used similar pre-processing to our model. Even though the model had a much higher accuracy of 67%, an attack success rate of only 13% was found. In general, the parameters of the differential evolution algorithm need to be tuned for the expected range of input values. This is left for future work. In summary, our results show that the proposed model outperformed the control model for all pixel attacks under identical attack settings.

Table 3: Adversarial Attack Results

Model	Top-1 Accuracy	Pixels	Attack Success
AlexNet	36.1±0.4	1	8.5±0.4 %
		3	10.6±2.2 %
		5	10.4±2.1 %
AlexNet + Contour	38.0±1.1	1	8.22±1.8 %
		3	9.76±2.0 %
		5	9.55±1.75 %

4 Conclusion

Contour Integration is a technique that the brain's object detection system uses to augment feed-forward signals under difficult viewing conditions. In this study, we present a novel model of contour integration that is embedded inside a large-scale artificial object classification system. We show that the model replicates many of the neurophysiological properties of contour integration and provides some robustness to adversarial attacks for these systems. As such, the preliminary results presented above indicate that the inclusion of contour integration offers some protection against adversarial attacks. We plan to further refine the results by tuning the model for better accuracy, incorporating more contour integration kernels, and testing against multiple other adversarial attacks.

5 Acknowledgment

The authors thank the Natural Sciences and Engineering Research Council of Canada (NSERC) and the Canada Research Chairs program.

References

[1] He, K., et al., Deep residual learning for image recognition. *In Proceedings of the IEEE conference on computer vision and pattern recognition* (2016).

[2] Krizhevsky, A., Sutskever, I. and Hinton, G.E., Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems* (2012).

[3] Kurakin, A., Goodfellow, I. and Bengio, S. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533* (2016).

[4] Stettler, D. D., et al., Lateral connectivity and contextual interactions in macaque primary visual cortex. *Neuron* (2002).

[5] Field, D. J., Hayes, A. and Hess, R.F. Contour integration by the human visual system: evidence for a local association field. *Vision research* (1993).

[6] Li, W., Piëch, V. and Gilbert, C.D., Contour saliency in primary visual cortex. *Neuron* (2006).

[7] Li, Z., 1998. A neural model of contour integration in the primary visual cortex. *Neural computation* (1998).

[8] Ursino, M. and La Cara, G.E., A model of contextual interactions and contour detection in primary visual cortex. *Neural Networks* (2004).

[9] Piëch, V., et al., Network model of top-down influences on local gain and contextual interactions in visual cortex. Proceedings of the National Academy of Sciences, p.201317019.

[10] https://github.com/heuritech/convnets-keras.

[11] https://github.com/Hyperparticle/one-pixel-attack-keras.

[12] Deng, J., et al. Imagenet: A large-scale hierarchical image database. *In Computer Vision and Pattern Recognition* (2009).

[13] Su, J., Vargas, D.V. and Kouichi, S., One pixel attack for fooling deep neural networks. *arXiv preprint arXiv:1710.08864* (2017).

[14] Kapadia, et al., Spatial distribution of contextual interactions in primary visual cortex and in visual perception. *Journal of neurophysiology* (2000).

[15] Kapadia, et al., Improvement in visual sensitivity by changes in local context: Parallel studies in human observers and in V1 of alert monkeys. *Neuron* (1995).

[16] Howard, A.G., et al., Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861* (2017).