HOW DO CONVNETS UNDERSTAND IMAGE INTENSITY?

Jackson Kaunismaa[1] and Michael Guerzhoy[1,2]
[1]University of Toronto, [2]Li Ka Shing Knowledge Institute at St. Michael's Hospital
jackson.kaunismaa@mail.utoronto.ca, guerzhoy@cs.toronto.edu

ABSTRACT

Convolutional Neural Networks (ConvNets) usually rely on edge/shape information to classify images. Visualization methods developed over the last decade confirm that ConvNets rely on edge information. We investigate situations where the ConvNet needs to rely on image intensity in addition to shape. We show that the ConvNet relies on image intensity information using visualization.

1 INTRODUCTION

Convolutional Neural Networks (ConvNets) usually rely on edge/shape information to classify images. Shape is much more important than color in most situations (see e.g. (Reppa et al., 2020) for experiments on ImageNet). Both animals (Hubel & Wiesel, 1962) and ConvNets (Zeiler & Fergus, 2014) process images by detecting edges and corners in them. However, in some scenarios, color/intensity is important for object classification. Examples include medical imaging where the intensity of the "shadows" in the image may be important (Ezoe et al., 2002), classification of objects where color is an important cue (e.g. bird species Welinder et al. (2010) classification), and scene recognition for landscape photos where color is more informative than shape (Gallagher et al., 2004).

In this paper, we explore how a ConvNet classifies images when color/image intensity is important. For simplicity, we focus on situations where *image intensity* is important. We generate synthetic images which can only be classified by paying attention to intensity, in addition to shape. We achieve this by making the class depend on intensity via a highly non-monotonic function. We visualize how the ConvNet classifies the images to show that the ConvNet uses image intensity as an important cue when "seeing" our synthetic inputs.

2 TASK: SYNTHETIC DATA

We generate a dataset of greyscale images, each of which contains a single object of uniform intensity. We include additive noise at multiple scales. Classifying "dark" vs "light" is trivial. In order to make the task difficult, we make the target output be a non-monotonic function of the intensity of the object. See Fig. 2. We consider three classes, with, e.g., Class 0 corresponding to image intensity ranges of 0 to 30, 120 to 150, and 210 to 240.

3 HOW DOES THE CONVNET UNDERSTAND INTENSITY?

3.1 CLASSIFICATION

We train several ConvNets (see Appendix 5.4 for the details) on 250k synthetic images. We achieve a correct classification rate of 98.2% with a large enough network on the test set (91.7% with a smaller network), with a base rate performance of 35.3%.

For the smaller network, the correct classification performance drops to 84.5% from 91.7% if the pixels of the image are permuted so that there are no shape cues in the image, indicating that the network relies on shape cues as well as image intensity cues.

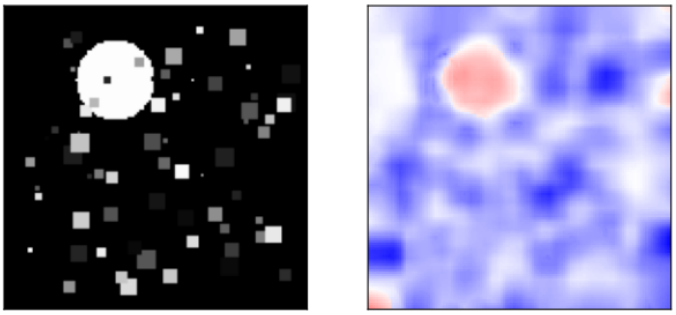

Figure 1: Left: Sample image from the synthetic dataset. The target is the class associated with the intensity of the grey circle, and the noise elements are the squares of various intensities and sizes. Right: Successful visualization of salient pixels. The interior of the circle is correctly identified as important.

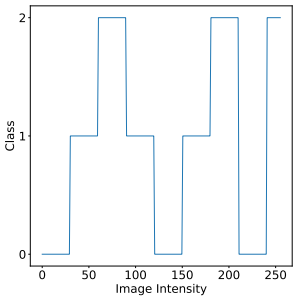

Figure 2: The target function of the task. E.g., circles of intensity 30-60 are in Class 1, and circles of intensity 180-210 are in Class 2

3.2 VISUALIZATION

We visualize the pixels in the images that are important for the correct output using a method similar to Guided Backpropagation (Springenberg et al., 2014). On many test images, we see evidence that the network cares about the intensity in addition to the shape. We plot the average activation of a given convolutional channel across spatial dimensions to show that intermediate units in the network are highly dependent on intensity. Details are in the Section 5.3. A sample successful visualization is shown in Fig. 2.

4 CONCLUSION

We investigated how ConvNets understand images where the main cue is image intensity rather than shape. We demonstrated that it is possible to visualize the network to confirm that it uses intensity in addition to shape. Although shape is more important than color in most image data, our work could be useful in beginning to understand how ConvNets process image data where intensity and/or color is more important.

In this paper, we focused on scenarios where the target class is a highly non-monotonic function of the intensity of the input. Scenarios where the target class is a highly non-monotonic function of the 3D color are theoretically possible; however, among other issues, cameras are usually not calibrated with respect to color Tedla et al. (2022).

Future work includes applying our methods and analysis to natural datasets, as well as empirically analyzing situations where the color, and not just the intensity, is important.

URM Statement: the authors qualify under the URM criteria.

The authors thank Prof. Michael S. Brown for his insightful comments and suggestions.

5 APPENDIX

5.1 PERFORMANCE

We trained several networks on both the task presented in section 2, as well as a modified, harder task, where the target is the same non-monotonic function of intensity, but the pixels in the input are permuted so that the network cannot rely on shape to identify the areas containing the relevant target intensity.

Task	Network	Test Accuracy (%)
Unpermuted	Large	98.2
Unpermuted	Small	91.7
Permuted	Large	97.6
Permuted	Small	84.5

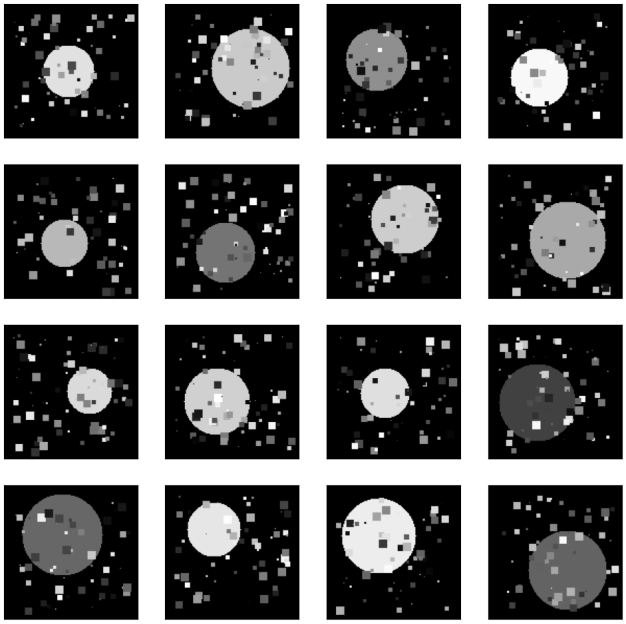

Figure 3: A selection of several examples from the unpermuted synthetic dataset.

5.2 INTENSITY RESPONSE GRAPHS

In order to visualize how a given neuron responds to intensity, we display a "intensity response graph." The intensity response is estimated by taking a large sample of synthetic images with circles of varying intensity, and plotting the resulting activation of the neuron (Figure 5.2). We consider the activations after the ReLU, as they are essential for representing the "thresholding" behavior that intensity classifiers require. To visualize the variability in activation at a given intensity level, we plot each sampled (circle intensity, response) pair, as a pale blue dot. The average activation of the neuron at a given intensity level is depicted with a red line. We plot the associated target function of intensity as a grey line in the background. The reader can compare the grey line with the red line to see how the average output corresponds to the target.

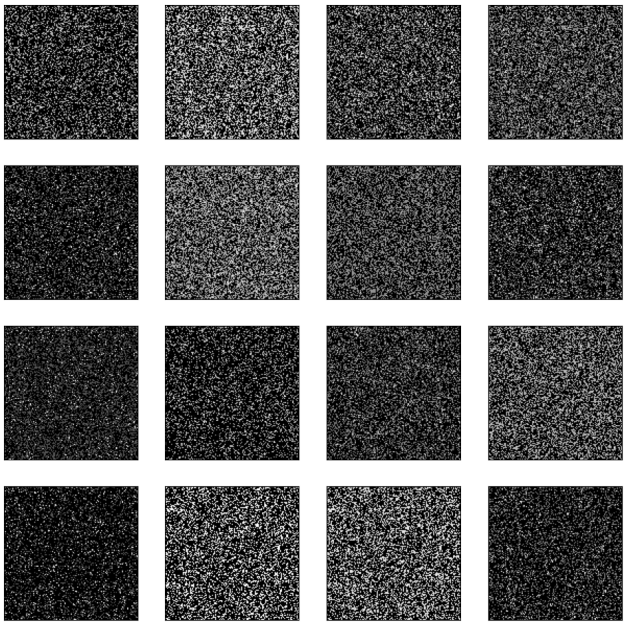

Figure 4: A selection of several examples from the modified synthetic dataset with permuted pixels.

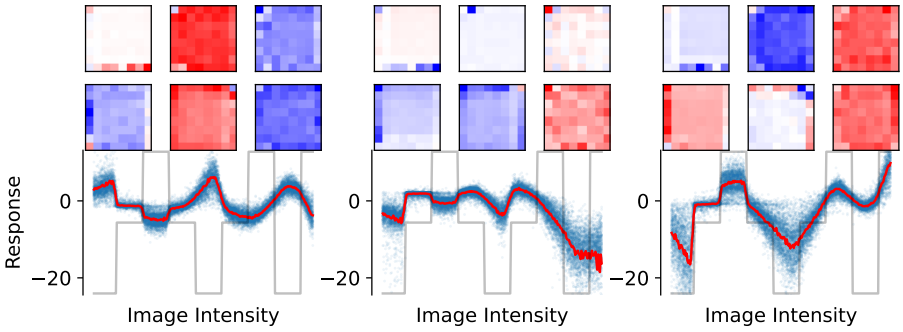

Figure 5: Visualization of the output layer of the small network. Column i corresponds to output logit i. Top: Each subsection of the grid in a column i represents the weights that are dotted with the output of the last convolutional layer to produce output logit i. Bottom: The intensity response graph of the three output logits of the network.

5.3 VISUALIZATION

We first visualized the network using Guided Backpropagation (Springenberg et al., 2014)[1]. Guided Backpropagation and similar methods for producing saliency maps are known to tend to emphasize edges in the image regardless of whether they are actually important to the output (Adebayo et al., 2018). We modify Guided Backpropagation to allow us to demonstrate that our network does use pixels far away from edges.

Guided Backpropagation computes a saliency map indicating which locations in the input were important for the output using a modified version of the gradient of the output with respect to the input pixels. In our visualization, we compute a modified version of the gradient with respect to a

[1]See Li, Karpathy, and Johnson for a clear presentation of Guided Backpropagation `http://cs231n.stanford.edu/slides/2016/winter1516_lecture9.pdf`. We recommend that the reader familiarize themselves with Li et al.'s presentation before reading this Section

local linear approximation of image patches, allowing as to visualize which *patches* in the images are important for the output.

We run Principal Component Analysis on a training set of image patches from our dataset. We then consider the quantity $f(I+\alpha p)$, where I is the image, p is a learned principal component of a patch, padded with 0's to be the same size as the image, and f is our network. The quantity $\partial f(I+\alpha p)/\partial\alpha$ is then analogous to the gradient of the output with respect to the input. We modify the computation of $\partial f(I+\alpha p)/\partial\alpha$ analogously to Guided Backpropagation, removing the contributions of paths in the network that contain negative edges.

To compute the importance of a particular pixel in the image, we take the maximum over multiple scales of the modified $\partial f(I+\alpha p)/\partial\alpha$ for principal components p for each pixel, and then linearly interpolate those quantities to obtain the saliency map.

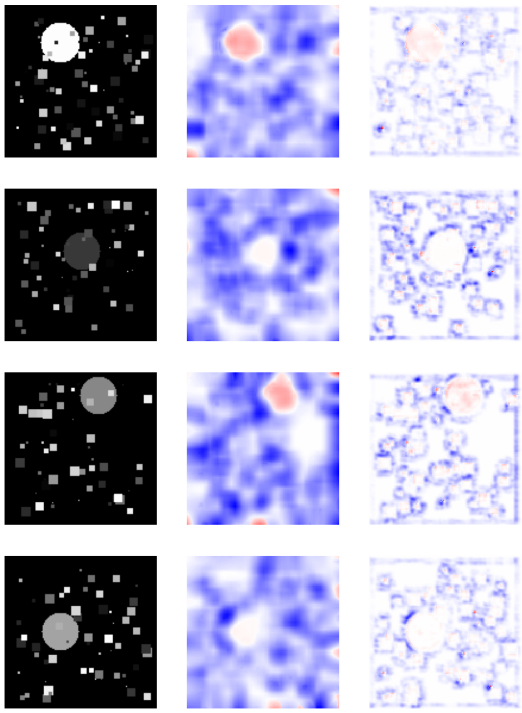

Figure 6: Selected examples of saliency maps. Column 1: image from dataset. Column 2: Saliency map computed using our method. Column 3: Guided Backpropagation.

5.4 Model Architecture

For ease of visualization and interpretation, we considered both a "small" ConvNet and a "large" ConvNet consisting of the following layers:

Layer	In Channels	Out Channels	Stride
1	1	16	1
2	16	16	1
3	16	32	1
4	32	32	1

Table 1: Architecture of large network.

Layer	In Channels	Out Channels	Stride
1	1	2	4
2	2	2	1
3	2	6	4
4	6	6	1

Table 2: Architecture of small network.

The convolutional layers are followed by a single fully-connected layer with 3 output logits in each network, and all convolutions are with 3x3 kernels. Each layer in the network uses Batch Normalization and ReLU non-linearities.

Network hyperparameters were tuned with a random search over learning rate, initialization variance, and weight decay. They were trained using the Adam optimizer and the cross-entropy loss on $250,000$ samples from the synthetic dataset. We emphasize that the goal of this paper was not to obtain high performance on our specific dataset, but rather to analyze how convolutional networks solve the task.

REFERENCES

Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. Sanity checks for saliency maps. *Advances in Neural Information Processing Systems*, 31, 2018.

Toshiharu Ezoe, Hotaka Takizawa, Shinji Yamamoto, Akinobu Shimizu, Tohru Matsumoto, Yukio Tateno, Takeshi Iimura, and Mitsuomi Matsumoto. Automatic detection method of lung cancers including ground-glass opacities from chest x-ray ct images. In *Medical Imaging 2002: Image Processing*, volume 4684, pp. 1672–1680. SPIE, 2002.

A.C. Gallagher, J. Luo, and W. Hao. Improved blue sky detection using polynomial model fit. In *2004 International Conference on Image Processing, 2004. ICIP '04.*, volume 4, pp. 2367–2370 Vol. 4, 2004. doi: 10.1109/ICIP.2004.1421576.

David H Hubel and Torsten N Wiesel. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of physiology*, 160(1):106, 1962.

Irene Reppa, Kate E Williams, W James Greville, and Jo Saunders. The relative contribution of shape and colour to object memory. *Memory & Cognition*, 48:1504–1521, 2020.

Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806*, 2014.

SaiKiran Tedla, Yunyuan Wang, Maitri Patel, and Michael S Brown. Analyzing color imaging failure on consumer-grade cameras. *JOSA A*, 39(6):B21–B27, 2022.

Peter Welinder, Steve Branson, Takeshi Mita, Catherine Wah, Florian Schroff, Serge Belongie, and Pietro Perona. Caltech-ucsd birds 200. 2010.

Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I 13*, pp. 818–833. Springer, 2014.