Understanding Image segmentation using Occlusion and Negative space images.

Jason Driver
Computer Science
University of California, Davis
Davis, USA
jydriver@ucdavis.edu

Suraj Kesavan
Computer Science
University of California, Davis
Davis, USA
spkesavan@ucdavis.edu

Devika Joshi
Computer Science
University of California, Davis
Davis, USA
dmjoshi@ucdavis.edu

Abstract

Fully Convolutional neural networks[9] provide state of art results for tasks like region based segmentation. Image segmentation helps in understanding the nature and context of an image by giving insights into object structure, depth perception and figure/ground organization. Understanding the convnets has been a difficult task with fewer test platforms to check on every layer of the convnet. In this paper, we discuss the method of occulusion tests and provide an overview on their behavior with different layers on the residual neural networks. We also discuss the importance of understanding figure/ground organization and occlusion in understanding the perceptual organization of an image.

1. Introduction

Image segmentation mainly aims to cluster pixels into salient image regions of interest in a scene or annotate the data. Image segmentation has been studied in detail based on edge detection [12] and depth layering [7]. Edge detection works by detecting discontinuities in brightness of an image giving no inherent details on the perceptual organization of the image [5]. According to Gestalt psychology [4], Human brain tends to fill in the missing information automatically making us understand perceptual organization even when an object is occluded. Gestalt's law of closure [4] state that things are grouped together if they seem to complete some entity in an image. Negative space images, inspired from the artistic drawings of M.C.Escher, provide the perfect example of how human beings recognize the law of closure naturally. Negative space, in art, is the space around and between the subject of an image. The space around the subject of the image also reveals an interesting shape or an object. Negative space images also help in classification of image based on figure/ground organization. Our aim is to perform experiments on how figure/ground organization can be predicted using the inherent logic of negative space images.

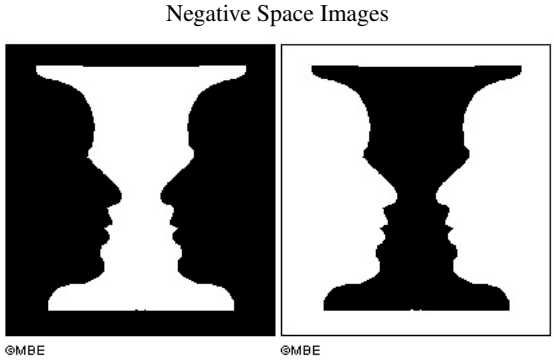

Figure 1. The picture depicts the example of a negative space image. Image on the left has the figure in white and ground in black and vice versa.

Spectral clustering algorithms [16] are employed to find the connectivity between the pixels in an image and are traditionally used to find the figure/ground segregation. We employ graph cut algorithms to predict the figure/ground using the minimum spanning tree algorithm. We believe spectral clustering could be used to identify between white and black segments in an image. The output of the algorithm leads us to predict images from the shapes efficiently [16]. A wide range of vision problems make use of good segmented objects. For instance, image indexing and object

recognition tasks use segments for the task of mapping the segments to labels.

Occlusion is a concept that two objects that are spatially separated in the 3D world might interfere with each other in the 2D image plane. Occlusion has been employed as a technique to introduce bias in the network where the network is fed in with occluded images. We perform experiments by systematically occluding different portions of the input image with a size adjustable grey square, and monitoring the output of the classifier. These experiments follow the idea induced by the paper by Zeiler et al [19], and are tested on the Facebook's Residual Neural Network [8].

Our work is split into two parts, We provide a literary survey on how negative space image could help in understand the perceptual organization and we provide the results of various occlusion test conducted on a residual neural network. In the next section, we list out the related work done in the field of analysis of figure/ground organization and occlusion testing. The following sections explain the experiments performed using negative space images and our technical approach on building an occlusion test platform in Python for a residual neural network model.

2. Related Work

Literature on segmenting images is large and has been built over the last thirty years with applications in various fields of computer vision. The early work on image segmentation was mainly based on edge detection and boundary detection. Boundary detection was mainly performed using spectral methods, where one formulates an eigenvalue system to determine a low level pixel grouping problem. The eigenvectors produced from the system are used to predict boundaries by performing laplacian transformations on the corresponding matrices. Notable work in using spectral methods have been MCG [2], gPb [1], PMI [6], and Normalized Cuts [13]. But the computations involved in spectral methods are high. For example, an image from ImageNet dataset with dimensions of 256*256 would have 65,536 pixels and its similarity matrix would be of dimensions 65,536*65,536, which is amounts to 4,294,967,296. Hence this method is generally slow to process. Adding to this, the selection of affinity functions are tricky, leading to improper results in most of the occasions.

Despite a lot of work being done in image segmentation, not a lot of work has been done to perform figure/ground segregation and none relating the negative space images. Some of the works which are related are,[18] deals the issue by creating affinity matrices relating pixels, patches and pixel-patch mutual ownership. [3] deals with image segmentation as a graph-cut problem and provides a greedy algorithm to segment the edges based on their weights. This method brought down the computation down and performed in O(nlogn) where the matrix is of the size n*n. However these methods are handcrafted and are difficult to tune.

Affinity matrix which acts as a metric on how close or similar are two points in a space, were usually generated manually until Turaga et al [15] found a method to make convolutional networks predict the affinity graphs based on weights using a 4-layer network with three hidden layers.

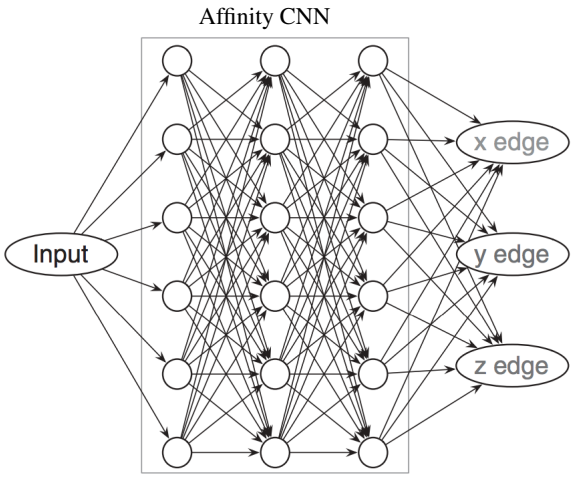

Figure 2. Convolution Neural network for Affinity matrix by Turaga et al

The same idea was incorporated in Affinity-CNN [10] by Maire et al where they train a CNN to predict pixel-centric pairwise relations for figure/ground embedding. They use a novel technique of angular embedding [17] [14] for representing the affinities between the pixels using a unit semicircle. This deep affinity network use the basic layout of Alexnet and it provides pixel affinities across 3 different scales along x-axis, y-axis and z-axis along the 8 neighbors for each pixel. The network is tested on the Berkeley Segmentation Dataset.

Occlusion tests were introduced by Zeiler et al [19] in an attempt to analyze and visualize neural networks. The authors attempt to predict the location of the object in the image by occluding different parts of the image using a grey square and monitor the classifier. Occlusion based classification methods were used to create maps of top misclassifications for when the network looked at images of the object class, but had areas occluded from the algorithm.

The method we plan on using is not novel, but has not been widely used in more recent papers for analyzing trained networks and would provide an alternative, and interesting perspective on more recent networks. We plan on using this technique to investigate the current state of the art classification networks, like the residual neural network, and to use this technique on a newly released residual neural network by Facebook's research group.

3. Technical Approach

3.1. Experiments on Negative Space Images

One main advantage of negative space images is that there are only two types of pixels which makes segmentation an easier task to pursue. Using basic spectral clustering and random coloring algorithm, we generated the following output shown in the figure 3. One main disadvantage of our algorithm is its inability to separate the figure and ground and color them coherently.

Spectral clustering algorithm on negative space images

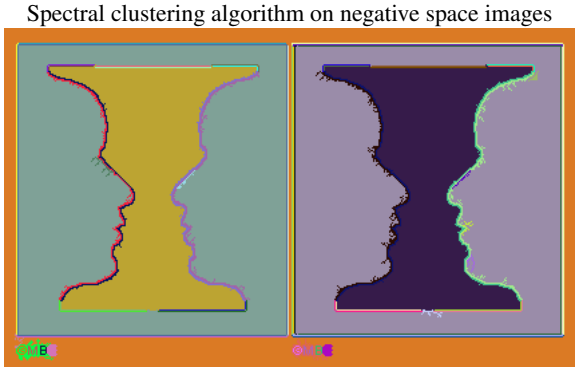

Figure 3. Spectral coloring to detect segments

Curve reconstruction seemed a plausible method to segregate the figure and ground and also would give more details on the shape of the object. We used Delaunay triangulations to perform curve reconstruction, where for a set of points in a plane, triangulation is such that no point in plane is inside the circumcircle of any triangle. Figure 4 shows the output of Curve construction. The shape of the object can be easily predicted with the outline from figure 4.

Delanuay Triangulation

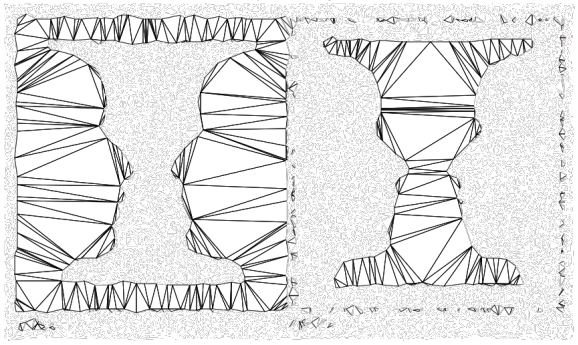

Figure 4. Delanuay Triangulation of negative images with threshold equal to 0.5

We tried to train a Affinity-CNN based on the work by Maire et al to generate the affinity graphs on the BSDS dataset- figure/ground dataset [11]. The images of the dataset were converted to two contours of black and white respectively. But the generated affinity graphs show loss of image content as compared to the ones generated for rgb images. This clearly suggests that the area of negative-space images requires further research.

3.2. Occlusion Sensitivity Based Classification Technical Approach

Occlusion sensitivity based classification performed in Zeiler et al [19], provides a novel approach for target localization within an image, and gives another way for us to view semantic representations of our target within an image. These methods have been shown to provide insights into neural networks and provide us with more methods for fine tuning neural networks.

3.2.1 Environment

We used an *unix environment for all our experiments, Ubuntu 16.04 with GPU enabled. The logic is coded in Python and works on Torch models. The Python program does not require a GPU, and only requires a working model.

3.2.2 Algorithm

- Generate a (height*width) number of images with occlusion regions centered at every pixel coordinate.
- Loop through every image into a trained network and record the top classification choices.
- Generate a map of the same height and width as the original image, where the top classifications each map to a random color.

Resize and Generate Set of Occluded Images

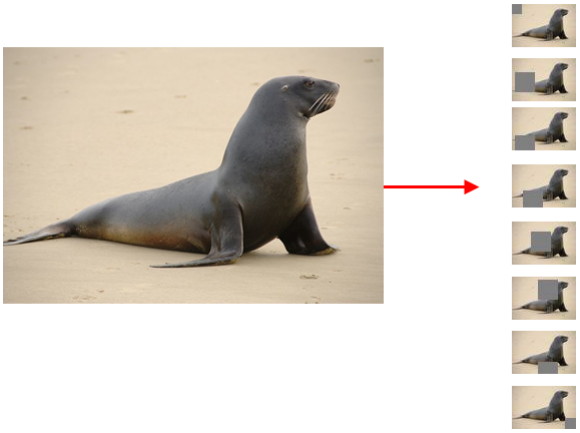

Figure 5. Generate a height*width number of images with occlusion regions centered at every pixel coordinate.

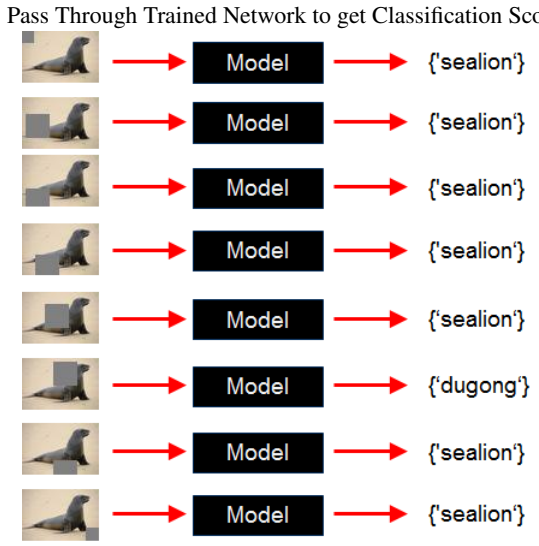

Figure 6. Pass through every image into any type of trained network that takes in images, and output in csv format the top classification choices

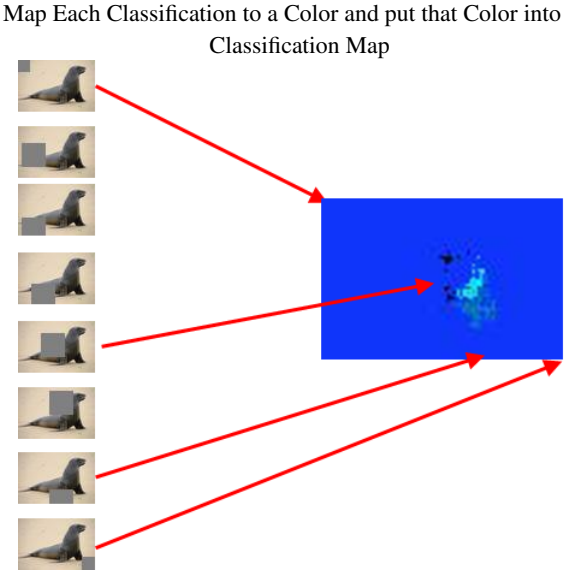

Figure 7. Generate a map of the same height and width as the original image, where the top classifications each map to a random color.

3.2.3 Implementation

This section explains the code, input parameters and the structure of the output heatmap images.

Input parameters

- Occlusion window radius(int) - required
- Occlusion window color(r,g,b) - default set to grey

Classification on occluded images

Occluded images are generated using an occlusion window on every pixel. We do not pad our output images, but ignore out of bounds occlusion generation. These are stored into the temp directory named based on the pixel coordinates, i.e. x0y0.jpg. In our particular case, we modified the Torch classification file for passing trained models that Facebook research group had released to output in a format like such [imageName, class1, score1] into *Output.txt*.

Color Map Generation

Classification color map is generated from the *Output.txt*, where each class is mapped to a unique color using a hash function. The hash function was tested for color collisions, and it was found to occur very rarely with high (¿1000) number of top classes.

Output format

- *Occluded Images* - Every occluded image is stored in a directory.
- *Classification scores* - Scores obtained from the trained CNN model.
- *Color map* - Image heat map created based on the classification.

3.2.4 Bottlenecks

- The Framework generates a large number of occluded images, based on the size of the image. This leads our algorithm to scale poorly.
- Loading the model and passing through height*width number of images is heavily dependent on the IO speed of the GPU used.

4. Experiments

4.1. Occlusion Sensitivity Based Classification Experiments

We performed occlusion based classification experiments on Facebook's implementation of a residual neural network architecture. The model was trained on ImageNet with 1000 classification classes. The same architectural units were used, and the group trained different models of varying lengths. One key feature of their implementation to improve performance on ImageNet is to use image scaling and aspect ratio augmentation, where images of varying scales and aspect ratios can be classified [8].

To perform the experiment on ResNet, we wanted to find the minimum resolution needed to work with our occlusion test. First,we tried a re-sized /textitsealion image of 64x43 resolution on the smallest network. From the classification map produced, we observed too much noise as show in Figure 8. Next, we tried the same resolution, but on a deeper network with a better classification error.

From Figure 10, we can conclude that the resolution of 96x64 had a good amount of data for these networks to localize around the sealion without a lot of noise while still

18-Layer ResNet with 64x43 and occlusion radius 10

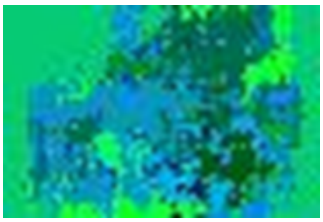

Figure 8. Noisy classifications, localizing around entire sealion image. *Note: the upper left corner color is the color of a correct classification for all of these images, and is a sealion.

200-Layer ResNet with 64x43 and occlusion radius 10

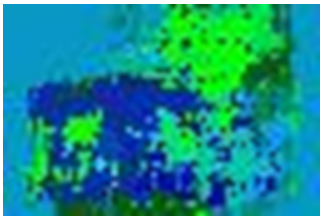

Figure 9. Noisy classifications even on a better network, localizing around entire sealion image.

200-Layer ResNet with 96x64 and occlusion radius 15

Figure 10. We increased the image data by 1.5 and the occluder radius by 1.5.

200-Layer ResNet with 96x64 and occlusion radius 5 and 6

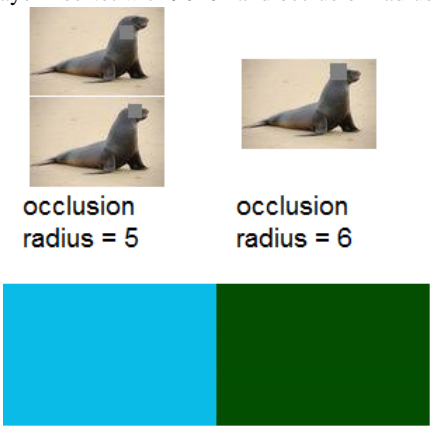

Figure 11. The generated classification maps give a top classification of sealion for both of these occluder radius. An example/s of the input images for each are given.

being computationally fast. Next, we try the 96x64 resolution images against varying occlusion window radius to

200-Layer ResNet with 96x64 and occlusion radius 25 and 30

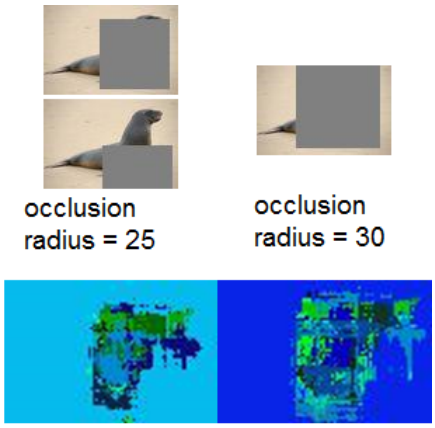

Figure 12. The generated classification maps can be seen to begin localizing with mis-classifications around the sealion. Occlusion radius of 30 is a window of 61x61. An example/s of the input images for each are given.

200-Layer ResNet with 96x64 and occlusion radius 35 and 40

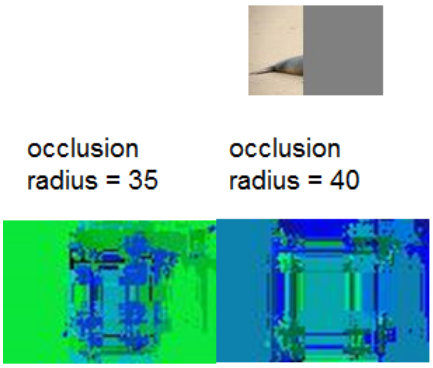

Figure 13. The occluder radius is too large at 35 and 40, with occluder windows larger than one side of the images used. This creates an artifact in the classification maps and can be seen in both of these, with the occluder of radius 40 giving very pronounced artifacts. An example for the occluder radius 40 is given to show this.

determine the best resolution for our classification map.

Later, we tried the same resolution on the 200 layered network, and got similar results as before. Since noise seemed like an issue for this network at the 64x43 resolution as seen on Figure 9. In an attempt to fix noise issue, we tried increasing all image parameters by 1.5 times.

Next, we perform experiments by increasing the occluder's radius to avoid mis-classification. As expected, if the radius is larger than one side of the image then artifacts are generated with this method. We chose the resolution 96x64 and radius of 25 going forward with our experiments, as this was seen to localize the mis-classifications well.

The figures presented in this paper confirm that using

18-Layer ResNet and 50-Layer ResNet with 96x64 and occlusion radius 25

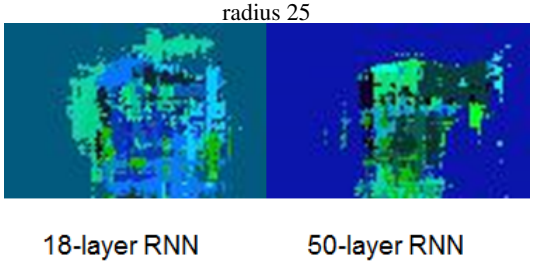

Figure 14. Smaller depth to the residual networks are confirmed to not localize as well around the ground truth of the image. This can be observed in the mis-classification region size being larger for the 18 layer network then the 50 layer network. Top classifications when edges are occluded are consistently the target class, the sealion.

101-Layer ResNet and 200-Layer ResNet with 96x64 and occlusion radius 25

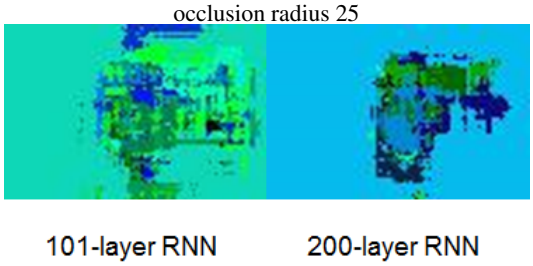

Figure 15. This confirms that the more depth to the residual network, the better it localizes around the ground truth. This can be observed in a tighter mis-classification region for the 200 layer network than the 101 layer network. Top classifications when edges are occluded are consistently the target class, the sealion.

a different localization methods can localize best with increased depth. It also confirms that there is a minimum data input needed for a given network to produce good classifications over noisy classifications. With our tests, it was also confirmed that there is an optimal occluder radius needed to get the best results from this method, and we go through the steps on how to find that. The top mis-classification for this method was the dugong, or manatee. The reason might be due to the color of the occluder used, as the sealion is darker in color and the dugong is grey in color.

5. Conclusion

We believe that we have made much progress on a problem that is crucial to Image Segmentation. The key is to reason together about segmentation and figure/ground relationships is taking advantage of Occlusion and Negative-Space Images. Further progress can be made by recreating affinity-CNN using Alex-Net. We also believe Occlusion Sensitivity based classification is a probable future direction to gain deeper insight into pixel-centric relations.

We have also shown that an previous test method used in [2] is effective in testing state-of-art networks for occlusion using mis-classification maps. We have confirmed the previously held beliefs that deeper networks are better for classification, and networks perform better with more data. We introduce a method for finding optimal image size and occluder radius in this paper. To avoid the manual tuning of parameters, we can vary the occluder radius by selecting a blob area to find the optimal ratio of misclassificationArea/totalArea on the image pyramid automatically. In the future, the framework can be adapted to work as an individual class heatmap generator to aid in finding the semantically relevant portions of the image for the particular network. It is also flexible such that different models in different frameworks can be tested against each other.

References

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

[2] P. Arbeláez, J. Pont-Tuset, J. T. Barron, F. Marques, and J. Malik. Multiscale combinatorial grouping. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 328–335, 2014.

[3] P. Felzenszwalb and D. Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59(2), 2004.

[4] G. Q. He. *Quantification of two Gestalt Laws using curve resconstruction.* PhD thesis, Concordia University, 2008.

[5] J.-J. Hwang and T.-L. Liu. Pixel-wise Deep Learning for Contour Detection. *ArXiv e-prints*, Apr. 2015.

[6] P. Isola, D. Zoran, D. Krishnan, and E. H. Adelson. Crisp boundary detection using pointwise mutual information. In *European Conference on Computer Vision*, pages 799–814. Springer, 2014.

[7] Z. Jia, A. Gallagher, Y. J. Chang, and T. Chen. A learning-based framework for depth ordering. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 294–301, June 2012.

[8] H. Kaiming, Z. Xiangyu, R. Shaoqing, and S. Jian. Deep residual learning for image recognition. *CVPR 2015*, arXiv:1512.03385.

[9] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. *CVPR 2015*, arXiv:1411.4038.

[10] M. Maire, T. Narihira, and S. X. Yu. Affinity CNN: Learning Pixel-Centric Pairwise Relations for Figure/Ground Embedding. *ArXiv e-prints*, Dec. 2015.

[11] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proc. 8th Int'l Conf. Computer Vision*, volume 2, pages 416–423, July 2001.

[12] Muthukrishnan and M.Radha. Edge detection techniques for image segmentation. *International Journal of Computer Science Information Technology (IJCSIT)*, 3(6), 2011.

[13] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, 22(8):888–905, 2000.

[14] X. Y. Stella. Angular embedding: from jarring intensity differences to perceived luminance. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 2302–2309. IEEE, 2009.

[15] S. C. Turaga, J. F. Murray, V. Jain, F. Roth, M. Helmstaedter, K. Briggman, W. Denk, and H. S. Seung. Convolutional networks can learn to generate affinity graphs for image segmentation. *Neural computation*, 22(2):511–538, 2010.

[16] U. von Luxburg. A Tutorial on Spectral Clustering. *ArXiv e-prints*, Nov. 2007.

[17] S. Yu. Angular embedding: A robust quadratic criterion. *IEEE transactions on pattern analysis and machine intelligence*, 34(1):158–173, 2012.

[18] S. Yu and J. Shi. Object-specific figure-ground segregation. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 2, pages II–39. IEEE, 2003.

[19] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *European Conference on Computer Vision*, pages 818–833. Springer, 2014.