Homework 0: Alohomora!
Phase 1

Naitri Rajyaguru
Master of Engineering in Robotics
University of Maryland, College Park
Email: nrajyagu@umd.edu

I. Introduction

Boundary detection is an important problem in computer vision. Early stages of vision processing includes extracting features out of image which is important for analyzing. There can be many features and among them one such is Edge. Edge detection is usually the first step in recovering information from images. Because of this importance, Edge detection holds active part in research area. In this report, the basic properties of an image like texture and color will be used for detection of edge. It provides evidence of how it outperforms classical edge detectors like Canny and Sobel by suppressing edges resulting from textures. The report also discusses detailed implementation of PB-lite edge detector.

II. PB-Lite Methodology

This section discusses the outline of the process for probability based edge detection algorithm. The figure below represents procedure for the same. Overall, the main steps includes: (1) Low level feature extraction by filtering (2) Extracting texture, brightness and color features of an image by clustering (3) pb-score assignment using combination of classical detectors and textures.

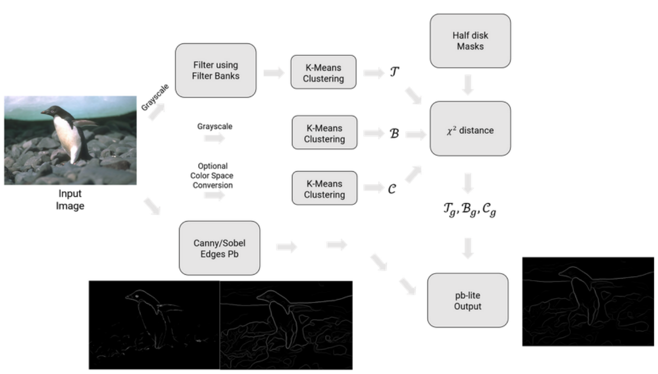

Fig. 1. Pb-lite Pipeline Overview

A. Filter Banks

Filtering is mainly used for low-level extraction of features. To capture all kind of textures like of various scales, orientations present in an image, we use different types of filter banks. Here, three types of filter banks are used: Oriented DoG filter, Leung-Malik Filter and Gabor Filter.

1) Oriented Derivative of Gaussian Filter: Difference of Gaussian is a feature enhancement algorithm which takes difference of two gaussians with same mean but different standard deviation. A DoG filter can be created by taking derivatives in x and y direction using Sobel filter and convolving it with a Gaussian kernel. As shown in the figure below, the DoG filter bank is created at 2 different scale [3,5] with 16 orientations spaced equally form 0 to 360. The number of filters in this bank will be scale * orientations. For current implementation, number of filters are 32.

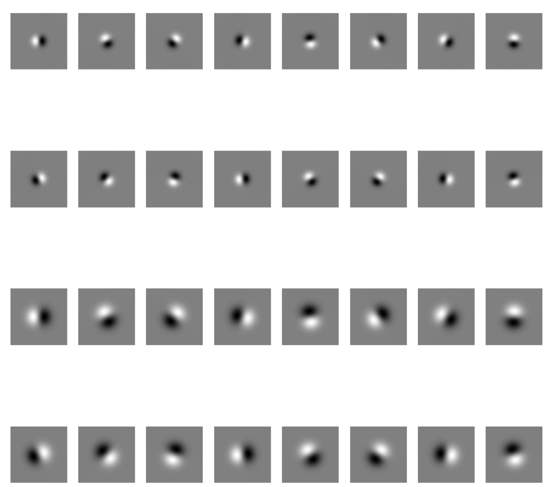

Fig. 2. DoG Filter Bank

2) Leung-Malik Filter: Leung-Malik filters are multi scale and multi oriented filter bank with 48 filters. It consists of first and second derivative of Gaussians at 6 orientations and 3 scales; 8 Laplacian of Gaussian filters at two sigmas and 4 gaussians. It consists of two versions that is LM small and LM large with different scale factors.

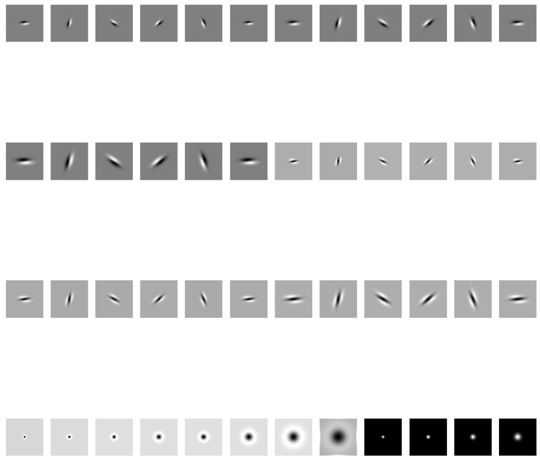

Fig. 3. Leung-Malik Small Filter Bank

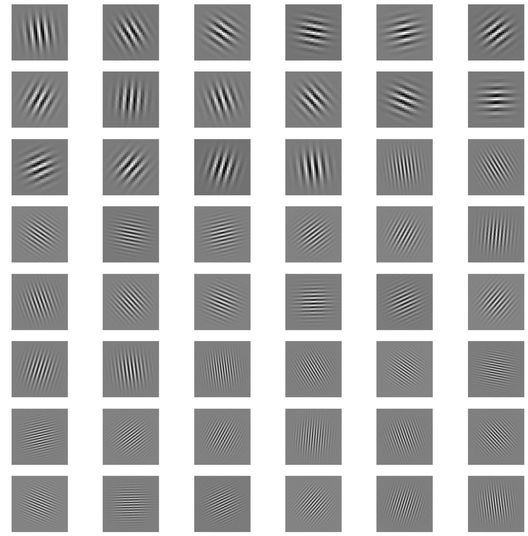

Fig. 4. Gabor Filter Bank

3) Gabor Filters: A Gabor Filter is sinusoidal signal of particular frequency and orientation, modulated by a Gaussian wave. Different parameters of filter controls shape and size of it. The equation for the same is given below

$$g(x, y; \lambda, \theta, \psi, \sigma, \gamma) = \exp\left(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}\right)$$
$$\exp\left(i\left(2\pi\frac{x'}{\lambda} + \psi\right)\right)$$

where

$$x' = x\cos\theta + y\sin\theta$$

and

$$y' = -x\sin\theta + y\cos\theta$$

$\lambda$is the wavelength of the sinusoidal component controlling width of strips of Gabor function θ controls orientation of the Gabor function. Ψ The phase offset of the sinusoidal function. σ is the sigma/standard deviation of the Gaussian envelope controlled how many stripes can be in the function. γ is the aspect ratio controlling height of the function.

B. Texton, Color and Brightness Map

1) Texton Map: The filters explained in previous section are used to detect textures in an image.Any combination of filter banks can be used for extraction of texture properties but here all the three filter banks are combined which results into vector of filter responses. As filter response vectors are generated, they are clustered together using k-means clustering with K = 64. Hence, pixels with similar textures forms one cluster. The output of k-means clustering is Texton Map.

2) Color Map: The color values of an image are clustered together using k-means clustering with K = 16. Color Map contributes the color property R,G and B that each pixel has.

3) Brightness Map: The brightness values of an image are clustered together using k-means clustering with K = 16. Brightness Map contributes the changes in intensity of light property for each pixel.

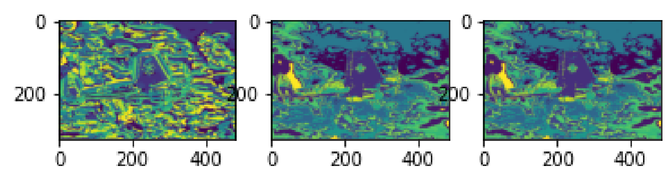

Fig. 5. Texton, Color and Brightness Map for Image 1

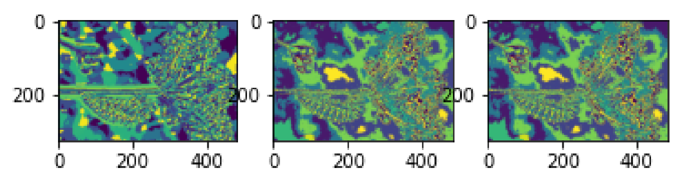

Fig. 6. Texton, Color and Brightness Map for Image 2

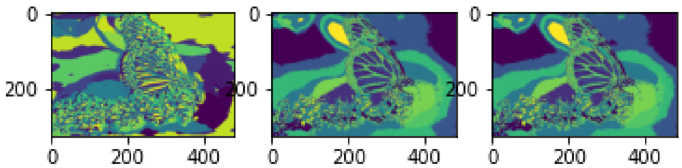

Fig. 7. Texton, Color and Brightness Map for Image 3

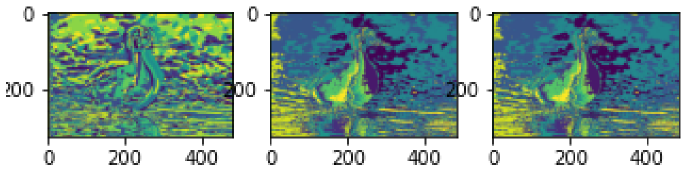

Fig. 8. Texton, Color and Brightness Map for Image 4

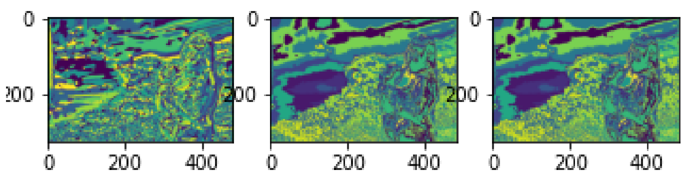

Fig. 9. Texton, Color and Brightness Map for Image 5

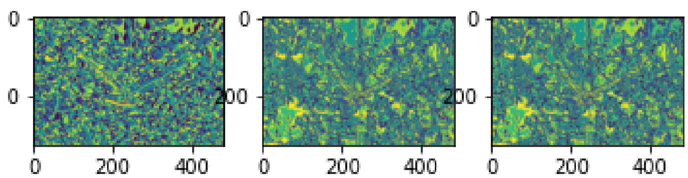

Fig. 10. Texton, Color and Brightness Map for Image 6

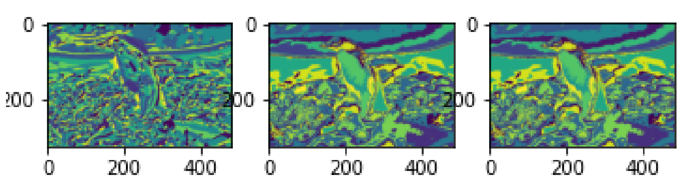

Fig. 11. Texton, Color and Brightness Map for Image 7

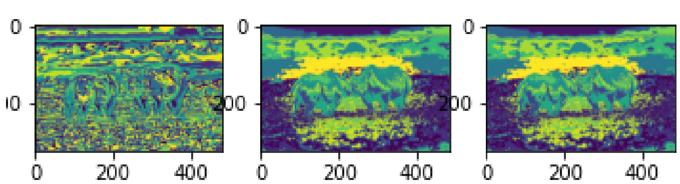

Fig. 12. Texton, Color and Brightness Map for Image 8

C. Texton, Color and Brightness Gradient Maps

The Texton, Color and Brightness map generated in the previous section represents each of the feature at every pixel.

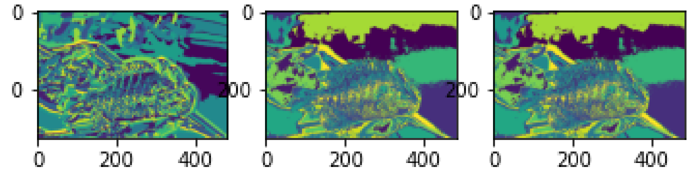

Fig. 13. Texton, Color and Brightness Map for Image 9

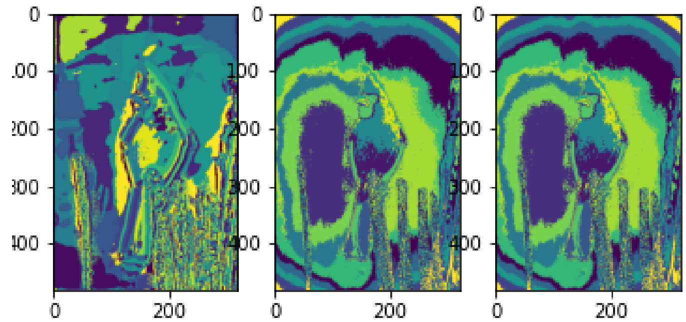

Fig. 14. Texton, Color and Brightness Map for Image 10

The gradient measurement is performed to know how much all features distribution is changing at a given pixel. For this purpose, half-disc masks are used. Half-disks masks are binary images of half circle with different scales and rotation values. The gradients are generated by convolving with left and right half-disc pair with the map created in previous step.The chi-squared distance between the mask pair and binary image is computed. The chi-squared distance is used for comparing two histograms. It is computed as follows:

$$\chi^2(g,h) = \frac{1}{2}\sum_{i=1}^{K}\frac{(g_i - h_i)^2}{g_i + h_i}$$

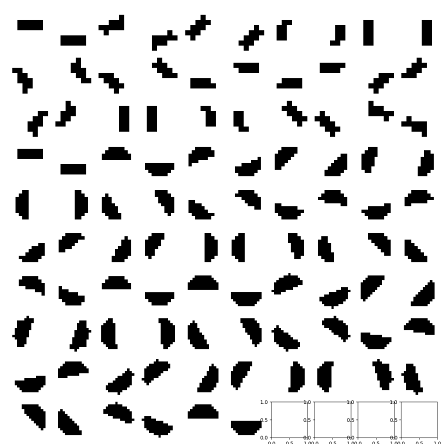

Fig. 15. Half-disc mask

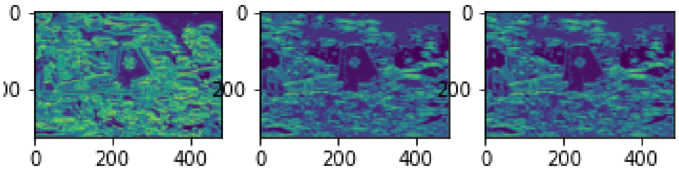

Fig. 16. Texton, Color and Brightness Map for Image 1

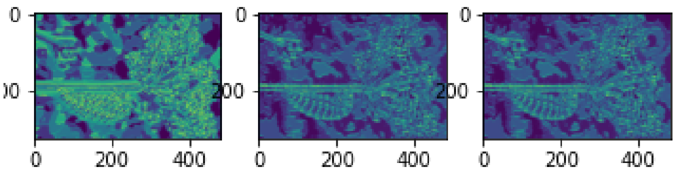

Fig. 17. Texton, Color and Brightness Gradient for Image 2

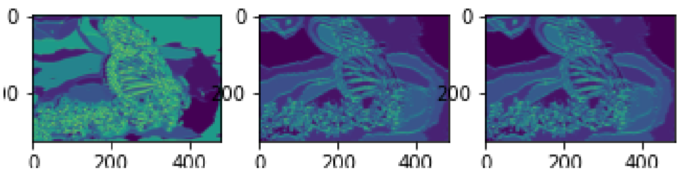

Fig. 18. Texton, Color and Brightness Gradient for Image 3

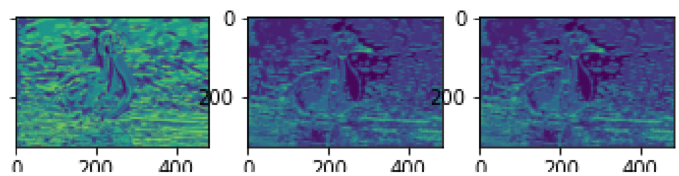

Fig. 19. Texton, Color and Brightness Gradient for Image 4

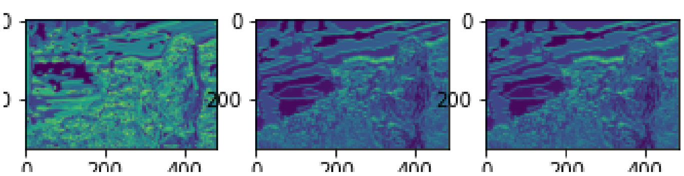

Fig. 20. Texton, Color and Brightness Gradient for Image 5

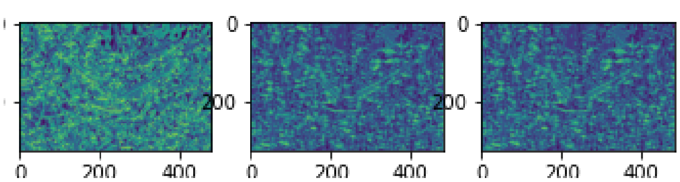

Fig. 21. Texton, Color and Brightness Gradient for Image 6

D. PB-Lite Output

The gradient maps which are generated are combined with classical edge detectors like Canny and Sobel baselines using this equation:

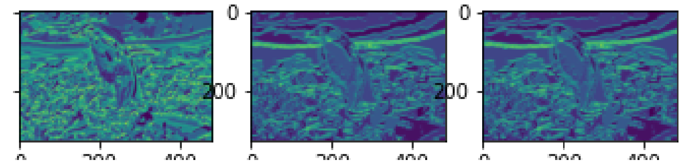

Fig. 22. Texton, Color and Brightness Gradient for Image 7

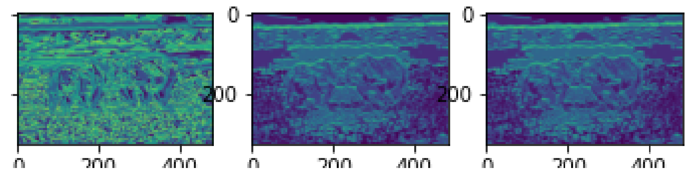

Fig. 23. Texton, Color and Brightness Gradient for Image 8

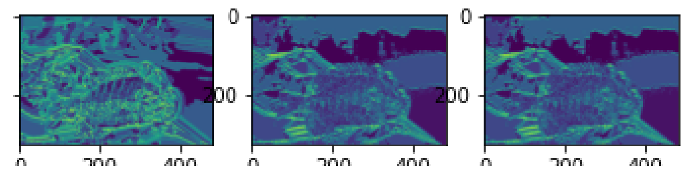

Fig. 24. Texton, Color and Brightness Gradient for Image 9

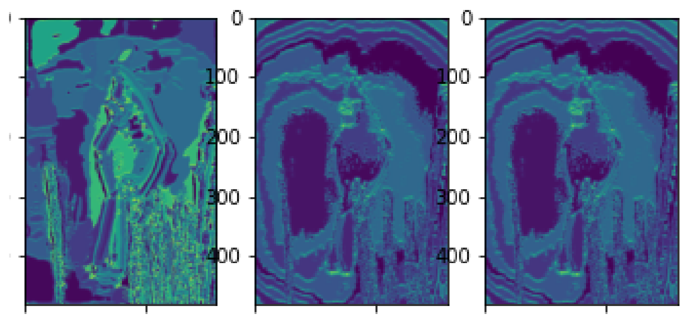

Fig. 25. Texton, Color and Brightness Gradient for Image 10

$$PbEdges = \frac{(\mathcal{T}_g + \mathcal{B}_g + \mathcal{C}_g)}{3} \odot$$
$$(w_1 * cannyPb + w_2 * sobelPb)$$

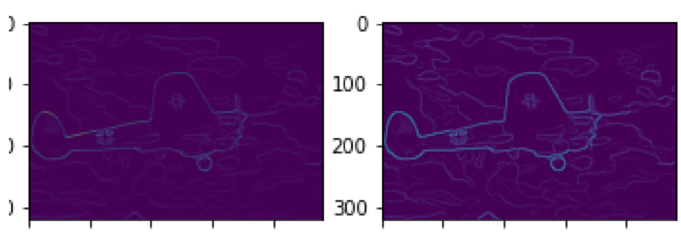

Fig. 26. Pb-Lite and Canny Edge for Image 1

III. Conclusion

The weights of Canny and Sobel detector are kept from combination of trial and error. It is evident that if weight

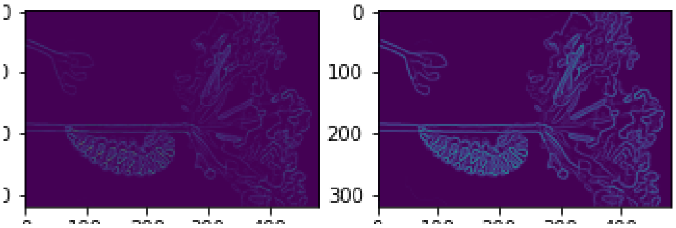

Fig. 27. Pb-Lite and Canny Edge for Image 2

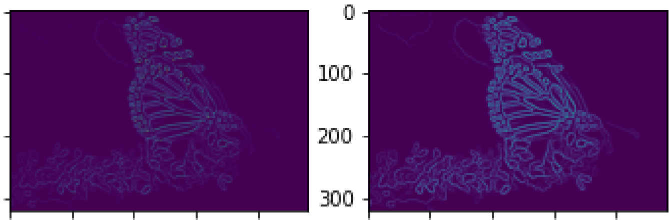

Fig. 28. Pb-Lite and Canny Edge for Image 3

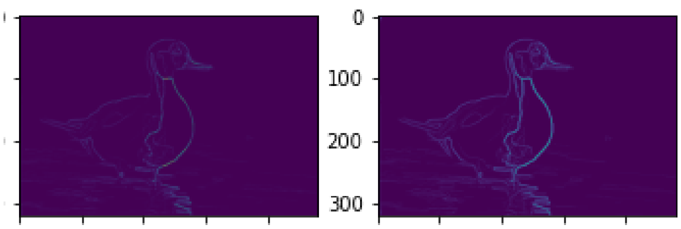

Fig. 29. Pb-Lite and Canny Edge for Image 4

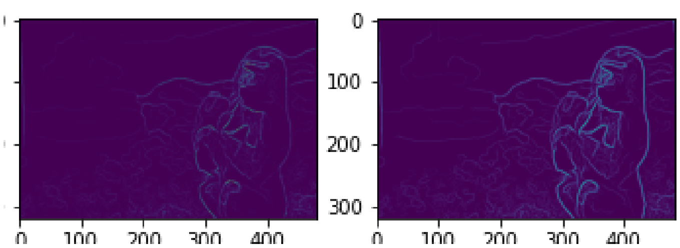

Fig. 30. Pb-Lite and Canny Edge for Image 5

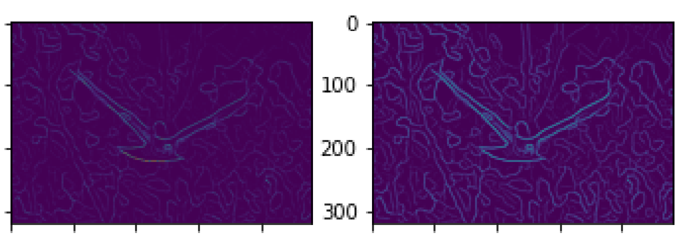

Fig. 31. Pb-Lite and Canny Edge for Image 6

of Canny edge detector is kept more, then edges are more stronger but it suppresses background completely while Sobel detector preserves overall intensity of pixel. We can conclude that Pb-Lite detector performs well in comparission to other

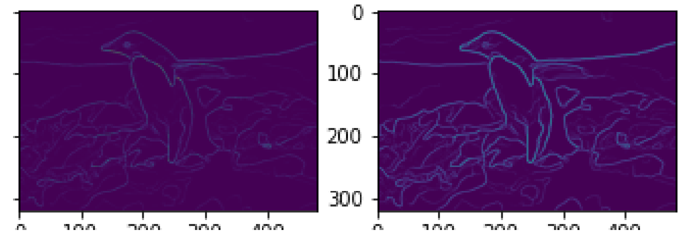

Fig. 32. Pb-Lite and Canny Edge for Image 7

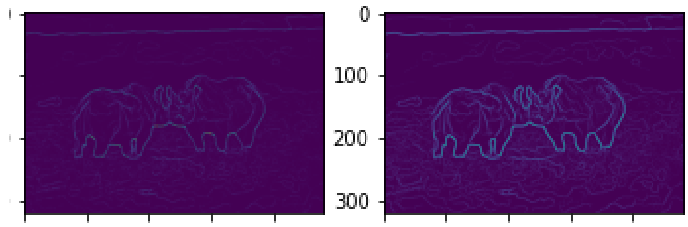

Fig. 33. Pb-Lite and Canny Edge for Image 8

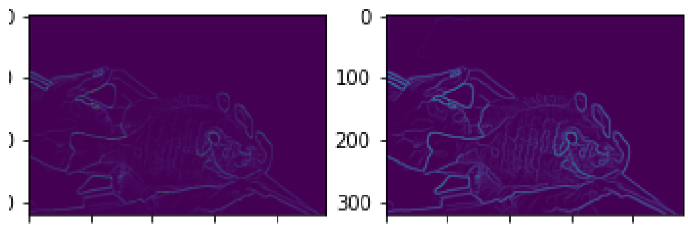

Fig. 34. Pb-Lite and Canny Edge for Image 9

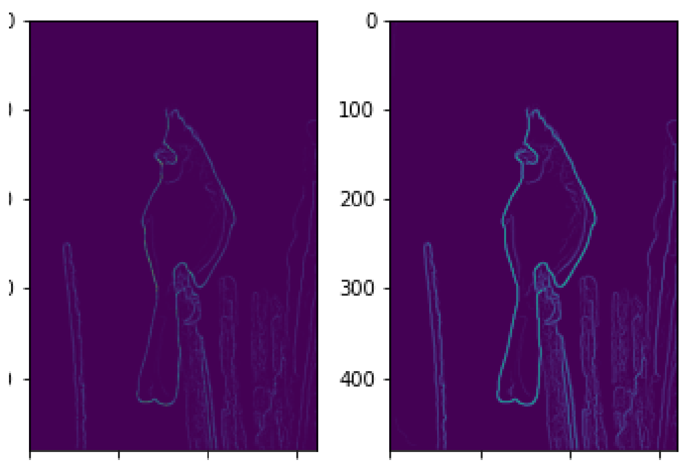

Fig. 35. Pb-Lite and Canny Edge for Image 10

detectors as it provides more control over many features which are useful for edge detection.

References

[1] T. Leung and J. Malik.Representing and recognizing the visual appearance of materials using three-dimensional textons.International Journal of Computer Vision, 43(1):29-44, June 2001.

[2] https://medium.com/@anuj_shah/through-the-eyes-of-gabor-filter-17d1fdb3ac97