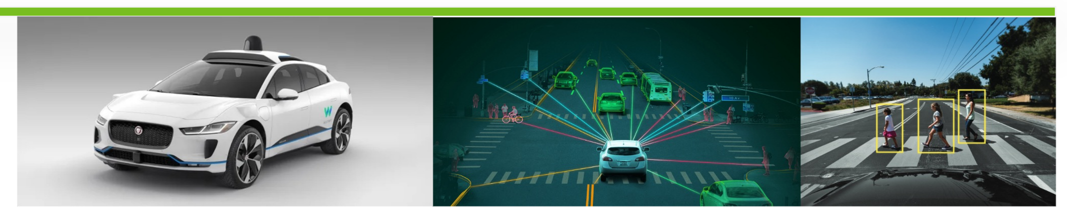
Evaluating Image Edge Mapping Operators as it Pertains to Autonomous Vehicle Object Detection Algorithms

CISA Student Showcase

Michael Gross and Dr. Michelle Zandieh

See Full Report

Introduction



Background

With the immersion of autonomous vehicles, the need for quicker processing of data was identified to be the bottleneck of object recognition. More immediate detection leads to faster decisions by internal automotive computers, which have a higher safety margin on the road. Most onboard cameras capture high-definition, color video requiring more extensive interpretive computations. However, current technologies utilize edge mapping techniques, reducing the data size of the processed images by converting them from color to black and white. Additionally through this process, a matrix operator is applied to the intensity of small groups of pixels to extract the most critical features of the image: the edges. As a result, less relevant properties are filtered out and the content of the image can be processed much easier.

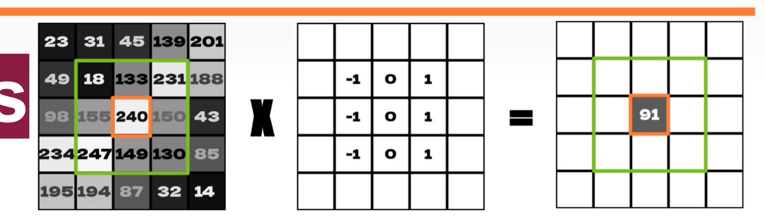
Research Question

Which of the commonly used matrix operators is best suited to identify objects in edge-mapped images when evaluating the run-time and confidence rating?

Objectives

1. Determine matrix operators commonly used in image manipulation
2. Locate large image databases containing the following categories:
   a. Car
   b. Trailer Truck
   c. Pedestrian
   d. Bicycle
   e. Street Sign
3. Design and Construct MATLAB code to use the edge mapping operators to create convolutional images
4. Utilize AlexNet, a MATLAB machine learning architecture toolbox, to train the convolutional neural network (CNN) to correctly identify objects in edge-mapped images
5. Evaluate the efficiency of each edge mapping operator to convolute each image and the effectiveness of the CNN to correctly identify the objects via the following quantitative factors:
   a. Time Elapsed (sec)
   b. Confidence Rating (%)
6. Analyze resulting data using a weighted decision matrix to determine the best-performing matrix operator, given the criteria.

Edge Mapping Process



Purpose

The operator matrices will be used to create a convoluted image via element-wise multiplication with subsets of the original image. As a result, the less important data (ie. non-edges) will be far less visible after this process, and edges that are more critical become more visible. The contrast of the dark and light colors creates what's called an "edge map".

Mathematical Methodology

Prewitt:

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \quad G_y = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$

Roberts Cross:

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad G_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

Sobel:

$$G_x = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \quad G_y = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

Horizontal and Vertical Convolution:
Sums the absolute value of element-wise multiplication of the operator and subset of the original image. The result is localized to the center location (i, j) of the new partial convoluted image.

$$\frac{\partial f}{\partial x}_{(i,j)} = \Sigma\left|G_x \circ m_{(i,j)}\right| \qquad \frac{\partial f}{\partial y}_{(i,j)} = \Sigma\left|G_y \circ m_{(i,j)}\right|$$

Total Convolution:
The magnitude of the partial components (horizontal and vertical) are calculated.

$$\nabla f = \sqrt{\frac{\partial f}{\partial x}^2 + \frac{\partial f}{\partial y}^2}$$

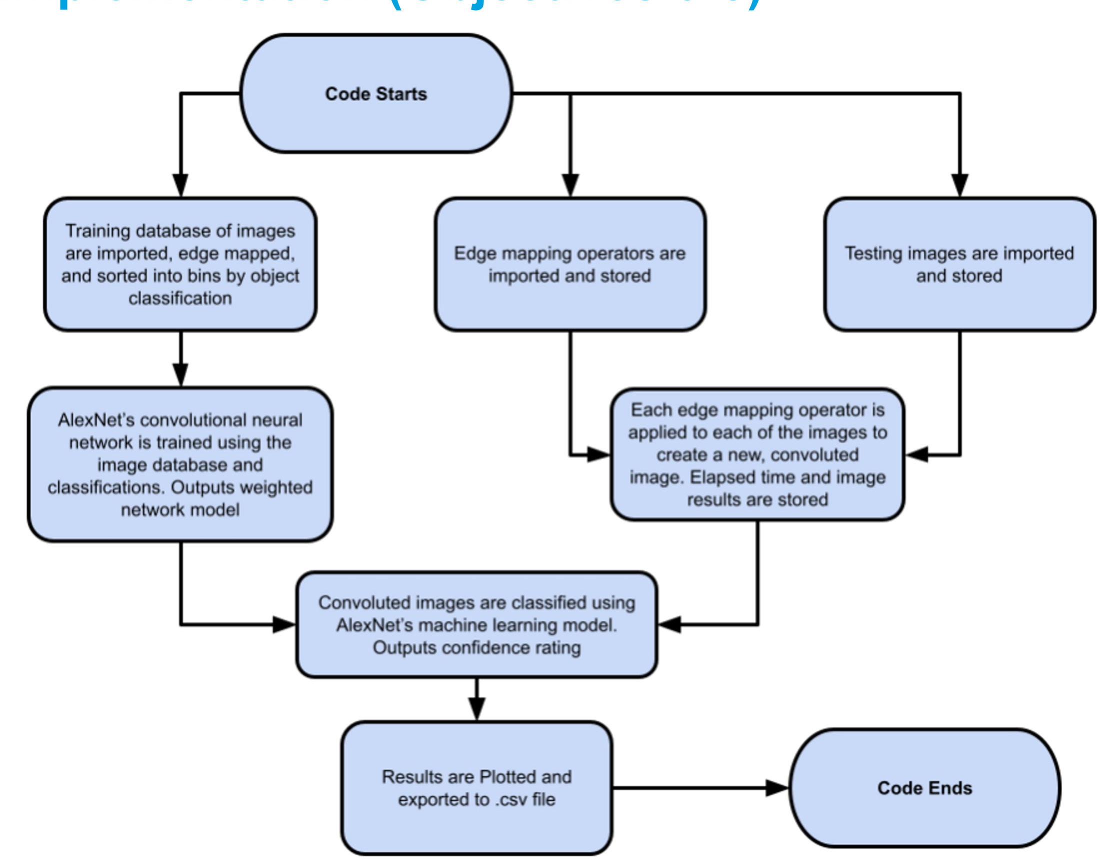
Variable	Definition
i	row number index of original image
j	column number index of original image
n	total number of rows of original image
k	total number of columns of original image
G	edge mapping operator partial gradient (eg. Prewitt, Sobel, Roberts Cross)
m	subset of original image (see example formats below)

$$\begin{bmatrix} m_{i-1,j-1} & m_{i,j-1} & m_{i+1,j-1} \\ m_{i-1,j} & m_{i,j} & m_{i+1,j} \\ m_{i-1,j+1} & m_{i,j+1} & m_{i+1,j+1} \end{bmatrix} \quad \begin{bmatrix} m_{i-1,j-1} & m_{i,j-1} \\ m_{i-1,j} & m_{i,j} \end{bmatrix}$$

Magnitude Adjustment:
The edge map is adjusted to fit within the boundaries of pixel intensity (0-255). The max value of the dataset is used to linearly adjust the model.

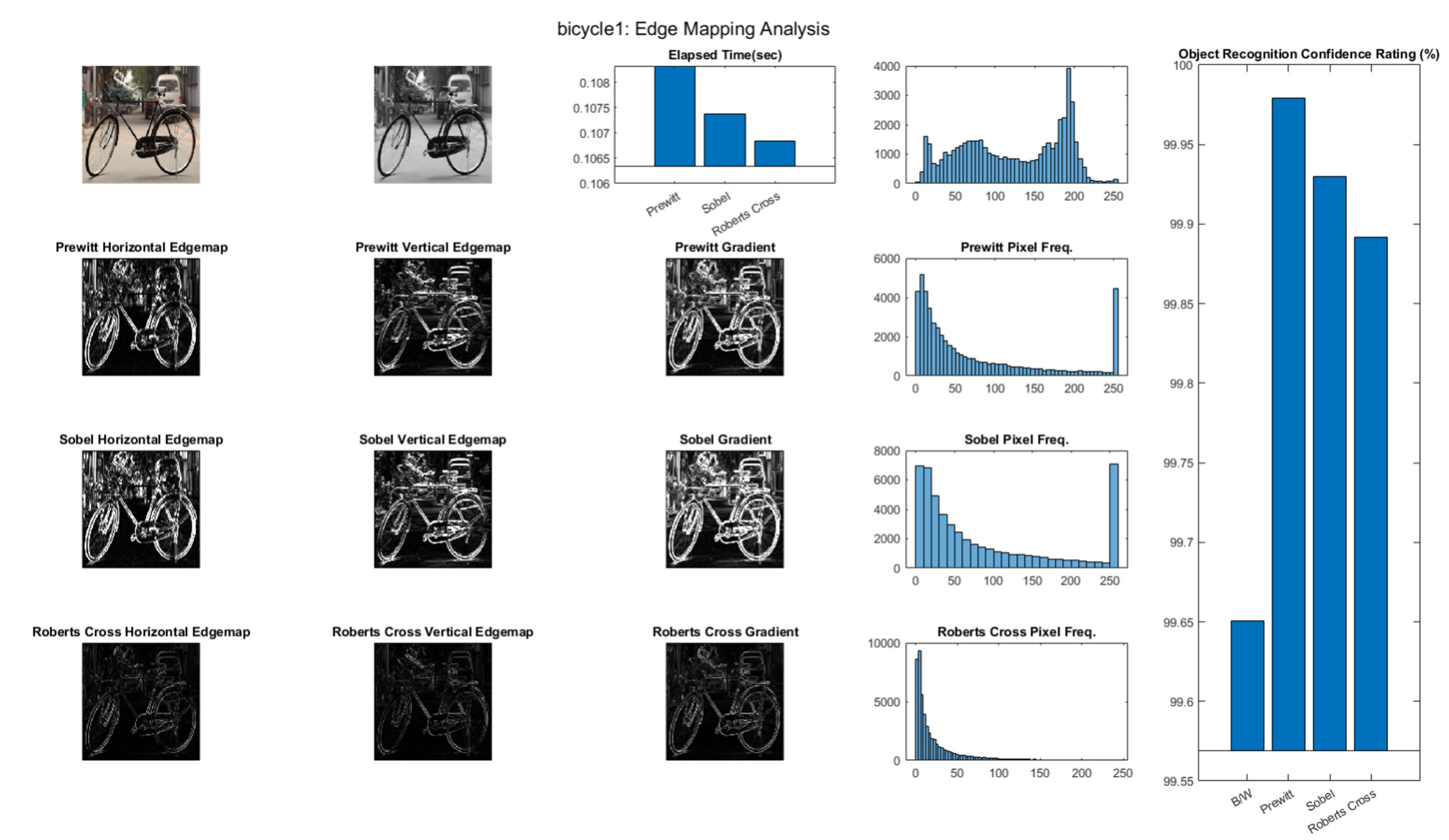
$$\nabla f_{mag\,(i,j)} = \left|\nabla f_{(i,j)} * \frac{\nabla f_{max}}{255}\right|, \; 255 \leq \nabla f_{max} \leq \sqrt{2\left(255^2\right)}$$

MATLAB Implementation (Objectives 3-5)



Abbreviated Results

Original Image	Orginal Image Size (Width)	Orginal Image Size (Height)	Total Image Size (Pixels)	Operator	Final Image	Criteria: Elapsed Time (sec)	Criteria: Object Detection Confidence Rating (%)	Criteria Ranking: Elapsed Time (sec)	Criteria Ranking: Object Detection Confidence Rating (%)	SCORE	OVERALL RANK
	227	227	51529	Roberts Cross		0.1083386	99.98%	4	3	8.5	1
	227	227	51529	Roberts Cross		0.1053504	99.89%	1	6	10	2
	227	227	51529	Sobel		0.1099609	100.00%	9	1	10.5	3
	227	227	51529	Prewitt		0.1087171	99.98%	7	4	13	4
	227	227	51529	Prewitt		0.1125406	100.00%	11	2	14	5



Conclusion

Given that the convolutional neural network, AlexNet, requires the input data to be 227 by 227 pixels in size, it's easy to see how minute the difference is between each of the elapsed times and how this may be attributed to minor changes of internal computing speeds whilst running the test. When the edge mapping process was tested independently on the larger original image sizes, ranging from tens of thousands to millions of pixels, there was a larger difference between the operator performances. The larger operators (Sobel and Prewitt) are able to perform operations on more of the image at a given time, thus reducing the time required to generate a convoluted image. Despite the top listed performance of the Roberts Cross, Prewitt had the best overall consistency across the range of data and categorization of image objects. Shown in the example graphic, the resulting Prewitt edge map detects critical edges without over saturating the image with unnecessary information. This can be contrasted with the adjacent Sobel and Roberts Cross edge maps.

References

Jain, R. C., Kasturi, R., & Schunck, B. G. (1995). Edge Detection. In Machine Vision (pp. 140–185). Chapter, McGraw-Hill.

Object recognition. MathWorks. (n.d.). Retrieved October 25, 2022, from https://www.mathworks.com/solutions/image-video-processing/object-recognition.html

Stanford Artificial Intelligence Laboratory. (n.d.). Retrieved October 12, 2022, from https://ai.stanford.edu/~syyeung/cvweb/tutorial1.html