
ECE 420 Final Project Prototype ID Card Scanner

_____ Katerina Baxter and Binh-Minh Nguyen _____

Project Motivation

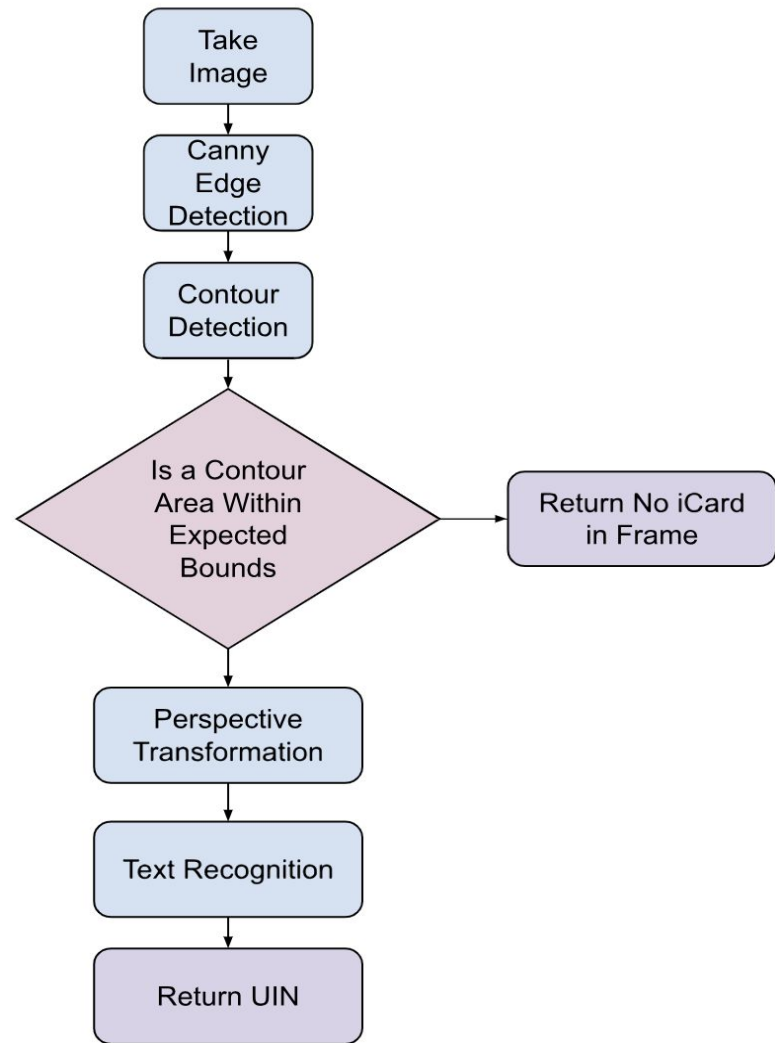
Image processing algorithm to detect the presence of an iCard in the frame and automatically read off and save the ID card number

Potential Applications:

- RSO email signup at quad day
- ECE OpenLab equipment checkout



Algorithm Overview



Canny Edge Detection

Pre-processing: dilate image

- 1) Gaussian Kernel applied to smooth image
- 2) Gradient in the X and Y direction taken using Sobel operators
- 3) Edge magnitude and direction (round to 0, 45, 90, 135) at each pixel is determined
- 4) Non-maximal suppression removes pixels that are not the local maximum in a small length in the each direction
- 5) Hysteresis thresholding sets the edges to 0 or 1 (high and low threshold to keep continuous edges)

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

Gaussian Kernel

g_x			g_y		
-1	0	1	-1	-2	-1
-2	0	2	0	0	0
-1	0	1	1	2	1

Sobel Operators applied to get G_x and G_y

$$\sqrt{G_x^2 + G_y^2} \quad \arctan\left(\frac{G_x}{G_y}\right)$$

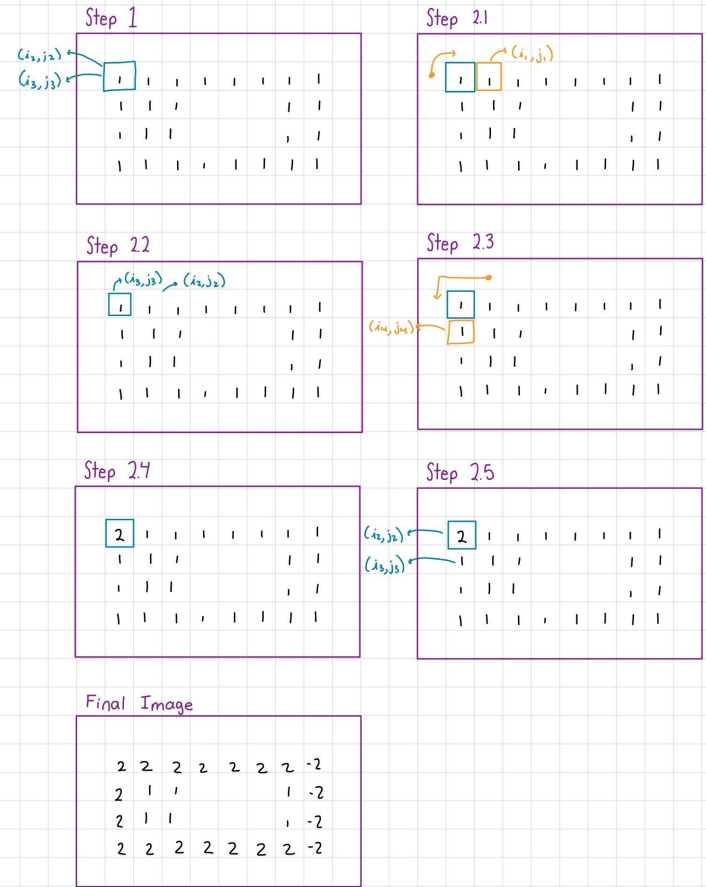
Equations for magnitude(left) and direction (right) of each pixel



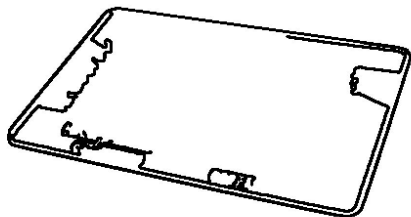
Suzuki's Contour Tracing Algorithm

- 1) Move along an image from left to right for each row until an outer border is detected (pixel value = 0 and previous pixel = 1)
- 2) Trace the border by moving clockwise around each pixel and setting the detected edge pixels to a value of 2. Continue until we return to the start point
- 3) Record that we have detected an edge in the row and continue scanning. Stop scanning at the bottom right of the image.

OpenCV Contour Tracing -> array with the points of each contour



Contour Selection and Perspective Transformation



1) convexHull: fills in the gaps in the detected contours

2) Choose contour with maximum area

3) approxPolyDP: approximate polygon curves with a specified precision, recursively divides the lines to remove points as long as they are within the specified error distance from the new line

4) Use cross product to make sure that the points of the contour are clockwise starting with the long edge

5) getPerspectiveTransform: applies a matrix transformation to map the source to a specified output

6) Crop the image to just the transformed iCard

Text Recognition



- 1) Crop the perspective transformed image to the text of interest
- 2) Apply a gaussian blur to the text
- 3) Otsu's Threshold: calculate threshold value by finding two bimodal histogram values then using their mean as the threshold value to compute a threshold image
- 4) Tesseract OCR to get text from the image

Prototype Testing



Accuracy Metrics - Python Prototype

ICard	Was the Camera Moving?	Number of Frames Analyzed	Frame Detection Accuracy	UIN Detection Rate	False UIN Rate
0	yes	205	88.83%	69.40%	2.92%
0	no	175	97.14%	98.82%	0.01%
1	yes	359	83.14%	25.8%	0.00%
1	no	161	100.00%	57.76%	0.00%
2	yes	272	58.09%	68.35%	1.47%
2	no	171	93.57%	65.63%	7.60%

Testing Plan - Android Application

The Android implementation will scan until an iCard is found and a UIN is identified.

Validation Metrics:

- UIN accuracy: rate at which the UIN we read off and present to the user is the correct number
- Time to UIN identification: the number of images / time it takes until the UIN is identified

Timeline and Deliverables

11/13

Prototype
Demonstration

11/20

Final Project
Proposal

12/9

Final Report

11/21

Milestone 1

- Live display the frame with a border drawn around an iCard if detected

12/04

Milestone 2

- Text Recognition to identify the UIN from the detected iCard
- Button to start and stop image collection

12/11

Final Demo

- Refined User Interface and Testing Complete