

Barcode Detection and Decoding for Fraud Prevention

Introduction

The goal of this project was to develop a basic implementation of barcode detection based on computer vision and image processing techniques using Python and Open Computer Vision Library (OpenCV)

Background

When I read an article in August 2019 about a man who made more than \$1 million in scam using fake barcodes at big-box retailers, I looked more into the barcode scams. Soon after reading that, I found the problem statement for my first image analysis project. Having worked with a Computer Vision peer, I always found the image analytics fascinating.

Project methodology

The project has two bases:

1. Image pre-processing and detection
2. Image decoding

The ***pre-processing*** involves four major parts: Calculation of Sobel gradient, making the image blur, kernel application, and erosions & dilations.

A good practice that I learned in my Image Processing class in undergrad was to start a project as: load the image, resize it and convert it to grayscale

Step 1:

Calculated the x & y gradients of the training barcode images using the OpenCV gradient filter Sobel and subtracted the y-gradient from the x-gradient. I particularly used Sobel to make the image more resistant to noise.

Step 2:

Blurred the image and applied a threshold to the image to later apply the kernel

Step 3:

Constructed a closing kernel and apply it to the thresholded image with one input images

Step 4:

Performed a series of erosions and dilations. TO complete the process flow, I found “k” biggest contours in the thresholded image and sorted the contour. Lastly, computed the rotated bounding box of the largest contour to draw a bounding box around the detected barcode, and display the image

Barcode decoding

To read the barcode from image files, I used ZBar BarCode Reader in Python. I created a reader with the ImageScanner class from zbar. Finally, the core of the project: scanned the image for barcodes and parsed the argument to output barcode from the image.