IC Detection on a PCB

Abstract: This paper explains an image processing system that detects Integrated Circuits (IC's) in an image of a Printed Circuit Board (PCB). Also it describes Algorithm to identify the chip using Optical Character Recognition (OCR). Here we used various Google downloaded images and OpenCV in Python.

I. INTRODUCTION

Identifying integrated circuits (ICs) on a printed circuit board (PCB) can be a difficult task for a human being. Our project aims to overcome this limitation by automated chip detection using OpenCV algorithms in Python language.

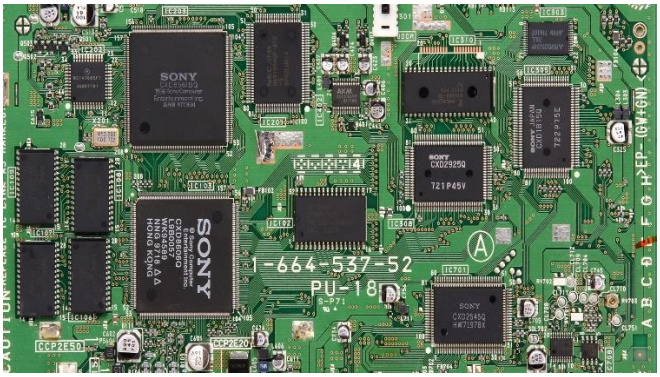

Fig. 1. An example PCB image

II. ALGORITHM DESCRIPTION

There are two main parts of our algorithm: Localizing each IC on the PCB, extracting and saving it. Then we use Tesseract OCR engine to read the labels of each detected IC.

A. PREPROCESSING

Before we apply any algorithms to our image we have to process it to obtain proper image. To save processing time we will resize the image maintaining the aspect ratio. After that remove noise from the image using Gaussian Blur.

B. SEGMENTATION

First we convert the RGB image to HSV (Hue, Saturation, Value) colour space. The H channel creates a mask for all the integrated circuits on the board. We then threshold the image to get a binary image However, it is not perfect.

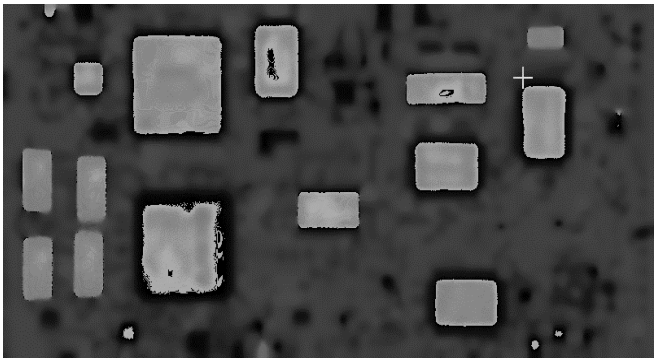

Fig. 2. H channel

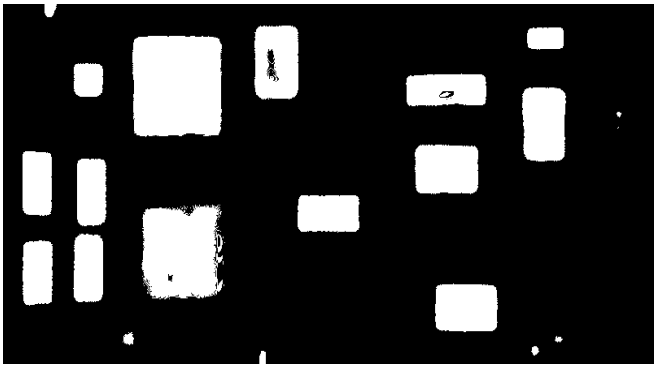

Fig. 3 Threshold image of H channel

C. MORPHOLOGICAL OPERATIONS

As we can see the thresholding is not perfect so we performed morphological operations (first dilate and then erode the dilated image) on the threshold image.

Fig. 4. Threshold image after morphological operations

D. EXTRACTION

To extract the integrated circuits from PCB we first have to find edges of the ICs. We used Canny Edge Detection algorithm to find edges. After that we found contours for those edges. These contours also included small areas on PCB that were not IC so, we had to exclude them. We excluded every contour which had less area than the mean area of contours. We found bounding rectangle for remaining contours and cropped that from the original image. This gave us all the integrated circuits on the PCB.

Fig. 5. Example of extracted IC

E. OPTICAL CHARACTER RECOGNITION (OCR)

Once the ICs were saved on the disk, we identified the ICs by reading their labels using OCR. For OCR, we used Tesseract OCR engine. If we pass the ICs directly to Tesseract OCR engine it will fail to read the labels so we pre-processed the image by smoothing and thresholding it. We used Otsu's algorithm to threshold the image. We also eroded the image to make edges clear.

Fig. 6. IC after thresholding and eroding

We then passed this image to Tesseract OCR engine and got the labels in text format.

For figure 6 we got the following output:

```
SONY
CXDZ9250

721P45V
```

Here, we can see that output of OCR is not perfect but the accuracy level is satisfactory.

Fig.7. Example of IC which was not perfectly identified due to bad lighting

For the IC in figure 7 we got no output from the OCR.

III. CONCLUSION

Although, we are getting good accuracy for this particular image but for images taken in dim lighting or images that contain very small ICs or other circuit elements the result is not good. Our algorithm works when we apply it on a high resolution well lit printed circuit board image.