IC Image Analysis for Counterfeit Detection

Logan Schmalz
EECS
University of Kansas
Lawrence, KS
loganschmalz@ku.edu

Aidan Schmelzle
EECS
University of Kansas
Lawrence, KS
schmelzle@ku.edu

Will Thomas
EECS
University of Kansas
Lawrence, KS
30wthomas@ku.edu

Abstract—**Counterfeit ICs pose a threat to end users who may expect certain reliability and security from the ICs they purchase. Various machine learning and image processing techniques have been deployed to defend against the manufacture and sale of counterfeit hardware. Unfortunately, the detection of counterfeit ICs can be very costly and difficult. The arms race between malicious engineers and those combatting their attacks has provided some foundational methodologies for us to build from. Through this project, we hope to gain a deeper understanding of detection analysis with an image processing program. We have two primary objectives: to detect and mark defective IC pins, and secondly, to detect and report IC markings. We hope to use our results to identify counterfeit ICs with a high degree of accuracy while maintaining a low cost.**

Keywords—object detection, image processing, computer vision, edge detection, optical character recognition

I. Objective and Motivation

While improvements in technology and manufacturing have contributed to the booming computing industry, there are associated quality and security concerns. Rapid production times spur design houses to outsource production. Further, inauthentic parts filter into the market through different avenues. These are just two examples of vulnerable steps in production where the consumer ranges from companies to hobbyists. As quickly as vulnerabilities are found and exploited, researchers work to bolster security protocols. Building a foundational understanding of these systems and methodologies enables the next generation of researchers and industry professionals to buffer against attack. A few preventative measures involve the analysis of IC components and IC markings. Bent or scratched pins may suggest parts are being recycled, are inauthentic, have been tampered with, or all of the above. Conflicting, nonsense, or layered IC markings may also signal to a consumer that a part is inauthentic. We will deploy image processing and optical character recognition (OCR) techniques on a small dataset of IC chip images to do our own analysis. We will seek to identify bent pins and text.

The remainder of this paper is organized as follows: Section II will explain project methodology broken down into pin detection and marking subsections. Section III will detail results produced by our program. Future work will be discussed in Section IV, and finally, we will conclude in Section V.

II. Methodology

The explanations of our methodology will be divided between IC pin detection and IC marking detection.

A. Pin Detection

1) Image Preprocessing: A vital step of both the pin detection and marking detection implementations happened during image preprocessing. First, images are loaded into code, and shadows are removed through a series of steps. Beginning with Contrast-Limited Adaptive Histogram Equalization (CLAHE), image contrast is enhanced. Next, a bilateral filter is applied to smooth image appearance and reduce noise. Then, adaptive thresholding is applied to remove shadows.

2) Image division: The next step in preprocessing occurs in the division of the image into 2 significant parts: the isolated center of the IC (where markings will be detected) and the image with the excluded center (which will aid in pin detection). This happens in a secondary series of steps: application of bilateral filter for smoothing, application of Otsu's Thresholding to make edges clear, and finally, the application of contour detection to find the largest contour (which represents the center of the IC).

3) Boxing: After the center of the image has been removed, an edge detection operator entitled *Canny* from OpenCV can be applied [1]. This allows contours to be found on the image which now consists exclusively of pure edges. Directional box merging is implemented to "box" pins on the IC's edges. Pins can then be labeled as "Up," "Down," "Left," or "Right" depending on their location on the IC. Up/Down and Left/Right pins can be immediately distinguished due to the vertical or horizontal natures of their respective boxes. Divides between the groups can then be made by labeling them relative to the image center. Finally, any

	A-D-64QFP-14B		A-J-28SOP-01B		A-D-64QFP-15B		C-T-28SOP-04F	
Ambiguous Classification	Ambig Right	Ambig Wrong	Ambig Right	Ambig Wrong	Ambig Right	Ambig Wrong	Ambig Right	Ambig Wrong
True Positive	6	3	4	3	2	1	3	1
False Positive	3	6	0	1	1	2	0	2
True Negative	55	55	60	60	25	25	25	25
False Negative	0	0	0	0	0	0	0	0
Accuracy	61/64 = 95.3%	58/64= 90.6%	64/64= 100%	63/64= 98.4%	27/28= 96.4%	26/28= 92.9%	28/28= 100%	26/28= 92.9%

Fig. 1. Pin Detection Results.

spuriously-labeled pins can be reassigned to their correct mapping by utilizing the centroid of a group of boxes. At this point, boxes can be re-merged with a more aggressive directional bias [2]. For each group, outliers can be detected with simple mean and standard deviation calculations and are labeled as suspect/ defective pins [3].

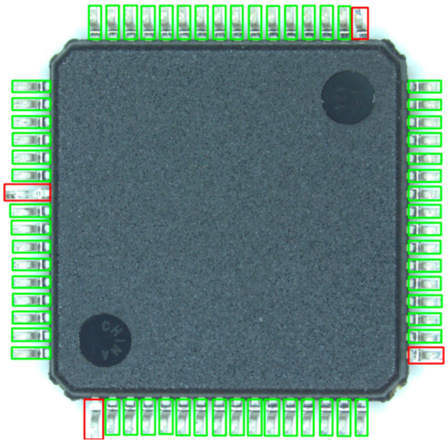


Fig. 2. Example Pin Detection.

A successful result is displayed above in Figure 2.

B. *Marking Detection*

1) Image Preprocessing: A second round of preprocessing has to be performed preceding marking detection. Before images can be passed into the OCR (optical character recognition) model, they have to be cleaned. Otherwise image grain, detected edges from non-character elements, or IC features may be parsed as text.

The image center which is disregarded in pin detection is utilized in analysis here. Isolating the center already shows drastic improvement in OCR results as pins and extra space can no longer be mistakenly parsed as text. To further prep the image, noise is smoothed and thresholding is applied to the image. Next, a series of opening and closing contours is performed. “Opening” refers to the morphological operation which executes erosion followed by dilation to eliminate small noise that is not part of markings. Then remaining large bits of noise can be identified from the contours list by size and are removed. Finally “closing,” or dilation followed by erosion, is performed to fill gaps in the remaining markings. Notable steps in this process can be seen in Figure 3. After concluding the final closing step, the images are saved and ready to be passed to OCR.

The OCR module we selected was EasyOCR [4]. EasyOCR is comprised of three components: feature extraction with ResNet and VGG, sequence modeling with long short-term memory (LSTM) networks, and finally, decoding with connectionist temporal classification (CTC) [4]. We selected OCR due to its portability, overall satisfactory performance, and because it was easily integrated into our code.

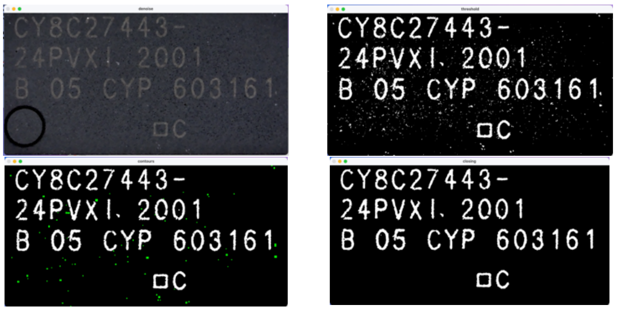


Fig. 3. 4 major steps of image preprocessing preceding marking detection. Isolate center and smooth noise (top left). Apply thresholding (top right). Morphological opening; identify remaining noise (bottom left, shown in green) and fill with black. Morpholgical closing; prepared for OCR (bottom right).

III. Results

Leaving some room for improvement, our program produces satisfactory results. In particular, our program demonstrates sound pin detection, boasting a 0% false-negative rate. This statistic is highlighted in Figure 1. After image preprocessing, OCR produced an 89% accuracy in detection of IC markings. Analysis of image “C-T-48QFP-19F-SM.png” yielded poor results continuously despite preprocessing. The image suffered

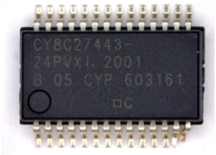
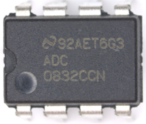
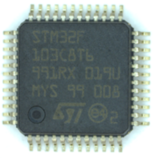
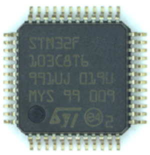
IC Image:				
Desired Text:	CY8C27443- 24PVXI 2001 B 05 CYP 603161 C	92AET6G3 ADC 0832CCN	STM32F 103C8T6 991RX 019U MYS 99 008 2	STM32F 103C8T6 991UJ 019U MYS 99 009 2
OCR-Produced Output:	CY8C27443 - 24PVXI 2001 8 05 CYP 603161 OC	N92AET6G3 ADC 0832CCN O	S7732F JOJLBTB A91RX 019U M_S 93 008 762	STM3ZF 103C8T6 991UJ 019U MYS 99 009 6O2
Correct Characters / Total:	32 / 33 = .939	18 / 18 = 1.00	20 / 31 = .645	30 / 31 = .968

Fig. 4. Marking Detection Results. Reusults in an average accuracy of 89%.

from excessive grain and poor image contrast. However, significantly better results from a similar image

leave us to wonder whether the problem lies in the representation of characters on the IC itself. This is what prompted us to consider the benefits of training a model on the unique typefaces utilized in IC marking. When tailoring the extent of preprocessing to each individual image, accuracy was only improved. As we are currently unable to automate this process, we opted to leave our program with the uniform series of steps which produced the most optimal results across the board.

IV. Future Work

Future work may consist of extending analysis beyond the small dataset we worked with for this project. Further analysis using more advanced pre-trained OCR models is another aspect deserving of consideration. Contingent on performance after implementing alternate OCR models, our needs may necessitate training our own model to recognize and parse the nonstandard fonts used for IC markings. As one of the images in our dataset had fairly severe amounts of noise and grain, we would likely be able to circumnavigate performance interference caused by this issue by training on a larger variety of images and image qualities.

V. Conclusion

We successfully implemented a rudimentary system for IC counterfeit detection. As we originally aimed for 80-90% accuracy, our program exceeded our goal.

Outsourcing and diversifying means of manufacture has rapidly become a norm in the production of tech; however, we have now observed firsthand the tedious process required to identify suspect hardware. Even small quantities of malicious hardware reaching the market can cause a domino-effect of malicious activity. Reaching project completion has equipped us with more in-depth knowledge about hardware security and the steps that must be taken to justify the benefits of participating in the market as it is. Learning about the methodologies of this process alongside the risks of placing unwarranted trust in other parties instills an understanding and appreciation for security and defense mechanisms in us. We are hopeful for future advancements in this area of research.

Acknowledgments

The authors would like to thank Dr. Hoque for his support throughout the duration of this project and for providing foundational knowledge for our hardware security project.

References

[1] Bradski, G, "The OpenCV Library," https://github.com/opencv/opencv/wiki/CiteOpenCV, (Accessed December 10, 2024).

[2] Chu, I, "RE: How to Merge Nearby Bounding Boxes,"https://stackoverflow.com/questions/66490374/how-to-merge-nearby-bounding-boxes-opencv , (Accessed December 10, 2024).

[3] Harris, C.R., Millman, K.J., van der Walt, S.J. et al. Array programming with NumPy. Nature 585, 357–362 (2020). https://doi.org/10.1038/s41586-020-2649-2.

[4] Mahajan, A, "EasyOCR: A Comprehensive Guide,"https://medium.com/@adityamahajan.work/easyocr-a-comprehensive-guide-5ff1cb850168, (Accessed December 10, 2024).