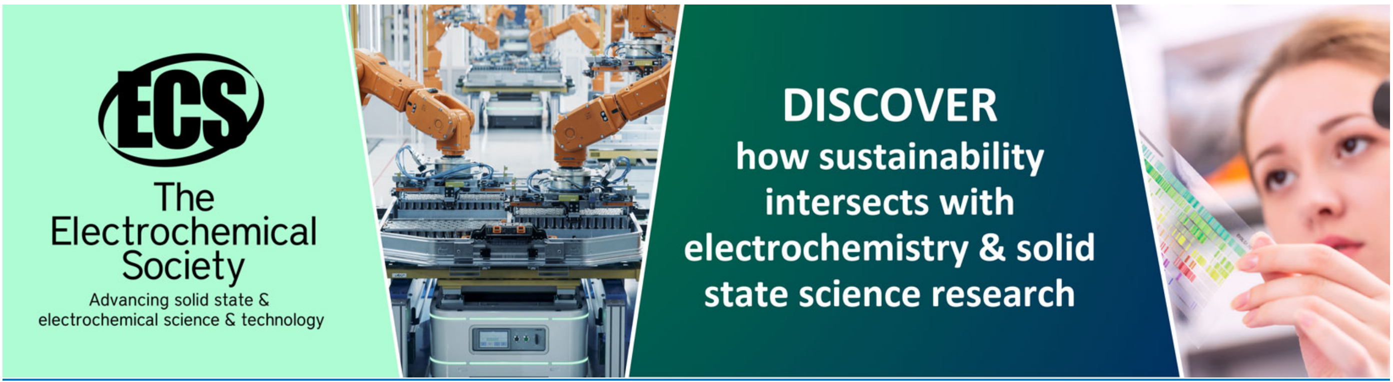
Journal of Physics: Conference Series

PAPER • OPEN ACCESS

Counterfeit Electronics Detection Using Image Processing and Machine Learning

To cite this article: Navid Asadizanjani *et al* 2017 *J. Phys.: Conf. Ser.* **787** 012023

View the article online for updates and enhancements.


ECS
The Electrochemical Society
Advancing solid state & electrochemical science & technology
DISCOVER
how sustainability intersects with electrochemistry & solid state science research

Counterfeit Electronics Detection Using Image Processing and Machine Learning

Navid Asadizanjani, Mark Tehranipoor and Domenic Forte

Electrical and Computer Engineering Department, University of Florida
Gainesville, USA.

E-mail: nasadi@ece.ufl.edu

Abstract. Counterfeiting is an increasing concern for businesses and governments as greater numbers of counterfeit integrated circuits (IC) infiltrate the global market. There is an ongoing effort in experimental and national labs inside the United States to detect and prevent such counterfeits in the most efficient time period. However, there is still a missing piece to automatically detect and properly keep record of detected counterfeit ICs. Here, we introduce a web application database that allows users to share previous examples of counterfeits through an online database and to obtain statistics regarding the prevalence of known defects. We also investigate automated techniques based on image processing and machine learning to detect different physical defects and to determine whether or not an IC is counterfeit.

1. Introduction

Counterfeit ICs are a growing issue in the global market, with some counterfeit ICs infiltrating high-risk applications such as those in the military or medical sectors [1], [2]. Counterfeit ICs may feature a reduced lifespan, fail under conditions mandated by the manufacturer's specification, leak secure data from the system to a remote location (i.e., hardware Trojans [3]), and more. In some cases, the adversary copies the design and labels counterfeit ICs with his own brand name or with the original equipment manufacturer's name using reverse-engineering techniques [4], [5], [6]. Since the adversary does not incur the engineering research and development costs, his products are priced lower, thereby resulting in a loss for authentic products.

The taxonomy of counterfeit component types and defects in electronic components have been discussed in [7]–[8], where the two main classes of defects are classified as physical and electrical defects. Here we will study the physical defects and try to expedite and facilitate the detection and recordkeeping of such physical defects. With the help of high-precision microscopes, most of the physical defects can be identified by a subject matter expert (SME). The SME has to carefully test the entire IC for each defect. This is a labor-intensive and timely process. For a modern electronics manufacturer, it is not feasible for an SME to analyze all the chips and components manually because a single US company can import tens of millions of chips annually. Some examples of these defects focused in this proposal are presented in Fig. 1 [9].

The US Chamber of Commerce has identified 11 issues in the current state of the global electronic supply chain in regard to counterfeiting [10]. In this paper we begin to address the need for stricter

Published under licence by IOP Publishing Ltd

testing protocols and quality control practices. We will develop preliminary algorithms that analyze optical images and automatically extract defects using artificial neural networks and image processing algorithms.

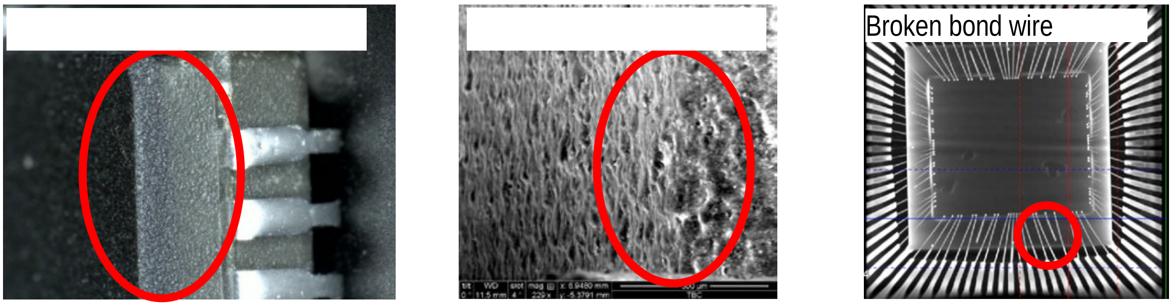


Figure 1. Physical defects seen on electronic components.

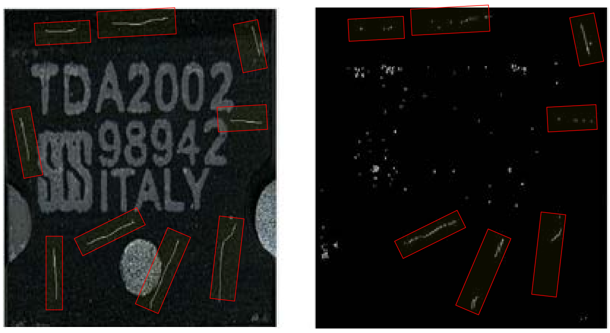
2. Automated Defect Detection

We investigate two techniques for detecting defects on ICs are presented: 1) image processing and 2) artificial neural network (ANN)–based algorithms.

2.1. Defect Detection Using Image Processing

There are many image processing algorithms and filters used to detect features on different images. These algorithms are designed in a way to highlight one specific property of a feature based on its histogram. These features can be defects on ICs such as scratches, color variations, ghost marking, etc. It is very important to select the right algorithm and right threshold in order to better highlight the defect. Figure 2a shows top view of an IC with scratches. We have applied image processing techniques on this image including Hough transform, an edge-enhanced maximally stable filter, and Sobel filter to detect these scratches [11]-[13] as shown in Fig. 2b.

Although filters are widely used for detection, the results are not accurate enough to make a decision on the authenticity of a chip. Many defects cannot be identified through this method alone. For example, many counterfeits display variations in color that can be identified based on the intensity of grayscale pixels rather than the placement of binary pixels within the image. To determine whether color variations exist on a given IC, an entirely different image processing method is needed in addition to the scratch detection algorithm.



(a) (b)

Figure 2. The counterfeit IC with scratch defect a) original image b) after image processing.

Such cases increase the complexity of developing a human independent platform to determine the authenticity of an IC based on filters. Ideally, one single algorithm should be used to identify a wide variety of defects under various imaging conditions. This will not be possible without getting help from machine learning. Artificial neural networks are one of the most popular machine learning techniques which we have identified as a promising candidate to achieve this goal. Considering the fact that such an algorithm can train itself and become smarter as more samples are tested, this is a huge advantage to other algorithms and filters.

2.2. Defect Detection Using ANN

ANN is a machine-learning technique that is modeled similarly to the structure of the human brain. A typical ANN consists of neurons in different layers, such as input, hidden, and output layers. The information is stored in the interconnections between neurons across the adjacent layers [14]. The number of hidden layers should be at least one, and the number can be increased depending on the application. The output of a neuron is calculated by using an appropriate activation function for the input value. Step function, *tanh*, and *arctan* are some of the examples of activation functions that will be used in this proposal.

In the structure of a typical ANN each neuron from the input layer is connected to each neuron in the first hidden layer. The input value at a node of a hidden layer is the weighted summation of all the input nodes. The factor by which each node contributes to the value of the input of the next node will decide the weight matrix between the two layers. Thus, the weight matrix stores crucial information which is unique for each class of inputs. The output of the neuron is computed using the activation function. This weighted summation of adjacent layers continues until the output layer. During the training phase, the expected output is compared with the output obtained by the computations. The error is then sent back to the network layers to update the weights.

The backpropagation algorithm (i.e., propagating the error in the backward direction of the network) is a popular artificial neural network algorithm used for training the network. The basic idea behind backpropagation algorithm is to reduce the error energy of a neuron in the neural network. In general, backpropagation method requires a known or desired output for each input value, therefore calculating the error will help to quantify this step. Differentiating the error energy functions with respect to the weights gives the condition for minimum error energy. These error values will be used to adjust the weights associated with input values for each node. This learning phase will be iterated until the error meets an accepted value. The neural network is then trained for more iterations for the same image until the error level is less than the accepted tolerance level.

The complexity of the operations in the ANN depends on factors such as the number of nodes in each layer, the learning rate, the number of iterations done for training, and the number of hidden layers. The artificial neural network is an extremely useful tool for pattern recognition applications. The ANN has been studied extensively for applications such as character recognition, face recognition, etc. We have used backpropagation algorithm to Train ANN. In the next section, we will discuss our implementation of ANN using backpropagation on a chip with scratches.

2.2.1. ANN Implementation. The implementation of the counterfeit IC detection using artificial neural network can be divided into two parts: 1) **training phase**; and 2) **testing phase**. In our experiments, we divide the available set of images into 2 samples for training and 4 for testing. In the future, this proportion can be varied to check the efficiency of the system. In addition, while we focus on scratches only, we can train for multiple defects at once.

Training Phase: We utilized backpropagation algorithm discussed earlier to train the artificial neural network, using tanh, the hyperbolic tangent function, as the activation function. Only one hidden layer was used to reduce the complexity of the ANN, but two and three layers will be investigated in this work to optimize the results.

Figure 3 shows the flowchart for the counterfeit IC detection procedure. The input image is read using the MATLAB function imread, which gives a matrix consisting all pixels of the image. The image is then converted to a column matrix by arranging the columns of the input image matrix one below the other. If the input image is of size M × N, the new column matrix will have a size of MN × 1. This can be reduced by resizing the image using various preprocessing techniques such as binning. However, resizing will decrease the quality of the image and resolution of the features to be detected.

Other preprocessing techniques such as filters to remove pixels over or below a threshold, edge detection, normalizing the image, and converting RGB images to gray scale can be used for preprocessing images and make images ready for ANN analysis. The input image is normalized by dividing pixel values by 255 for grayscale images. The techniques applied for the processing of the image are the same for training and testing till this step. However, during training the output for a particular input image is known.

The weight matrices between input, hidden, and output layers are initialized with random values between −0.5 and +0.5. After preprocessing the images, expected output layer is initialized with its corresponding class. The linear combination of the initial column matrix and the first weight matrix gives the input for the hidden layer. Using the activation function tanh, we get the output of hidden layer. Since the linear combination of input with weight matrix can be very expensive computationally, activation function helps keeping them in the range of −1 to 1. This output will then be multiplied by weight matrix between the hidden and output layer to calculate the input.

Similar to nodes in the hidden layer, the output of output layer is calculated using the activation function. The error at the output is calculated by comparing activation function with the expected output given at the start of training. The error is propagated back to the network. This loop can be continued until we reach a point where the error is within an expected range.

Testing Phase: During testing, we do not use backpropagation algorithm, as the ANN is already trained with error corrections for the weights. In fact, we do not know the output value prior to the computations in ANN.

ANN will then determine if the IC is defected or not depending on its output. We apply the same preprocessing methods that were employed during the training phase for the images. In order to test this technique, we compare the output of the ANN during testing with the expected output and calculate the efficiency of the system. The flow of training and testing of the ANN is explained in Figures 3 and 4.

The output classes considered for this experiment were 0 and 1. The output value 0 represents no defect condition, and the output value 1 indicates that the IC under consideration is defected.

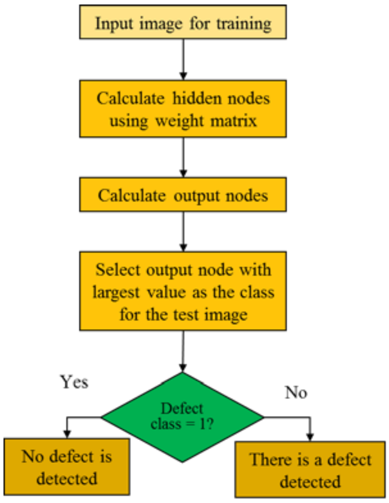

Input image for training
Calculate hidden nodes using weight matrix
Calculate output nodes
Select output node with largest value as the class for the test image
Yes
Defect class = 1?
No
No defect is detected
There is a defect detected

Figure 3. Logical flowchart for artificial neural network testing.

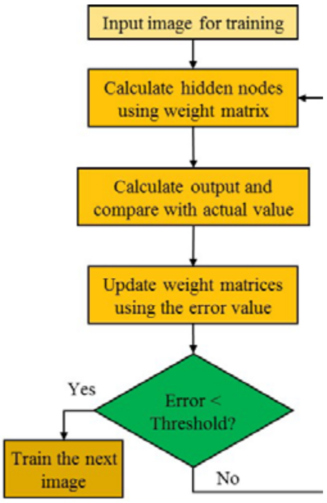

Input image for training
Calculate hidden nodes using weight matrix
Calculate output and compare with actual value
Update weight matrices using the error value
Yes
Error < Threshold?
Train the next image
No

Figure 4. Logical flowchart for artificial neural network training.

2.2.2. Preliminary Detection Results Using ANN. The backpropagation ANN was implemented using MATLAB platform to detect scratches on chips. Standard images were taken from 6 different WSI-PSD813F1 chips with and without scratches. The training was done using two images with and without defects for 10 iterations each. Figure 5 shows the images used for training and ANN was tested for detecting counterfeit ICs. The original image had more than 4 million pixels. This means we need 4 million input nodes, where we have only two output nodes. This was resolved by reducing the size of the images to a resolution of 512×512 during preprocessing stage. The number of hidden nodes were kept as a variable to be adjusted depending on the results. One hundred hidden nodes gave us good results. As we increase the number of nodes, the computational time rises. For our experiments, we set the number of hidden nodes as 100. The number of IC images used for testing were four, out of which two were counterfeit. The weight matrices were initialized using random functions available in MATLAB between −0.5 to 0.5. The images used for testing is shown in Fig. 6. ICs (a) and (c) have scratch defects and are correctly detected by the ANN.

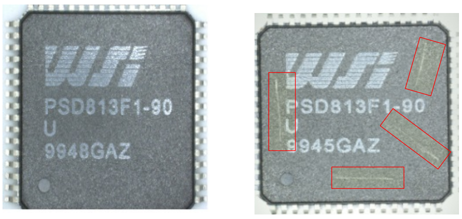

(a) (b)

Figure 5. Images used for training a) IC without defect b) IC with highlighted scratches.

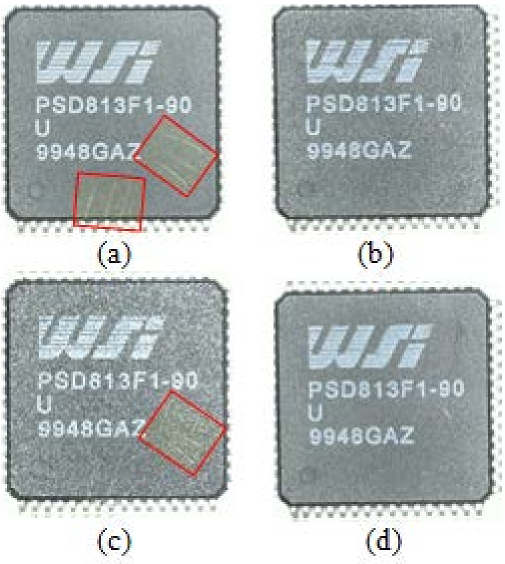

(a) (b) (c) (d)

Figure 6. IC images with scratch defect used for testing.

3. Conclusion and Future Work

In this paper, we described a counterfeit database that has been developed to improve recordkeeping of counterfeit incidents, support education and increase awareness about counterfeit ICs/defects, and provide resources for research in automated counterfeit IC and defect detection. As an exemplary instance, data was used to investigate image processing and neural network algorithms for scratch detection. Image processing can successfully detect ICs with scratches, but fails when the scratches are not highly visible or their color is not significantly different from that of the IC surface. Another current limitation is that the exact same algorithm will not apply for other defects. However, the artificial neural network should be more amenable to other types of defects in the future. In future work, we will extend the existing ANN technique to detect other physical defects and to data from additional sources (X-ray, SEM, etc.).

Acknowledgments

The authors would like to acknowledge support from National Science Foundation (grants CCF-1423282, CCF-1559772, and CNS-1558516) for this work.

References

[1] U. Guin D. DiMase and M. Tehranipoor, "Counterfeit Integrated Circuits: Detection, Avoidance, and the Challenges Ahead," Journal of Electronic Testing: Theory and Applications (JETTA), vol. 30, no. 1, pp. 9-23, 2014.

[2] U. Guin, K. Huang, D. DiMase, J. Carulli, M. Tehranipoor, and Y. Makris, "Counterfeit Integrated Circuits: A Rising Threat in the Global Semiconductor Supply Chain," Proceedings of the IEEE, vol. 102, no. 8, pp. 1207–1228, 2014.

[3] U. Guin, D. Forte and M. Tehranipoor, "Anti-counterfeit techniques: From design to resign", Proc. 14th Int. Workshop Microprocessor Test Verification (MTV), pp. 89-96.

[4] K. M. Buddhiraju and I. A. Rizvi, "Comparison of CBF, ANN and SVM Classifiers for Object-based Classification of High Resolution Satellite Images", Proc. IGARSS, pp. 40-43, 2010.

[5] X. Zhang, K. Xiao, and M. Tehranipoor, "Path-Delay Fingerprinting for Identification of Recovered ICs" in Proc. IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFTS), 2012.

[6] N. Asadizanjani, "3D Imaging and Investigation of Failure and Deformation in Thermal Barrier Coatings Using Computed X-ray Tomography." (doctoral dissertation) , 2014.

[7] S. Quadir, J. Chen, D. Forte, N. Asadizanjani, S. Shahbazmohamadi, L. Wang, J. Chandy, and M. Tehranipoor. "A survey on chip to system reverse engineering." *ACM Journal on Emerging Technologies in Computing Systems (JETC)* 13, no. 1 (2016): 6.

[8] N. Asadizanjani, S. Shahbazmohamadi, M. Tehranipoor, and D. Forte. "Non-destructive PCB Reverse Engineering Using X-ray Micro Computed Tomography." In *41st International Symposium for Testing and Failure Analysis (November 1–5, 2015)*. Asm, 2015.

[9] www.counterfeit-ic.org

[10] Measuring the Magnitude of Global Counterfeiting available online http://www.theglobalipcenter.com/wp-content/themes/gipc/map-index/assets/pdf/2016/GlobalCounterfeiting_Report.pdf

[11] R. Karri, J. Rajendran, K. Rosenfeld and M. Tehranipoor, "Trustworthy Hardware: Identifying and Classifying Hardware Trojans", Computer, vol. 43, no. 10, pp. 39-46, 2010.

[12] S. Haykin, S. Simon, "Neural Networks and Learning Machines", Vol. 3. Upper Saddle River, NJ, USA: Pearson, 2009.

[13] E. Ardizzone, H. Dindo, and G. Mazzola, "Multidirectional Scratch Detection and Restoration in Digitized Old Images," Eurasip Journal on Image and Video Processing, vol. 2010, no. June 2015, 2010.

[14] U. Guin D. Forte and M. Tehranipoor, "A Comprehensive Framework for Counterfeit Defect Coverage Analysis and Detection Assessment", Journal of Electronic Testing (JETTA), Volume 30, Issue 1, pp 25-40, February 2014.