Real Time Automated Counterfeit Integrated Circuit Detection using X-ray Microscopy

Kaleel Mahmood,[1] Pedro Latorre Carmona,[2,*] Sina Shahbazmohamadi,[1] Filiberto Pla,[2] Bahram Javidi,[1]

[1] Department of Electrical and Computer Engineering, University of Connecticut, 371 Fairfield Way, Storrs, CT, USA, 06269

[2] Department of Computer Languages and Systems, Institute of New Imaging Technologies, Jaume I University, Campus del Riu Sec s/n, Castellón de la Plana, Spain, 12071

**Corresponding author: latorre@uji.es*

Received Month X, XXXX; revised Month X, XXXX; accepted Month X, XXXX; posted Month X, XXXX (Doc. ID XXXXX); published Month X, XXXX

Determining the authenticity of integrated circuits is paramount in preventing counterfeit and malicious hardware from being used in critical military, healthcare, aerospace, consumer, and industry applications. Existing techniques to distinguish between authentic and counterfeit integrated circuits often includes destructive testing requiring subject matter experts. We present a non-destructive technique to detect counterfeit integrated circuits using X-ray microscopy and advanced imaging analysis with different pattern recognition approaches. Our proposed method is completely automated, and runs in real time. In our approach, images of an integrated circuit are obtained from an X-ray microscope. Local binary pattern features are then extracted from the X-ray image followed by dimensionality reduction through principal component analysis, and alternatively through a non-linear principal component methodology using a stacked autoencoder embedded in a deep neural network. From the reduced dimension features, we train two types of learning machines, a support vector machine with a non-linear kernel, and a deep neural network. We present experiments using authentic and counterfeit integrated circuits to demonstrate that the proposed approach achieves an accuracy of 100% in distinguishing between the counterfeit and authentic samples.

OCIS codes: (340.7440) X-ray imaging; (100.4996) Pattern recognition, neural networks; (150.1135) Machine vision, algorithms.

1.Introduction

Counterfeit integrated circuits (ICs) represent a serious threat to the functionality and reliability of electronic equipment in the military, industry, and in healthcare. The Committee of Armed Services of the United States Senate found that there were 1,800 cases of counterfeit electronics entering the U.S. supply chain from 2009 to 2010 alone [1] and the Semiconductor Industry Associates estimate that counterfeiting costs U.S. semiconductor companies 7.5 billion dollars annually [2]. Globally, it has been estimated that illegitimate production of components costs electronic companies worldwide 100 billion dollars annually [3]. Currently there are two approaches to identify counterfeit hardware, physical analysis and electrical testing [4]. We will briefly discuss each approach and the advantages our method has over the existing techniques.

Physical analysis of integrated circuits falls into two main categories, interior testing and exterior testing [5]. Interior testing often destroys the samples being analyzed, so in a scenario where there is a large batch of chips with a few counterfeits mixed in, this technique is not feasible. Exterior testing inspects ICs using metric based tests. The results of these tests depend largely on the subject matter expert [6], and therefore are prone to human error and lack automation. Electrical testing of integrated circuits has been presented as a means to overcome the deficits associated with physical analysis; however these methods also come with their own set of disadvantages [6]. Electrical parametric testing is problematic because the variance in electrical parameters between integrated circuits of the same type can be high, even if all of the samples are authentic, and changes in the electrical parameters can also be due to the age of the device rather than counterfeiting [7]. Functional and structural testing often require a specific chip infrastructure or extraneous information such as a circuit netlist or integrated circuit scan chain that is not always available. Finally Burn-in testing which measures circuits under stressful conditions suffers from the same sample destructive problems as certain physical tests mentioned above.

X-ray imaging is a powerful technique to inspect the features and details of objects which are unable to be viewed using the visible spectrum of light [8-13]. Thus, X-ray imaging is a useful tool for the inspection of ICs by examining the details concealed by the packaging surface. We propose a real time automated IC authentication technique using X-ray microscopy integrated with image processing and classification algorithms to inspect the ICs. Images captured with an X-ray microscope are first registered using a series of correspondence points obtained through normalized cross correlation filtering and the affine transform. After registration, we apply local binary pattern (LBP) feature extraction to characterize the IC images. The dimensionality of the resulting features is reduced using linear principal component analysis (PCA), as well as non-linear PCA through a stacked autoencoder. Two classification approaches are implemented to distinguish between the authentic and counterfeit ICs. One approach is based on a support vector machine (SVM) with a radial basis function kernel, and the other approach is based on a deep neural network. Our method has advantages over physical testing because it is non-destructive and completely automated, thus it is independent of subject matter experts. In comparison to electrical testing, our technique does not rely on parameters that have the potential to show great variance and does not require additional information about the chip, such as a circuit netlist or circuit scan chain. Finally, our entire process, from the image capture step to the classification of the IC sample runs in 2 seconds, making it a viable option for military or industrial implementation.

The rest of the paper is organized as follows: In section 2, we elaborate on the steps involved in our approach including the X-ray microscopy setup and the feature extraction and pre-classification algorithms. Section 3 discusses our two image classification approaches and the details related to their training. In section 4, we discuss our experimental results. Conclusions and related future work are given in section 5.

2. Counterfeit IC Detection

Figure 1 shows a diagram of the counterfeit IC detection method we implement. In our approach an X-ray image of the die of each circuit is taken using an X-ray microscope. Once the X-ray image of the die is captured, one image from the captured set is selected as the base image and the other images are registered with respect to this base image. Once all the images have been registered, each registered image is divided into blocks and a histogram of the LBP values is generated for each block. The histogram of each block is then concatenated together to form a vector of the histograms.

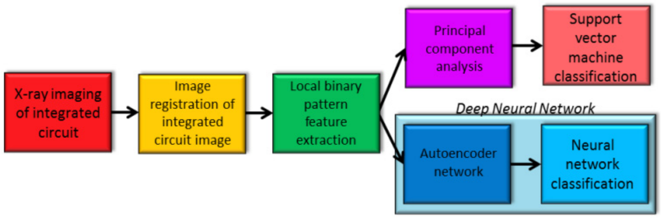
X-ray imaging of integrated circuit
Image registration of integrated circuit image
Local binary pattern feature extraction
Principal component analysis
Support vector machine classification
Deep Neural Network
Autoencoder network
Neural network classification

Fig. 1. System diagram of the counterfeit IC detection method.

We use two types of principal component analysis (PCA) on the vector representations of the histograms to reduce the dimensionality of the feature space. In our SVM based approach we use linear PCA to reduce the vector size. In our deep neural network approach a stacked autoencoder is encapsulated within the first two hidden layers of the network to perform non-linear PCA. Using the features as inputs, each classifier is then trained to distinguish between counterfeit and authentic ICs. In the following sub sections we will go over the X-ray capture process and the details pertaining to the pre-classification algorithms.

A. X-ray Microscopy System

X-ray microscopy has been used for a wide variety of applications including defect inspection [14], biomedical imaging [15] and image security [16], to name a few. In [15] noise analysis of X-ray tomography is done for improved detection of nanoparticles in the brain, and in [16] an X-ray backscatter imager is used for explosive device detection. Non-destructive defect detection using X-rays has also been investigated in previous literature such as [14,17]. In X-ray lamininography the sample and detector are synchronized to rotate while 180° out of phase with one another during imaging, and has been used to detect defects in printed circuit boards [14]. In X-ray grating interferometry a phase grating, in addition to an absorption grating is used to provide an absorption image, differential phase image, and visibility contrast image of the sample. This technique has been used to find micro-voids and scars in the encapsulant of ICs [17]. While the X-ray techniques mentioned above have potential, they have not been automated and have not been used for real time IC imaging like our proposed approach.

A Zeiss X-radia X-ray microscope was used for the X-ray imaging in our experiments, as shown in Figure 2. The voltage and power used for each X-ray image was chosen to be 90 KV and 8W, respectively. These parameters were chosen to allow sufficient transmission values (22-35%).The exposure time for each image was selected to be 1 second. This exposure time, in conjunction with the voltage and power values resulted in a sufficient number of X-ray counts (>5000) and provided a good signal to noise ratio. The source and detector were positioned sufficiently far away (source distance 100mm and detector 75mm) from the samples. All imaging parameters were selected to be in accordance with the parts manufacturer's data sheet to ensure that the parts were not overexposed to X-rays so the dosage would remain within allowable limits. The large field of view detector was chosen to be able to capture the entire die in a single image. In order to simulate real situations and further assess the robustness of our method, samples were mounted with different orientations and/or were tilted by the sample holder.

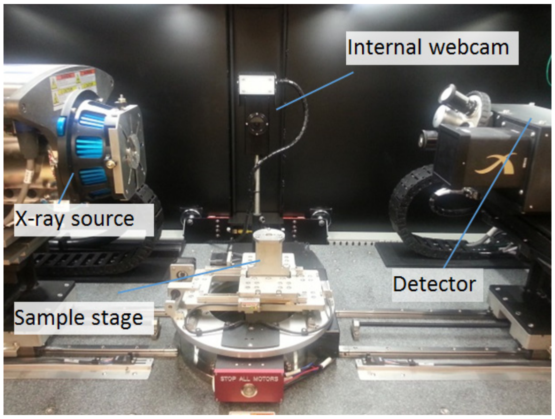

Fig. 2. X-ray microscope interior view.

B. Image Registration and Feature Extraction

We explored a number of mathematical approaches for registration and feature extraction of the IC images obtained by microscopy. Mutual information based registration [18] was initially considered to register the IC X-ray microscope images. Mutual information is a measure of the dependence among variables. When two gray scale images are correctly aligned, the dependence between their grey scale values is maximized, and consequently the mutual information value between the two images is also maximized. While this registration algorithm was successful in registering X-ray microscope images captured under the same operating parameters, this method did not perform in a satisfactory manner when attempting to register images to a base image taken under different X-ray operating parameters. In addition this registration algorithm was slow to converge for larger images.

The registration algorithm used in our approach is based on the affine transform without shear. The parameters of the affine transform are computed by minimizing the difference between a series of correspondence points obtained using filters. For our registration, we use the four corners of the dies of the chips as correspondence points between the two images. To avoid having to manually set the points, we use the cropped corners of the base image, and a normalized cross correlation filter [19] to determine the x and y pixel locations of the corners on each image being registered. Once we determine the x and y location of the four correspondence points on the image to be registered, we use linear least squares minimization [20] to determine the affine transform parameters, and apply the computed transform parameters to the image to be registered.

After registering the X-ray images we compute the local binary pattern (LBP) of every pixel in each image. LBPs are a non-parametric feature extractor originally used for face recognition [21]. LBP creates an 8 bit binary number for each pixel in an image that represents the pixel's relationship to the intensity of its closest neighbors. In our implementation of the LBP conversion method each pixel's grayscale intensity is compared to the grayscale intensity of its 8 closest neighbors, as shown in Figure 3. After each pixel's binary neighbor representation is generated the pattern type of the binary number is determined.

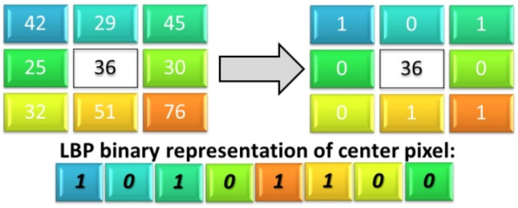

Fig. 3. Illustration of local binary pattern pixel calculation.

There are two significant binary pattern types, uniform and non-uniform patterns. A binary number with a uniform pattern has two or less bit changes and a binary number that has more than two bit changes is considered non-uniform. It has been shown in [21] that for the majority of images, 90% of LBP 8 bit binary values are uniform patterns. For our set of IC images, on average 85.49% of the pixels in each image could be directly represented by uniform patterns. The range of an 8 bit number is 0 to 255, and there are a total number of 58 uniform patterns in this range. As a result the LBP of each pixel can be represented by 1 of 58 values instead of the 256 different values that can be represented by an 8 bit number.

As stated in [21], despite the majority of patterns being uniform, non-uniform LBPs still exist in images. Any binary number with more than 2 bit changes is converted into a uniform pattern. In our appraoch, the non-uniform LBP is converted to a uniform LBP by measuring the Hamming distance between the current non-uniform 8 bit binary number and all 58 uniform patterns. The uniform pattern that has the shortest Hamming distance to the non-uniform pattern is used, and the non-uniform pattern is thereby replaced with the closest uniform approximation. If two or more patterns are tied for closest Hamming distance then the decimal Euclidian distance is computed for each of the closest uniform candidates and the candidate with the shortest Euclidean distance is selected.

3. Counterfeit IC Classification

We use two different strategies to classify our data. Our first approach is based on a support vector machine (SVM) with a radial basis function kernel [22,23]. The other classification approach is based on a deep neural network [24]. SVM works by creating a hyperplane in a high dimensional space that can act as a discriminator between two classes [25]. SVM achieves this separation by minimizing the following function:

$$f(w,\xi) = \frac{1}{2}w^T w + \mathrm{C}\sum_{i=1}^{N}\xi_i \tag{1}$$

given the following constraints:

$$y_i(w^T\phi(x_i)+b) \geq 1-\xi_i \qquad \text{for } i = 1,\dots,N \tag{2}$$

$$\xi_i \geq 0 \qquad \text{for } i = 1,\dots,N \tag{3}$$

where in Eq. (1), N is the total number of training samples, w represents a weight vector, C is the reciprocal of the regularization parameter used to avoid over fitting when the classification problem contains non-separable or noisy data points, and ξ_i is a slack variable for the i^{th} training sample. In Eq. (2), y_i represents the class label (1 or -1), b represents a bias term, x_i represents the data vector corresponding to the i^{th} training sample and $\phi(.)$ is a kernel function that maps the input vector to a higher dimensional space. We used the radial basis function kernel:

$$\phi(x, x_i) = \exp\left(-\frac{1}{2\sigma^2} * \|x - x_i\|^2\right) \qquad (4)$$

where σ is the width parameter and x_i is the i^{th} training sample. The SVM is set up with two classification parameters, as shown in Eqs. (1) and (4). The particular values of the SVM parameters, C and σ, were obtained through an exhaustive grid search. From the entire set of authentic and counterfeit ICs we randomly generated 4 different training and test partitions, each containing the same number of authentic and counterfeit samples. The training part of each partition was used to train the SVM with the specific C and σ values. The test part of each partition was used to determine the accuracy of the SVM. After the accuracy of SVM for each of the four test sets was computed, the average accuracy of the four sets was calculated and used as the qualifying metric to determine the best σ and C parameters.

The deep neural network consists of one input layer followed by a stacked autoencoder and a multilayer neural network classifier. The overall architecture results in four hidden layers and one output layer, as shown in Figure 4. A stacked autoencoder network makes up the input layer and first two hidden layers of the deep neural network. The third and fourth hidden layers, and the output layer of the neural network act as a classifier. The classifier part of the network and the autoencoder layers are trained differently to avoid the problems associated with training all the layers of a deep network using only backpropagation [26]. The stacked autoencoder was trained using a greedy layer-wise approach [27] as shown in Figure 5. In this training method, each layer is trained separately. For each layer L, the input to this layer from the previous layer L-1 is set as the target output. Layer L is then trained using stochastic gradient descent backpropagation for a given number of iterations, or until an error threshold is reached. After the weights of layer L have been trained, the process is repeated for the next layer L+1. The pseudocode for this training algorithm is given in Figure 6.

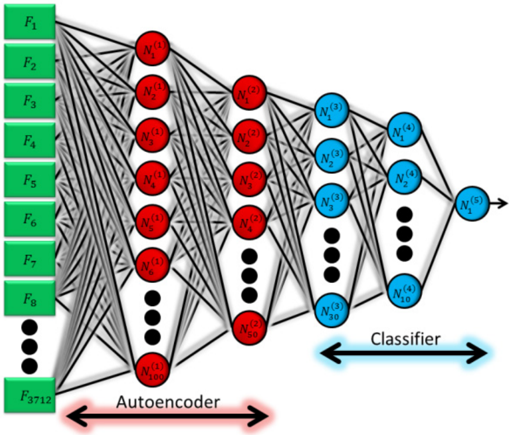

Fig. 4. Deep neural network used in our classification approach. The squares labeled F represent the input to the network. The neurons labeled N_i^j in red represent the i^{th} neuron in the j^{th} layer of the stacked autoencoder. The neurons labeled N_i^j in blue represent the i^{th} neuron in the j^{th} layer in the classification part of the network.

After the autoencoder part of the network was set up using greedy layer-wise training, the entire network was trained using stochastic gradient descent backpropagation as part of the fine tuning step to adjust the weights of the entire deep neural network. For our classification part of the network we used two hidden layers, which were obtained through empirical testing. It is possible to approximate any continuous function with a single hidden layer comprised of a finite number of neurons according to the universal approximation theorem [28]. However in our implementation, we chose two layers to reduce the overall complexity that a single hidden layer with a large number of neurons would create, and to improve the classification run time.

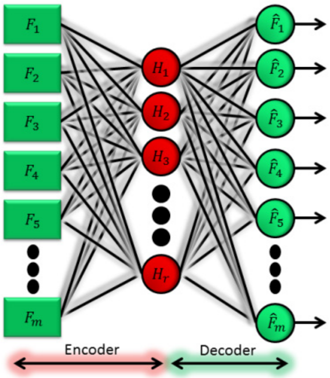

Fig. 5. Encoder-Decoder network setup. The neurons $H_1, \dots H_r$ represent the current layer L being trained. $F_1, \dots, F_m$ represents the inputs from the previous layer $L-1$ and $\hat{F}_1, \dots, \hat{F}_m$ represent the output neurons of the network which approximate a reconstruction of the input that is discarded after training.

4. Experimental Results

We used a set of 32 Intel Flash TB28F400 series ICs, of which 14 were authentic and 18 were counterfeit, to test our method. We divided our data set into two parts, a training set and a testing set, each containing 7 authentic and 9 counterfeit chips. For the SVM we used a radial basis function kernel with σ=5.455 and C=6.9519. These values were obtained using the exhaustive search detailed in Section 3. After training, the SVM was able to correctly identify 87.5% of the chips in the test set. The confusion matrix for the SVM is given in Table 1.

Table 1. Support vector machine confusion matrix

	Predicted Positive	*Predicted Negative*
Positive Class	True Positive(TP)=5	False Negative(FN)=2
Negative Class	False Positive(FP)=0	True Negative(TN)=9

The area under the curve (AUC) is defined as [29]:

$$AUC = \frac{TP}{2(TP+FN)} + \frac{TN}{2(TN+FP)} \quad (5)$$

where TP stands for true positive, TN stands for true negative, FN stands for false negative, and FP stands for false positive. For the SVM, AUC=0.857.

```
function GreedyLayerTraining $(x^1, \ldots, x^S)$

$x_j^i$ = The output vector from the $j^{th}$ layer of the network for the $i^{th}$ sample
$S$ = The total number of samples
$f_j(x)$ = The function representation of the $j^{th}$ layer of the network
$Q$ = The total number of autoencoder layers in the network
$N_p$ = The number of neurons in the $p^{th}$ layer

For L = 1: Q
        For $i = 1{:}S$
                For $j = 1{:}$ L
                        $x_{j-1}^i = f_{j-1}(x_{j-2}^i)$
                End
    1.  Create a three layer autoencoder network $K$, with an input
        layer with $N_{L-1}$ inputs, one hidden layer with $N_L$ neurons and
        one output layer with $N_{L-1}$ neurons.
    2.  Train autoencoder network $K$ with backpropgation using $x_{j-1}$
        as the input and $x_{j-1}$ as the output for all S.
    3.  After completion of training take the weights from the first
        hidden layer of $K$ and apply them to the deep neural network
        layer $L$.
        End
End
```

Fig. 6. Pseudocode for greedy layer-wise autoencoder training

The deep neural network consisted of one input layer of size 3712 followed by four hidden layers of size 100, 50, 30, and 10, with 1 neuron in the output layer. A stacked autoencoder was used as the first two hidden layers of the neural network and the third and fourth hidden layers of the network were used for classification. The deep neural network was able to correctly classify all of the chips in the test set, giving it an accuracy of 100%. The confusion matrix for the deep neural network is given in Table 2. For the deep neural network, AUC=1.

Table 2. Deep neural network confusion matrix

	Predicted Positive	*Predicted Negative*
Positive Class	True Positive(TP)=7	False Negative(FN)=0
Negative Class	False Positive(FP)=0	True Negative(TN)=9

Comparing the SVM to the deep neural network, we can see there is a discrepancy in performance between the two classifiers. This difference can be attributed to two main factors, the distribution of the data set created by linear PCA and the stacked autoencoder, and the difference in classifier architecture. It has been shown in [30] that autoencoders are more effective in reducing the dimensionality of a data set than standard linear PCA. In our data set, this same trend is also apparent. For the Intel Flash TB28F400 series IC used in our data set, there are two different ways the ICs can be counterfeit, as shown in Figure 7. The die replacement counterfeit ICs are distinctly different, while the bond wire configuration counterfeit ICs bear an extremely close resemblance to the authentic chips. This creates a problem in classification when using linear PCA because the bond wire counterfeit sample points are not easily distinguishable from the authentic sample points. This can be visually seen by graphing the first three principal components of each IC sample, as shown in Figure 8. In the blue circled region in Figure 8, the bond wire counterfeit and authentic sample points lie very close together. However, non-linear PCA (through an autoencoder) can generate a better separation of data points, as shown in Figure 9. Comparing the two principal component graphs, it can be seen that the circled region of difficult classification that is present in the linear PCA in Figure 8 does not occur in the non-linear PCA in Figure 9.

A deep neural network may also outperform an SVM due to their difference in classifier architecture. SVMs are considered a type 2 shallow architecture [24], and it has been shown in [26, 31] that a classifier with a deep architecture, such as a deep neural network, can outperform a shallow architecture. Examining the parameters used to create the SVM, we can also see certain architectural problems relating to our data set. In a normal classification approach, it is undesirable to closely fit a function to the data. SVM avoids this problem by using a slack variable, which is controlled by the parameter C. Use of this parameter allows some data points to fall inside the region of separation or on the wrong side of the hyperplane, which in our particular case results in more misclassifications. Alternatively, neural networks have no parameter to control over-fitting. Because the data points in the circled region in Figure 8 are so close together, it is necessary to fit the function precisely to the data points to achieve high accuracy.

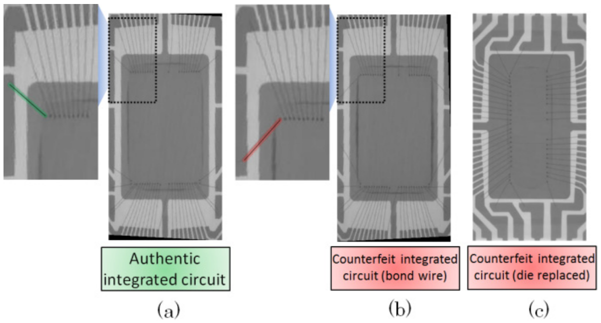
Authentic integrated circuit
Counterfeit integrated circuit (bond wire)
Counterfeit integrated circuit (die replaced)
(a)
(b)
(c)

Fig. 7. (a) Authentic integrated circuit, (b) counterfeit integrated circuit with incorrect bond wire configuration, and (c) counterfeit integrated circuit with replaced die. In the authentic integrated circuit and the bond wire counterfeit integrated circuit the bond wire configurations are enlarged and highlighted to denote the differences between the two.

Our image processing integrated with our classification algorithms in addition to the X-ray capture process is able to run in real time. Running the trained deep neural network or SVM on a sample using a computer with an Intel i7 central processing unit clocked at 2.2 gigahertz resulted in an average run time of 0.97 seconds. Combining the X-ray image capture process and the algorithm classification run time, our complete procedure runs in approximately 2 seconds for each IC sample.

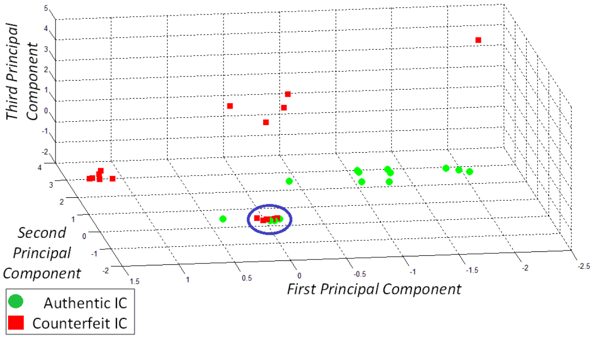

Fig. 8. Graph of the first three principal components of the counterfeit (red) and authentic (green) integrated circuits using linear PCA. Circled in blue is the classification area requiring precise fitting.

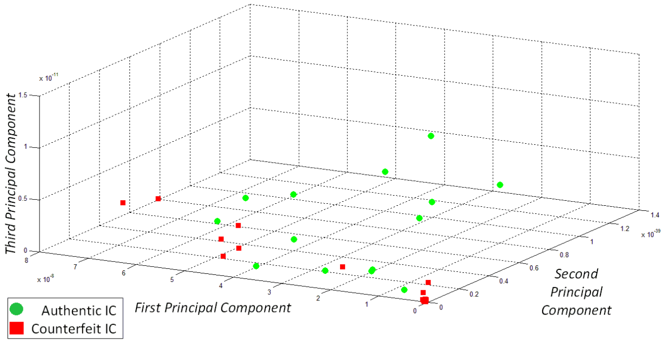

Fig.9 Graph of the first three principal components of the counterfeit (red) and authentic (green) integrated circuits using non-linear PCA (through a stacked autoencoder).

5. Conclusion

We have developed an effective technique using X-ray microscopy and a series of image processing and classification algorithms to successfully distinguish between authentic and counterfeit ICs. In our procedure, we first register the X-ray images using the affine transform, whose parameters are computed using linear least squares minimization, and then use LBPs to extract features. Dimensionality reduction is carried out using linear PCA, as well as non-linear PCA through a stacked autoencoder. We apply two separate classification approaches, one based on an SVM with a Gaussian kernel, and another using a deep neural network. Testing on a mix of several authentic and counterfeit chips results in a counterfeit detection rate of 87.5% using the SVM classifier, while the deep neural network using a stacked autoencoder achieves a 100% accurate detection rate. The better performance of a deep neural network (using a stacked autoencoder), over an SVM (with dimensionality reduction using PCA) has been observed in recent difficult classification problems [24] and has even been shown to come close to matching human visual perception [32]. In summary, our complete counterfeit detection procedure is highly accurate for the samples we have used in the experiments, runs in real time, is non-destructive, is independent of subject matter experts, and requires no additional information about the IC chip being tested. While we have primarily examined deep neural networks and SVM, a variety of other classification approaches could be considered in future work [33-39]. In addition, a variety of applicable microscopy approaches may be considered in the future [40].

Acknowledgements

This work was supported by the NSF grant NSF/CISE Award #CNS–1344271. The authors would also like to gratefully acknowledge the University of Connecticut Center for Hardware Assurance, Security and Excellence (CHASE) for their expertise and technical support contributing to the paper.

References

1. C. Levin, et al., "US Senate Committee on armed services Inquiry into counterfeit electronic parts in the department of defense supply chain", http://www.gpo.gov/fdsys/pkg/CRPT-112srpt167/pdf/CRPT-112srpt167.pdf.
2. F. Koushanfar, S. Fazzari, C. McCants, W. Bryson, M. Sale, P. Song, and M. Potkonjak, "Can eda combat the rise of electronic counterfeiting?" in *Design Automation Conference (DAC)* (IEEE, 2012), pp. 133–138.
3. M. Pecht and S. Tiku, "Bogus: electronic manufacturing and consumers confront a rising tide of counterfeit electronics," IEEE Spectrum **43**, 37–46 (2006).
4. M. Tehranipoor, H. Salmani, and X. Zhang, Integrated Circuit Authentication: Hardware Trojans and Counterfeit Detection (Springer, 2013).
5. U. Guin, D. DiMase, and M. Tehranipoor, "A Comprehensive Framework for Counterfeit Defect Coverage Analysis and Detection Assessment," Journal of Electronic Testing: Theory and Applications (JETTA) **30**, 25-40 (2014).
6. U. Guin, D. DiMase, and M. Tehranipoor, "Counterfeit Integrated Circuits: Detection, Avoidance, and the Challenges Ahead," Journal of Electronic Testing: Theory and Applications (JETTA) **30**, 9-23 (2014).
7. K. Huang, J. Carulli, and Y. Makris, "Parametric counterfeit IC detection via Support Vector Machines," in *IEEE Intl. Symp. On Defect and Fault Tolerance in VLSI and Nanotechnology Systems* (IEEE, 2012), pp. 7–12.
8. A. Pan, L. Xu, J. Petruccelli, R. Gupta, B. Singh, and G. Barbastathis, "Contrast enhancement in X-ray phase contrast tomography," Opt. Express **22**, 18020-18026 (2014).
9. A. Bronnikov and G. Duifhuis, "Wavelet-based image enhancement in x-ray imaging and tomography," Appl. Opt. **37**, 4437-4448 (1998).
10. G. Skinner, "Design and imaging performance of achromatic diffractive–refractive x-ray and gamma-ray Fresnel lenses," Appl. Opt. **43**, 4845-4853 (2004).
11. B. Arhatari, G. Riessen, and A. Peele, "Polychromatic X-ray tomography: direct quantitative phase reconstruction," Opt. Express **20**, 23361-23366 (2012).
12. G. Cheng, C. Hu, P. Xu, and T. Xing, "Zernike apodized photon sieves for high-resolution phase-contrast x-ray microscopy," Opt. Lett. **35**, 3610-3612 (2010).
13. O. Hofsten, M. Bertilson, J. Reinspach, A. Holmberg, H. Hertz, and U. Vogt, "Sub-25-nm laboratory x-ray microscopy using a compound Fresnel zone plate," Opt. Lett. **34**, 2631-2633 (2009).

14. T. Moore, D. Vanderstraeten, P. Forssell, "Three-dimensional x-ray laminography as a tool for detection and characterization of BGA package defects," IEEE Trans. on Components and Packaging Technologies **25**, 224-229 (2002).
15. H. Rositi, C. Frindel, M. Langer, M. Wiart, C. Olivier, F. Peyrin, and D. Rousseau, "Information-based analysis of X-ray in-line phase tomography with application to the detection of iron oxide nanoparticles in the brain," in Applied Optics **21**, 27185-27196 (2013).
16. A. Faust, R. Rothschild, P. Leblanc, J. McFee, "Development of a Coded Aperture X-Ray Backscatter Imager for Explosive Device Detection," IEEE Trans. on Nuclear Science **56**, 299-307 (2009).
17. M. Uehara, W. Yashiro, A. Momose, "Effectiveness of X-ray grating interferometry for non-destructive inspection of packaged devices," Journal of Applied Physics **114**, 134901-134906 (2013).
18. J. P. W. Pluim, J. B. A. Maintz, and M. A. Viergever, "Mutual-information-based registration of medical images: a survey," IEEE Trans. Med. Imag. **22**, 986–1004 (2003).
19. J.P. Lewis, "Fast Normalized Cross-Correlation," Vision Interface, 120–123 (1995).
20. C. Lawson, R. Hanson, "Solving Least Squares Problems," (Prentice-Hall, 1974).
21. D. Huang, C. Shan, M. Ardabilian, Y. Wang, and L. Chen, "Local binary patterns and its application to facial image analysis: A survey," IEEE Trans. Syst., Man, Cybern. C, Applicat. Rev. **41**, 765–781 (2011).
22. B. E. Boser, I. M. Guyon, V. N. Vapnik, "A training algorithm for optimal margin classifiers," in *Proceedings of the 5th Annual Workshop on Computational Learning Theory* (ACM, 1992), pp. 144–152.
23. C. Cortes, V. Vapnik, "Support-vector networks," Machine Learning **20**, 273–297 (1995).
24. Y. Bengio and Y. LeCun, "Scaling learning algorithms towards AI," in Large Scale Kernel Machines **34** (2007).
25. S. Haykin, "Neural Networks and Learning Machines," (Pearson, 2009).
26. Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, "Greedy layer-wise training of deep networks," in *Adv. in Neural Information Processing Systems* (2007), pp. 153-160.
27. V. N. Marivate, F. V. Nelwamondo, and T. Marwala, "Investigation into the use of Autoencoder Neural Networks, Principal Component Analysis and Support Vector Regression in estimating missing HIV data," in *Proceedings of the 17th World Congress of The International Federation of Automatic Control Seoul, Korea* (2008), pp. 682-689.
28. G. Cybenko, "Approximation by superpositions of a sigmoidal function," Mathematics of Control, Signals, and Systems **2**, 303–314 (1989).
29. N. Japkowicz, M. Shah, "Evaluating Learning Algorithms: A Classification Perspective", (Cambridge University Press, 2011).
30. G. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," Science, **313**, 504-507 (2006).
31. L. Arnold, S. Rebecchi, S. Chevallier, and H. Paugam-Moisy, "An introduction to deep learning," in *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning* (2011), pp. 477-488.
32. Y. Taigman, M. Yang, M. Ranzato, L. Wolf, "Deepface: Closing the gap to human level performance in face verification," in *Conference on Computer Vision and Pattern Recognition* (IEEE, 2014).
33. F. Sadjadi and A. Mahalonobis, "Target adaptive polarimetric SAR target discrimination using MACH filters," Appl. Opt. **45**, 7365-7374 (2006).
34. A. Mahalanobis, R. Muise, "A compressive sensor concept for target detection," in *Proceedings of SPIE 8398* (SPIE, 2012).
35. F. Dubois, "Automatic spatial frequency selection algorithm for pattern recognition by correlation," Applied Optics **32**, 4365-4371 (1993).
36. P. Refregier, "Noise Theory and Application to physics," (Springer, 2003).
37. F. Sadjadi, A. Mahalanobis, "Automatic Target Recognition," SPIE **8744**, ISBN: 9780819495358, 2013.
38. M. Ruiz-Llata and H. Lamela-Rivera, "Image identification system based on an optical broadcast neural network processor," Appl. Opt. **44**, 2366-2376 (2005).
39. M. Pohit, "Neural network model for rotation invariant recognition of object shapes," Appl. Opt. **49**, 4144-4151 (2010).
40. P. Ferraro, A. Wax, Z. Zalevsky, "Coherent Light Microscopy: Imaging and Quantitative Phase Analysis," (Springer, 2011).