Computer Vision for Hardware Trojan Detection on a PCB Using Siamese Neural Network

Gor Piliposyan
Dept. of Electrical Engineering and Electronics
University of Liverpool
Liverpool, UK
Gor.Piliposyan@liverpool.ac.uk

Saqib Khursheed
Dept. of Electrical Engineering and Electronics
University of Liverpool
Liverpool, UK
S.Khursheed@liverpool.ac.uk

***Abstract*—With advances in technology Hardware Trojan (HT) attacks on printed circuit boards (PCB) are becoming more sophisticated and the need for more effective HT detection methods is becoming crucial. Automated visual inspection (AVI) is one of the most promising solutions in detecting malicious implants on a PCB. It is non-destructive, effective in testing PCBs on an industrial scale, demands minimum human involvement, and can potentially identify malicious inclusions and modifications on PCBs at all stages of production and thereafter. In recent years, machine learning algorithms have been successfully applied, significantly improving the effectiveness of AVI methodologies. In this paper, an AVI methodology is proposed for detecting HTs on a PCB, using input data from a low-cost digital optical camera. It is based on a combination of conventional computer vision techniques and a dual tower Siamese Neural Network (SNN), modelled in a three stage pipeline. Further, a dataset of PCB images has been developed in a controlled environment of a photographic tent. The results show that the methodology has an average 95.6% classification accuracy for PCBs with HT inclusions with surface area between $4\ mm^2$ and $280\ mm^2$.**

***Index Terms*—PCB Inspection, Hardware Trojan Detection, Deep Learning, Automated Visual Inspection, Siamese Neural Network, Computer Vision**

I. Introduction

Several printed circuit board (PCB) assurance techniques have already been suggested and evaluated over the past years. These include in-circuit testing, functional testing, Joint Test Action Group (JTAG) boundary scanning and bare-board testing. Each of these assurance methods have advantages and limitations, i.e. situations where they can be less effective [1]. With advances in technology Trojan attacks become more sophisticated and so there is a growing demand for more effective Trojan detection methods. Ideally, the most effective solution would require minimal human involvement, be non-destructive and be able to detect as many types of malicious modifications, inclusions and defects on a PCB as possible. For example, a run time PCB monitoring technique proposed in [2] is based on power consumption with no human intervention. Power analysis methods, however, are inherently limited by the resolution of the sensors they use to probe, meaning ultra low power Trojans can evade detection.

Automated visual inspection (AVI) has the potential to satisfy all these expectations. It requires minimum human involvement and can quickly test a large number of PCBs. Unlike the above mentioned PCB assurance methods which can be used either when a board is fully populated or unpopulated, AVI can potentially identify malicious inclusions, modifications and defects on PCBs at all stages of production and even after sale. It has many advantages over manual visual inspection which is slower, more expensive, less effective and subject to human error [3], [4].

Automated visual inspection can be implemented in three steps, image acquisition, image analysis and authentication [1]. Image acquisition can be done using several imaging modalities [5], which can be categorised into three groups - surface, subsurface and volumetric. Depending on the requirements, multi-modal imaging approach may also be applied to detect, for example, malicious modifications between PCB layers or active components disguised as passive.

Collected images then need to be analysed for possible defects or malicious inclusions and modifications. Image analysis involves processing, feature extraction and classification stages. First the acquired images are processed to improve the quality, for example, by removing noise, altering illumination and enhancing contrast. The next stage is feature extraction when key characteristics such as shape, color and texture of the pursued objects are captured. This is followed by classification and grouping of similar style components such as metal traces, vias, integrated circuits (IC), capacitors, transistors or resistors [6]. With advances in deep learning methods, feature extraction and classification can be performed simultaneously using deep learning algorithms [1]. Text recognition is also used in classification for identification of markings such as serial numbers, which can be used, for example, for detection of counterfeit components by comparing the component's serial number with the manufacturer's original equipment serial number.

The final stage of automated visual inspection (AVI) is authentication where data is stored as a Computer Aided Design (CAD) file for comparing images of a fabricated PCB with the image of a golden PCB model [6].

Although AVI has been the most commonly used method for PCB assurance since 1960s [7] it had several limitations such as limited-area inspection, hard-coded specifications, and significant amount of subject matter expert involvement [8], [9]. Advances in deep learning in recent years, in particular image recognition, localization and segmentation, have been

successfully applied to AVI to overcome these limitations. Several AVI methods have been suggested, including canonical image processing method for detection, classification and localisation of several specific types of defects on a PCB [10], convolutional neural network for detecting six types of defects [11], automated detection for component placement by directly comparing golden and test PCBs [12], [13] and text detection on the PCB for verification purposes [14], [15].

All the suggested approaches designed for PCB defect detection do not distinguish between irregularities that are due to manufacturing defects and Hardware Trojan (HT) inclusions, which are malicious modifications e.g. for compromising sensitive information. In the Big Hack [16], for example, an extra component implanted on a PCB was an HT which was visually disguised as a legitimate component, albeit being marginally larger in size. The HT was designed to provide administrative access to the network for an outside attacker. Development of AVI approaches for monitoring the location, size, and appearance of PCB components, focusing in particular on HT detection, is very important [6]. This paper addresses that problem by proposing a novel optical AVI algorithm for HT detection on a PCB.

II. Previous work

In recent years AVI has been applied using several methods including image matching, feature extraction, and deep learning. Each of these approaches demonstrated effective performances in various defect detection tasks such as component, trace and via defect detection, and component classification.

Image matching methods have been mostly applied for detecting missing, displaced or replaced components [13], [17]–[19]. In particular, by using background subtraction 90% accuracy is achieved in [18] on detecting missing capacitors and resistors. By matching wavelet-transformed images, component inspection in electronic assembly lines is suggested in [19] with 86% accuracy. Feature extraction has also been adopted to classify component defects [12]. While image matching checks the whole PCB, feature extraction only inspects regions where illegitimate components are anticipated.

Feature extraction and deep learning methods have proven to be efficient for component classification [20], [21]. In [22] an automatic surface mount device classification method extracted color and edge information from color images of PCB parts. A neural network was used to classify chip-type packages and 97.6% average classification accuracy was achieved after adding additional edge information. Using a convolution neural network, authors in [23] and [24] also proposed a component classification method. Based on component images obtained from a PCB the method suggested in [23] separated components from their backgrounds and classified them, achieving 90.8% accuracy. By training IC images collected online, IC components were identified with 92.3% accuracy in [24].

Previous works have looked into AVI for quality assurance, however HTs pose a separate challenge. The novelty in this work is that the algorithm has been optimised and trained specifically for implanted HT component detection on a PCB. The proposed HT detection methodology has been trained and tested with three groups of HTs, categorised based on their surface area. The results show that it is possible to reach effective detection accuracy of 95.1% for HTs as small as $4\ mm^2$. In case of HTs with surface area larger than $280\ mm^2$ the detection accuracy is around 96.1%, while the average performance across all HT groups is 95.6%. These results can be further improved if higher resolution images are used.

The rest of the paper is organised as follows: the proposed methodology is presented in Section III. Section IV describes the experimental setup used for carrying out the research, with the results discussed in Section VI. Finally, the results of the paper are summarised in Section VII.

III. Proposed Methodology

The goal of this work is to develop a low-cost and fast PCB visual inspection tool. This is achieved by avoiding expensive and slow imaging modalities such as X-rays or high end microscopes. Instead, the approach adopted in the current paper is to detect HT contaminated PCBs by using a simple digital optical camera. This fundamental characteristic of the proposed method allows to develop an AVI tool with marginal time and resource overheads, inspecting all PCBs passing through a conveyor belt setup on production lines.

The proposed methodology pipeline is comprised of the following key stages

- Image alignment homography
- Application of Gaussian blurring
- Background image subtraction
- Suspicious region identification
- Cropping suspicious regions as image pairs
- Siamese Neural Network similarity estimation
- Confirmed dissimilar region marking on PCB
- Labelling the PCB on HT presence status.

It includes conventional computer vision techniques such as image alignment through homography, blurring filter and background subtraction. To align two images pixel coordinates of one of them should be multiplied by the homography matrix:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} \sim \mathbf{H} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \sim \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}, \tag{1}$$

$$x' = \frac{h_{11}x + h_{12}y + h_{13}}{h_{31}x + h_{32}y + h_{33}}, \quad y' = \frac{h_{21}x + h_{22}y + h_{23}}{h_{31}x + h_{32}y + h_{33}}, \tag{2}$$

where $\mathbf{H}$ is the estimated homography matrix, and (x', y') are the updated coordinate estimates of the pixel previously in location (x, y). Following image alignment phase a blurring filter kernel is applied to both images. This is done to smooth out minor misalignments on the edges of objects (e.g. wires, chips). In this work the kernel grid has been populated with a two-dimensional Gaussian distribution function $G(x, y)$, a.k.a. Gaussian blur filter:

$$G(x, y) = \frac{1}{2\pi\sigma^2} \mathrm{e}^{-\frac{(x^2 + y^2)}{2\sigma^2}}, \tag{3}$$

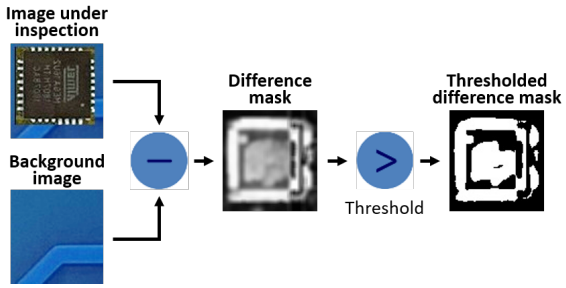


Fig. 1: Image subtraction and pixel value binary thresholding.

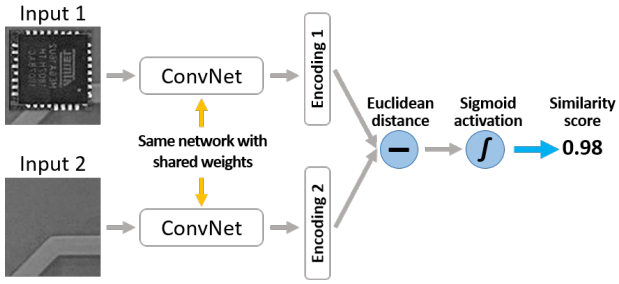


Fig. 2: Siamese Neural Network architecture overview.

where $\sigma_x = \sigma_y = \sigma$ is the standard deviation on both axes. Finally, image background subtraction and thresholding operations are demonstrated in Fig. 1. These are applied on whole PCB images, after which the resulting individual contours are regarded as suspicious regions and cropped out as separate images for later processing.

The proposed methodology also includes deep learning architectures such as convolutional neural network and feed-forward fully connected neural network. These are merged into a specific architecture called Siamese Neural Network (SNN) [25], [26]. The SNN has a particular type of neural network architecture (Fig. 2) where some weights are shared between two towers of convolutional neural networks. Each tower produces an embedding vector of its respective input image. Given a dataset of pairs of inputs, the network is trained to maximize the distance between the embeddings of the inputs coming from different classes, while minimizing the distance between embeddings coming from inputs of the same class. This process is referred to as supervised similarity learning. The particular choice of using SNN as the base algorithm comes naturally given the underlying problem being solved, i.e. comparing two images to detect differences.

In order to obtain a good quality dataset and minimise the ambient optical impact, the PCBs have been placed in a photographic tent with a built-in diffused light source (Fig. 3a). This way, when capturing the photographs, all PCBs have similar initial environmental conditions, independent from many external factors such as daylight, shifting shadows, color variations due to reflections from the surroundings [27]. On one hand it could be argued that outside of the laboratory's controlled environment the ambient lighting conditions could vary. On the other hand, it can also be assumed that implementing a photographic tent-like structure in a factory production line can be achieved with little extra effort. The motivation for using a photographic tent to begin with is to boost the algorithm performance by removing unnecessary complications (e.g. changing light source color or angle of incidence). This is a low-effort but high-impact improvement of the input data. Further, while producing the images the camera has been mounted on a static stand and a remote controlled shutter has been used to produce stable images.

IV. Experimental Setup

Capturing photographs: To build the PCB image dataset a $52cm \times 52cm \times 52cm$ FOSITAN photographic tent with an opening on the top and a built-in intensity-adjustable white LED light has been used. The intensity of the light source at the surface level of the PCB has been kept at 5380 Lux, with the measurements being taken by an AP-8801C digital light meter (Fig. 3b). Further, to minimise artifacts in the image, for example due to reflective surfaces on the PCB, several layers of light diffusing cloth sheets have been applied between the light source and the PCB, which is a standard practice for controlling image quality. Using this environment 101 images of a PCB have been captured of which 1 has been used as the golden model PCB and the other 100 were later used as source images for creating a much larger dataset of HT infected PCBs. Regarding the digital camera used to produce the images, a 12 Mp camera with a 1 cm sensor size and 1.4 μm pixel pitch has been utilised. The camera has a 26 mm equivalent focal length and f/1.8 aperture lens. The objective of this research is to develop a high quality automated visual inspection (AVI) algorithm which can work with moderate quality input images acquired through low cost digital optical camera modules.

Preprocessing photographs: Before two images can be compared with each other to detect differences, they have to be aligned. This is a crucial step in the proposed AVI pipeline. During image alignment process one image is warped to match the second. This effect is achieved by multiplying every coordinate in the image by the homography matrix $\mathbf{H}$ as shown in (1). It is important to note that homography can only be applied to objects on a 2D plane such as a PCB. In

(a) (b)

Fig. 3: (a) Photographic tent used for creating image dataset, (b) digital light meter used for light intensity control.

TABLE I: Groups of Hardware Trojans.

Group name	Small	Medium	Large
Surface area in $\mathbf{mm^2}$	4 to 9	15 to 50	280+
Surface area in $\mathbf{pixels^2}$	700 to 1500	2500 to 9000	50000+

this research OpenCV library has been used to compute the homography matrix and perform image alignment [28].

In order to perform homography, first the same points of the object (e.g. PCB) in both images need to located. The minimum required number of such point pairs to be able to perform homography is four. Increasing the number of such pairs will result in a more robust homography estimation and, hence, improved alignment of images. To acquire such point pairs in an automated manner, first several key-points on both images need to be located. Then these key-points should be matched into pairs. For example, it can be achieved with brute force matching by searching for the minimum euclidean distance between every couple of descriptor vectors belonging to a particular pair of key-points. Points referred to as key-points are typically distinctive corners, edges or sharp curves of the objects (e.g. PCB) present in the image. They are defined by (x, y) coordinates, size and orientation. Their respective descriptors, on the other hand, are unique markers of the key-points, independent from the orientation of the object in the image. The descriptors are vectors calculated internally by OpenCV [28] and help in search for matching key-point pairs from two different images of the same item.

At a later stage, another algorithm called Random Sample Consensus (RANSAC) is used to discard the key-point pairs with a high likelihood of being outliers. Such points can have a significant negative impact on the quality of the homography matrix. RANSAC is an iterative algorithm which validates a mathematical model built using a dataset with outliers [29]. The algorithm assumes that the dataset is a combination of both inliers and outliers. Inliers can be identified by a model with a special set of parameter values, whereas outliers do not fit the model in any condition. The iteration process repeats a fixed number of times and each time produces either a model which is rejected due to very few points being part of the consensus set, or produces a refined model together with a corresponding updated consensus set. The refined model is accepted only if the size of the updated consensus set is larger then that of the previous model.

Inserting Hardware Trojans: Three groups of HTs have been used in this research. They have been binned into groups of small, medium and large, based on their surface area (Table I). The HTs have been added with the help of Flip library on GitHub [30] developed by LinkedAI. The library is used for synthetic data generation on new 2D images from a batch of objects and backgrounds. In the scope of this research the background image is an HT free PCB Under Inspection (PUI), while the objects are the HTs. The idea is to take a random background image and a random object and place the object in a random location with a random integer multiple of 90° rotations. On top of that, the Flip library provides many of the conventional image augmentation functionalities, e.g. resize or colour shift the objects. Using Flip library and the 100 source images captured earlier as backgrounds, 7500 images of PCBs with inserted HT devices have been generated, i.e. 2500 images per HT group.

V. Hardware Trojan Detection Pipeline

The proposed HT detection pipeline consists of three main stages shown in Fig. 4. In the first stage the images of the golden model (GM) and a PCB under inspection (PUI) are compared to identify the suspicious regions on the PUI, where an HT could be present, but in this stage there is no definitive prediction whether that is the case. Instead, the information about these regions is passed forward to the next stage of the pipeline as a list of bounding boxes. In the second stage of the pipeline the algorithm uses the bounding boxes to crop out these sections as a set of smaller images. This step is repeated for both the GM and the PUI to create pairs of images. The resulting small image pairs are later normalised to 28×28 pixels and converted to grayscale. The second stage of the pipeline is illustrated in more detail in Fig. 5. Next, the normalised image pairs of the suspicious regions are passed on to the final third stage of the pipeline - the Siamese Neural Network (SNN). Normalizing the image sizes is necessary for the SNN since it can only work with a specific shape, while converting images to grayscale greatly reduces the number of parameters in the neural network. The SNN parses all image pairs, outputting a similarity score for every image pair.

In this work, the Keras framework [31] has been used for implementing the Siamese Neural Network (SNN). The network is trained to differentiate between the HT contaminated and HT clean image pairs. In case of the HT clean image pair class the two cropped images from GM and PUI should be very similar. The reason why such regions have been suggested by the previous stage of the pipeline is that although the image alignment algorithms have a good performance, they are not perfect and in some cases a slight misalignment gets interpreted as difference. This is where the SNN excels at differentiating between a misalignment and actual HT presence. Finally a threshold is applied to the similarity scores followed by a logic AND function to check if at least one HT is present on the PUI.

Motivational example: To explain the proposed methodology, a simplified case study can be considered, where 1000 PUIs are available, 100 out of which have an HT on board. The first stage of the proposed pipeline (Fig. 4) is optimised to mark all 100 HTs in suspicious regions, even though in the process the algorithm may also wrongly suggest many misidentified suspicious regions. For example, the algorithm may miss 1 HT, correctly mark 99 HT regions and further suggest 401 misidentified regions which do not contain an HT. These 500 suspicious region coordinates are cropped from the GM and the respective PUIs, normalised (Fig. 5) and passed to the SNN in the last stage of the pipeline (Fig. 4). The SNN individually compares all 500 image pairs to assess their similarity. For example, the SNN may have 95% accuracy,

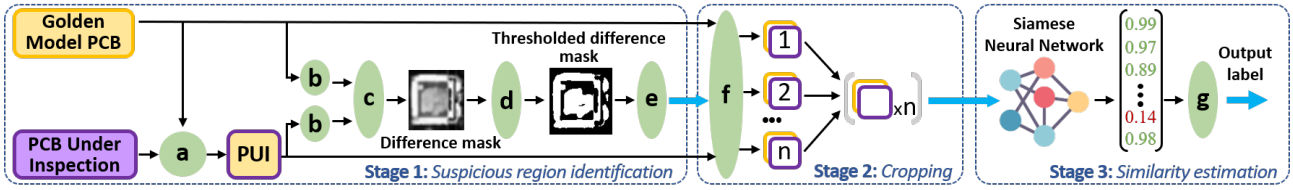


Fig. 4: The proposed Hardware Trojan detection pipeline, where: (a) Image alignment with homography, (b) Gaussian blurring kernel, (c) Background image subtraction, (d) Binary thresholding, (e) Suspicious region identification, (f) Cropping suspicious regions as image pairs from GM and PUI, (g) Labelling the PUI on HT presence status.

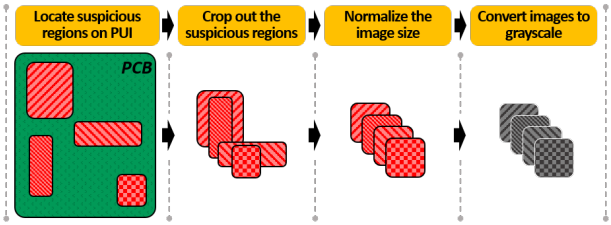


Fig. 5: Cropping and normalisation of suspicious regions.

TABLE II: Hardware Trojan detectability in top-left blue and rate of misidentified suspicious regions in bottom-right red.

Cutoff threshold	Small HT	Medium HT	Large HT
15	99.6% / 98.1%	99.1% / 95.9%	100% / 98.0%
25	99.4% / 96.2%	99.4% / 91.2%	100% / 95.7%
35	99.6% / 94.4%	99.7% / 87.6%	100% / 93.9%
45	99.6% / 93.0%	99.6% / 84.8%	100% / 92.3%
55	93.4% / 88.4%	98.2% / 76.8%	100% / 87.3%

detecting 94 out of 99 HTs. In such scenario 99 out of 100 HTs are detected by the first stage (99% detectability) and 94 out of those 99 HTs are detected in the final stage (95% accuracy) resulting in an overall effective 94% accuracy.

VI. Experimental Results

Since the proposed methodology is organised in a multistage structure (Fig. 6), it is possible to retrieve meaningful outputs from the intermediate stages. In fact, this is a crucial step in the overall optimisation process. In this paper the methodology has been optimised for two independent consecutive results.

Image thresholding: The stage of suspicious region identification through image thresholding has undergone a constrained optimisation problem, whereby the HT detection rate, i.e. detectability, has been maximised, such that the ratio of misidentified suspicious regions to the total number of predicted suspicious regions does not exceed 95%. Here detectability is defined by the ratio of all HTs which were included in at least one of the suggested suspicious regions. The suspicious region is defined as misidentified if it does not overlap with an HT or if the intersection over union (IOU) is below 10%. In other words, this algorithm has been optimised to find as many HT containing regions as possible, while keeping the rate of wrongly suggested suspicious regions in a reasonable range. This constraint on the optimisation has been introduced to avoid the trivial case of having the algorithm mark all of the PCB surface as suspicious. The optimisation was accomplished by calibrating the pixel value cutoff threshold for image binary thresholding (Fig. 7) using the 1 GM and 7500 PUI images with their ground truth masks of HT locations. The results for HT detectability, alongside with the respective rate of misidentified proposed suspicious regions, are presented in Table II, subject to varying cutoff thresholds. The cells satisfying the constraint of keeping the rate of misidentified suspicious regions below 95% have been highlighted with a light green background. The reason why the algorithm can afford to output so many misidentified suspicious regions is that the SNN in the last stage of the pipeline (Fig. 6) can discard them with high accuracy. The goal here is to mark as many HT containing regions as possible.

Siamese Neural Network: A Siamese Neural Network with around 872×10^3 trainable parameters has been trained to discern between the images of the same patches on two PCBs harbouring an HT component, while being able to recognise similar patches which are only slightly misaligned or misshaped (Fig. 8). The root cause of having such misaligned patches is the estimation of the homography matrix $\mathbf{H}$ in (1). The reason for having misshaped patterns could be, among other things, variations in the PCB production process as well as defects such as misaligned elements.

Information about the datasets used to train, validate and test the SNN model and their respective resulting prediction accuracies are shown in Table III. The datasets are comprised of pairs of images, where each pair represents one of the suspicious regions. Inside every pair the first image is cropped from the suspicious region on the PCBs under inspection, while the second is the matching region on the golden model PCB. These images are of the exact same regions on both PCBs, cropped

TABLE III: Siamese Neural Network prediction accuracy.

Dataset name	Train	Validation	Test
Dataset size	18000	6000	6000
Prediction accuracy (*small HT*)	98.8%	96.5%	**95.5%**
Prediction accuracy (*medium HT*)	98.4%	97.6%	**95.9%**
Prediction accuracy (*large HT*)	98.7%	98.5%	**96.1%**

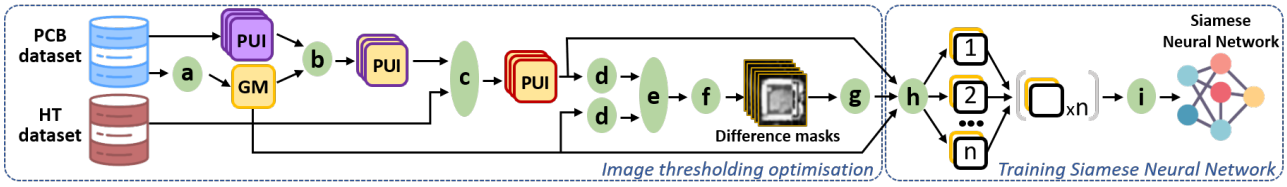
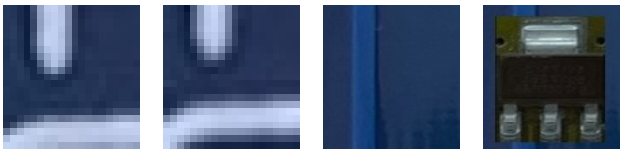


Fig. 6: Training process of the proposed methodology, where: (a) Choose one as the golden model (GM), (b) Align the remaining images to the GM, (c) Insert HTs on every PCB under inspection (PUI), (d) Gaussian blurring kernel, (e) Subtract GM from every PUI, (f) Binary thresholding, (g) Suspicious region identification, (h) Cropping suspicious regions as image pairs from their respective pre-blurred PUI and GM images, (i) Train Siamese Neural Network.

(a) *Source img.* (b) *thr.* $= 150$ (c) *thr.* $= 100$

Fig. 7: Given a grayscale source image (a), computed binary thresholding images with high (b) and low (c) threshold values.



(a) *Golden model PCB* (b) *PCB under inspection* (c) *Golden model PCB* (d) *PCB under inspection*

Fig. 8: Cropped suspicious regions. Image pair (a) and (b) are only misaligned, pair (c) and (d) harbour a Trojan component.

TABLE IV: Effective prediction accuracy.

HT size group	Small	Medium	Large	**All**
Image thresholding *(at 35)*	99.6%	99.7%	100%	**99.8%**
Siamese Neural Network	95.5%	95.9%	96.1%	**95.8%**
Effective accuracy	**95.1%**	**95.6%**	**96.1%**	**95.6%**

out based on coordinates from a single bounding box and later normalised as shown in Fig. 5. In total about $30K$ such image pairs have been collected and split into training, validation and testing datasets with a $60\% - 20\% - 20\%$ ratio. The performance of SNN has been analysed per HT group (Table III). The classification accuracies on testing dataset range from 95.5% to 96.1%, as expected, performing best on large HTs.

Effective accuracy: Since the proposed methodology has two consecutive, distinct and independent outputs with their respective accuracy scores, the effective accuracy of the methodology as a whole is the multiplication of the two. In other words, HT detection accuracy of the SNN applies only to the HTs which had previously been detected by the previous stage of the pipeline. For example, in case of medium size HTs the suspicious region identification with image thresholding stage resulted in HT detectability rate of 99.7% and the SNN had classification accuracy of 95.9%. The resulting effective accuracy of the methodology for medium size HTs is $(99.7\% \times 95.9\%) = 95.6\%$. The effective accuracies for all groups of HTs are presented in Table IV. As expected, the algorithm performance improves up to 96.1% as the HTs get larger, with the overall HT implanted PCB detection accuracy being around 95.6%.

VII. Conclusion

This paper proposes a methodology for detecting Hardware Trojan (HT) components on a printed circuit board (PCB) through automated visual inspection. It is assumed that an image of a trusted golden model (GM) of the PCB is available with which comparisons are made. The proposed technique provides an accurate and fast tool to detect HT inclusions on PCBs using a low-cost imaging modality - optical digital camera. To keep the operating conditions stable and avoid the negative impacts from variations in ambient lighting, a photographic tent (Fig. 3a) with internal diffused light source has been used to develop a PCB image dataset containing 7500 plus 1 images including the golden model. The proposed methodology is a pipeline of three sub-stages (Fig. 4). Initially, the first stage proposes suspicious regions on the PCB Under Inspection (PUI), where a potential HT could be located. In the second stage, these regions are cropped out as pairs of images from both the GM and PUI. In total 30000 such pairs of images are preprocessed (Fig. 5) preparing them for the final stage. In the final stage, a Siamese Neural Network (SNN) takes each of these 30000 image pairs as two separate inputs and outputs a similarity estimation. The results show that the proposed automated visual inspection pipeline, combining conventional computer vision techniques and deep learning, can detect HT devices with surface area from $4\ mm^2$ to $280\ mm^2$ implanted on a PCB with an effective accuracy of 95.6% (Table IV). Detection of ultra-small HT components with surface area under $4\ mm^2$ will be addressed in our future work.

Acknowledgment

This project was funded by the Department of Electrical Engineering and Electronics, University of Liverpool, UK.

References

[1] D. Mehta, H. Lu, O. P. Paradis, M. M. Azhagan, M. T. Rahman, Y. Iskander, P. Chawla, D. L. Woodard, M. Tehranipoor, and N. Asadizanjani, "The big hack explained: Detection and prevention of pcb supply chain implants," *J. Emerg. Technol. Comput. Syst.*, vol. 16, no. 4, 2020. [Online]. Available: https://doi.org/10.1145/3401980

[2] G. Piliposyan, S. Khursheed, and D. Rossi, "Hardware trojan detection on a pcb through differential power monitoring," *IEEE Transactions on Emerging Topics in Computing*, 2022. [Online]. Available: https://dx.doi.org/10.1109/TETC.2020.3035521

[3] H. Lu, D. E. Capecci, D. F. Pallabi Ghosh, and D. L. Woodard, *Computer Vision for Hardware Security*. Springer, 2021, ch. 18, pp. 493–527. [Online]. Available: https://doi.org/10.1007/978-3-030-64448-2_18

[4] G. Acciani, G. Brunetti, and G. Fornarelli, "Application of neural networks in optical inspection and classification of solder joints in surface mount technology," *IEEE Transactions on industrial informatics*, vol. 2, no. 3, pp. 200–209, 2006. [Online]. Available: https://doi.org/10.1109/TII.2006.877265

[5] J. Grzyb, K. Statnikov, R. A. Hadi, and U. R. Pfeiffer, "All-silicon integrated thz harmonic source and receiver components for future active imaging modalities," in *2014 39th International Conference on Infrared, Millimeter, and Terahertz waves (IRMMW-THz)*, 2014, pp. 1–2. [Online]. Available: https://doi.org/10.1109/IRMMW-THz.2014.6956429

[6] M. Azhagan, D. Mehta, H. Lu, and S. Agrawal, "A review on automatic bill of material generation and visual inspection on pcbs," in *International Symposium for Testing and Failure Analysis*, 2019, pp. 256–265. [Online]. Available: https://doi.org/10.31399/asm.cp.istfa2019p0256

[7] L. Watkins, "Inspection of integrated circuit photomasks with intensity spatial filters," *Proceedings of the IEEE*, vol. 57, no. 9, pp. 1634–1639, 1969. [Online]. Available: https://doi.org/10.1109/PROC.1969.7348

[8] M. Moganti, F. Ercal, C. H. Dagli, and S. Tsunekawa, "Automatic pcb inspection algorithms: A survey," *Computer Vision and Image Understanding*, vol. 63, no. 2, pp. 287–313, 1996. [Online]. Available: https://doi.org/10.1006/cviu.1996.0020

[9] W.-C. Wang, S.-L. Chen, L.-B. Chen, and W.-J. Chang, "A machine vision based automatic optical inspection system for measuring drilling quality of printed circuit boards," *IEEE Access*, vol. 5, pp. 10 817–10 833, 2017. [Online]. Available: https://10.1109/ACCESS.2016.2631658

[10] P. S. Malge and R. S. Nadaf, "Pcb defect detection, classification and localization using mathematical morphology and image processing tools," *International Journal of Computer Applications*, vol. 87, no. 9, pp. 40–45, February 2014.

[11] S. Tang, F. He, X. Huang, and J. Yang, "Online pcb defect detector on a new pcb defect dataset," 2019. [Online]. Available: https://doi.org/10.48550/ARXIV.1902.06197

[12] H.-H. Wu, X.-M. Zhang, and S.-L. Hong, "A visual inspection system for surface mounted components based on color features," in *International Conference on Information and Automation*, 2009, pp. 571–576. [Online]. Available: https://doi.org/10.1109/ICINFA.2009.5204988

[13] A. Crispin and V. Rankov, "Automated inspection of pcb components using a genetic algorithm template-matching approach," *Int Journal Adv Manuf Technol*, vol. 35, p. 293–300, 2007. [Online]. Available: https://doi.org/10.1007/s00170-006-0730-0

[14] C. Szymanski and M. R. Stemmer, "Automated pcb inspection in small series production based on sift algorithm," in *2015 IEEE 24th International Symposium on Industrial Electronics (ISIE)*, 2015, pp. 594–599. [Online]. Available: https://doi.org/10.1109/ISIE.2015.7281535

[15] R. Smith, "An overview of the tesseract ocr engine," in *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)*, vol. 2, 2007, pp. 629–633. [Online]. Available: https://doi.org/10.1109/ICDAR.2007.4376991

[16] J., Robertson and M., Riley, "The big hack: How china used a tiny chip to infiltrate u.s. companies," https://bloomberg.com/news/features/2018-10-04/the-big-hack-how-china-used-a-tiny-chip-to-infiltrate-america-s-top-companies/, 2018.

[17] F. Xie, A. Uitdenbogerd, and A. Song, "Detecting pcb component placement defects by genetic programming," in *2013 IEEE Congress on Evolutionary Computation*. IEEE, 2013, pp. 1138–1145.

[18] K. Sundaraj, "Pcb inspection for missing or misaligned components using background subtraction," *WSEAS Transactions on Information Science and Applications archive*, vol. 6, pp. 778–787, 2009.

[19] H. Cho and T. Park, "Wavelet transform based image template matching for automatic component inspection," *Int Journal Adv Manuf Technol*, vol. 50, p. 1033–1039, 2010. [Online]. Available: https://doi.org/10.1007/s00170-010-2567-9

[20] O. P. Paradis, N. T. Jessurun, M. Tehranipoor, and N. Asadizanjani, "Color normalization for robust automatic bill of materials generation and visual inspection of pcbs," in *ISTFA 2020*, 2020, pp. 172–179. [Online]. Available: https://doi.org/10.31399/asm.cp.istfa2020p0172

[21] W. Zhao, S. R. Gurudu, S. Taheri, S. Ghosh, M. A. Mallaiyan Sathiaseelan, and N. Asadizanjani, "Pcb component detection using computer vision for hardware assurance," *Big Data and Cognitive Computing*, vol. 6, no. 2, p. 39, 2022. [Online]. Available: https://doi.org/10.3390/bdcc6020039

[22] S. Youn, Y. Lee, and T. Park, "Automatic classification of smd packages using neural network," in *IEEE/SICE International Symposium on System Integration*. IEEE, 2014, pp. 790–795. [Online]. Available: https://doi.org/10.1109/SII.2014.7028139

[23] D.-u. Lim, Y.-G. Kim, and T.-H. Park, "Smd classification for automated optical inspection machine using convolution neural network," in *2019 Third IEEE International Conference on Robotic Computing (IRC)*. IEEE, 2019, pp. 395–398. [Online]. Available: https://doi.org/10.1109/IRC.2019.00072

[24] M. A. Reza, Z. Chen, and D. J. Crandall, "Deep neural network–based detection and verification of microelectronic images," *Journal of Hardware and Systems Security*, vol. 4, no. 1, pp. 44–54, 2020.

[25] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, and R. Shah, "Signature verification using a" siamese" time delay neural network," *Advances in neural information processing systems*, vol. 6, pp. 737–744, 1993.

[26] G. Koch, R. Zemel, R. Salakhutdinov *et al.*, "Siamese neural networks for one-shot image recognition," in *ICML deep learning workshop*, vol. 2, 2015.

[27] D. David and A. NG, "Designing effective traditional and deep learning-based inspection systems," *Vision Systems Desugn*, vol. 26, no. 5, pp. 10–14, 2021.

[28] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

[29] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, p. 381–395, 1981. [Online]. Available: https://doi.org/10.1145/358669.358692

[30] LinkedAi, "Flip," https://github.com/LinkedAi/flip.

[31] F. Chollet *et al.*, "Keras," https://keras.io, 2015.