MaGNIFIES: Manageable GAN Image Augmentation Framework for Inspection of Electronic Systems

Pallabi Ghosh[1*], Gijung Lee[1], Mengdi Zhu[1], Olivia P. Dizon-Paradis[1], Ulbert J. Botero[2], Damon L. Woodard[1] and Domenic Forte[1*]

[1]Department of Electrical and Computer Engineering, University of Florida, 968 Center Dr, Gainesville, 32603, Florida, United States.
[2]MIT Lincoln Laboratory, 244 Wood St, Lexington, 02421, Massachusetts, United States.

*Corresponding author(s). E-mail(s): pallabighosh@ufl.edu; dforte@ece.ufl.edu;
Contributing authors: lee.gijung@ufl.edu; zhum@ufl.edu; paradiso@ufl.edu;
jbot2016@gmail.com ; dwoodard@ece.ufl.edu;

Abstract

Electronic counterfeiting is a long-lasting problem that continues to cost original manufacturers billions, fund organized crime, and jeopardize national security and mission-critical infrastructures. Manual inspection is a popular and standardized way to detect counterfeit electronic components, but, it is time-consuming and requires subject matter experts for classification. State-of-the-art machine learning, deep learning, and computer vision-based physical inspection methods are promising to alleviate these issues. However, the main bottleneck for doing so is a lack of high-quality, publicly available counterfeit image data for training. Producing such datasets is also time-consuming and often requires expensive equipment. In addition, most test labs are not allowed to freely publish images taken from their customer's chips. One solution to this data shortage bottleneck can be addressed by augmenting synthetic data. In this paper, (i) data multiplication using Progressive GAN, StyleGAN, and classical methods in counterfeit data domain are explored; (ii) a novel framework, named MaGNIFIES, is proposed; and (iii) an efficient Convolutional Neural Network architecture is proposed, that can detect defective parts by training only on the synthetic dataset generated using (i) or (ii). For proof of concept, we have used low-quality images of resistors and capacitors with and without scratch defects as counterfeit and golden components respectively. We have also illustrated how our approach using MaGNIFIES addresses the shortcomings of the existing augmentation methods. Separate data augmentation detection models are trained with each type of augmented data generated using MaGNIFIES, as well as existing techniques, and tested on a test set of real data.

Keywords: Counterfeit Data Augmentation, Generative Adversarial Network, Image Quality, defect detection, CNN, Machine Learning

1 Introduction

Counterfeit and pirated products create an enormous drain on the global economy. It costs billions of revenues in legitimate economic activity and facilitates a black market that deprives governments of revenues for vital public services, forces higher burdens on taxpayers, dislocates hundreds of thousands of legitimate jobs, and exposes consumers to dangerous and ineffective

products. Moreover, counterfeit electronics can be hazardous, especially when incorporated into safety critical systems such as aircraft, trains, and submarines [1]. The counterfeit chip market has an estimated worldwide value of $75 Billion, and such chips are integrated into electronic devices reportedly worth more than $169 Billion [2]. It is projected that net job losses due to counterfeiting in 2022 will be between 4.2 and 5.4 million [3]. In addition, counterfeits have huge reliability and security concerns. It is estimated that 15% of all spare and replacement electronic parts purchased by The Pentagon are counterfeit [4]. In 2017, a Presidential Executive order was passed to assess and strengthen the manufacturing and defense industrial base and supply chain resiliency of the United States. Moreover, the ongoing chip shortage due to the COVID-19 pandemic only further aggravates the situation by creating huge gaps in the supply chain. Counterfeiters are exploiting such gaps to fill supply chain needs with more and more counterfeit components [5]. Fortunately, counterfeits can often be identified by surface defects on component packages [6, 7] such as *scratches*, differences in indent size and position, sanding/grinding marks, ghost markings, burn markings, mold variation, package damage, corrosion/contamination, and extraneous markings. However, manual physical inspection is costly, time-consuming, and subjective. To pinpoint counterfeits, subject matter experts (SMEs) need to inspect each component for defects and require regular training to keep up with counterfeit detect trends. To address this, a few works from academia introduce automation, but validation is limited to small datasets. Notable works in this domain include [8–16], and almost all highlight that the lack of data is a major barrier. Moreover, computer vision and machine learning (especially deep learning) are becoming more popular tools to boost automation. This further increases the need for large datasets, as deep learning requires very large amounts of data on the scale of at least hundreds of thousands of samples for both golden and counterfeit classes. This paper explores different GAN-based and classical data augmentation methods and also proposes a novel data augmentation framework, named MaGNIFIES, to bridge the gap in automated counterfeit detection and machine learning. An overview of the framework is illustrated in Figure 1. Toward

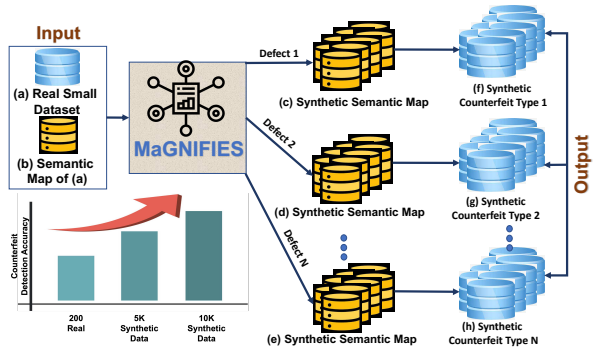


Fig. 1: Overview of MaGNIFIES

this end, we investigate Generative Adversarial Models (GANs) to generate arbitrary amounts of synthetic "golden" (known authentic) component images as well as defective counterfeit components. After training, such models can rapidly generate any number of synthetic component images without the need for expensive imaging equipment or the immense time to manually take images. Our framework is even capable of generating a variety of defects on counterfeit components, even when some type of original defective data is in scarcity. For example, it can generate components with defects such as scratches, marking imperfections, texture variation, etc. Additionally, such an approach can alleviate privacy concerns associated with test labs sharing counterfeit image datasets. Rather than directly sharing data, the GAN model trained by the data can be shared instead. In other words, derivatives of the original dataset can be generated by researchers or practitioners in the community while the original training dataset can be kept private. Afterward, these synthetic genuine models can be combined with other generative methods to create sets of component data superimposed with expected surface defects to create robust and effective counterfeit detection models in the face of data volume bottlenecks one would traditionally face in this space. This paper focuses mainly on scratch defects commonly found on counterfeit chips, but it can be extended to any other type of physical defect, such as corrosion, ghost markings, and others [6].This paper's contributions can be summarized as follows:

- Raises awareness of the data bottleneck in automated counterfeit detection algorithm development and proposes a novel solution to combat

the problem using different generative modeling (GAN) techniques and domain knowledge of counterfeits.
- Analyzes the pros and cons of both user controlled (i.e., Pix2pix) and uncontrolled (i.e., StyleGAN and ProGAN) GANs in augmenting data. Also, combines them into a domain-specific data augmentation framework named MaGNIFIES capable of generating a large, realistic variety of counterfeit component image datasets of various categories.
- Provides large datasets of resistors and capacitors, with and without defects, generated using both the controlled GANs and MaGNIFIES.
- Evaluates the synthetic image quality of data generated using GAN approaches vs. the classical data augmentation approaches. Discusses the pros and cons of each approach.
- Proposes a CNN architecture for the detection of defective resistors and capacitors and illustrates the considerable increase in accuracy over the classical machine learning approaches used in this domain.
- Offers evidence that the proposed framework can generate images beyond the data it was originally trained for (e.g., generate images for unseen chips and components).
- Provides the trained generative models and counterfeit chip dataset in the public domain (links to be added upon paper acceptance).

To the best of our knowledge in the hardware security and assurance domain, this is the first attempt to generate models that can generate unlimited realistic synthetic data to meet the large dataset requirement for deep learning models. Previously some work used data augmentation to address the bottleneck in Machine learning, but they were mostly classical data augmentation techniques capable of generating only a limited amount of data. In addition, the contributions here are not necessarily confined to the counterfeit detection domain. Research in other domains such as automated physical inspection for hardware Trojan detection, printed circuit board (PCB) bill of materials extraction, etc. also run into the same bottleneck. Thus, the promising results, tools, and datasets from this paper shall enable novel research in a variety of applications.

The remainder of this paper is organized as follows. A brief discussion on the available methods and datasets is provided in Section 2. The background on different tools and techniques used in this paper is described in Section 3. The data augmentation techniques and our proposed framework is explained in detail in Section 4 followed by the defect detection techniques in Section 5. Experimental setup and results are given in Section 6. Brief discussions on the pros and cons of the framework and future work are given in Section 7. Finally, the paper is concluded in Section 8.

2 Related Works

Some efforts have been made to manually create image datasets for integrated circuits (ICs) and PCBs. One of them is WACV [17], which has more than 8,000 annotated components from 47 images of 32 PCBs. FICS-PCB [18] has over 29,000 annotated components (about 6,000 of which are unique annotations) from 418 images of 31 PCBs and FPIC [19, 20] has over 40,000 semantically annotated components from 230 images of 73 PCBs. Although these datasets are relatively large, there are very few defects present in them. Also, even if there is one, the defects are not annotated. The available defect datasets are too small. Some of them include D-PCB [21], Amazon Lookout [22], and HRIPCB [23]. D-PCB consists of pairs of PCB images (one with synthetic defects and one without), but only offers 20 pairs. Amazon Lookout [22] consists of only 40 images of PCBs with defects and 40 images of PCBs without defects. HRIPCB [23] consists of about 3,000 annotated components; however, they are all from only 10 PCB images (each PCB has multiple defects). Another dataset that has a relatively decent amount of data is PCBexperiment [24] but it too lacks proper annotation of the defects and is classified based on manual quality assurance checks.

3 Background

3.1 Counterfeit IC Defects

Counterfeit ICs are often recycled, remarked, or cloned versions of authentic ICs. There are a list of defects that are commonly observed on the package indicating their fake nature. For example, the cloned version of an authentic IC may have the same lot number but a different country of origin, or a recycled IC may have some corrosion and/or scratches present on its surface due to wear and tear. A list of commonly observed package defects is tabulated in Table 1 [8]. Identification and localization of any of these defects will help in the identification of probable counterfeit chips.

3.2 Local Binary Pattern

Texture is a very useful feature that is widely used in solving many computer vision problems. An exact objective definition of "texture" is difficult, but subjectively texture is a property that gives information about the spatial arrangement of the color or intensities in an image. A popular technique used to define texture is Local Binary Pattern (LBP). For each pixel position, it compares the neighboring pixel values with its own value and generates a binary pattern. In this pattern, a value of '1' signifies that the neighboring pixel value is more than the pixel under observation, and '0' means it is less. Then the texel value, representing the pixel value of the texture map, is computed by converting the binary pattern to decimal. The entire texel value computation is illustrated using a 3X3 image in Figure 2. This technique is performed on all the pixels, across the image. Thus, using this map, the texture feature map is obtained. A sample texture feature map obtained from a capacitor is shown in Figure 2. Many state-of-the-art automated counterfeit detection approaches used LBP for classification of defects in smaller dataset and observed high accuracy [8–11, 13].

4 Data Augmentation

Data in the domain of security is not always readily available. Counterfeit components are one such security domain that suffers from data scarcity. Although the global supply chain is flooded with counterfeit components, incomplete documentation, due to the lack of a centralized forum to report the counterfeit components discovered, is a major reason for such scarcity. Another factor that has induced incomplete documentation is that companies tend to hide counterfeiting cases found in their name to protect their reputation from unfortunate publicity.

Table 1: Commonly observed package defects in counterfeit components and chips.

Package Defects		
Scratches	Extraneous Markings	Corrosion
Burned Markings	Marking Imperfection	Ghost Markings
Color Variation	Texture Variation	Invalid Lot/Date/Etc. Code
Indent Mismatch	Dirty Cavities	Package Damage

Automated counterfeit detection via machine learning is efficient but data-hungry. Insufficient data may lead to over-fitting of model, which is one of the major problems with using machine learning. In other words, the training will not be complete without a considerable amount of data leading to inaccurate detection. Data augmentation is one useful approach to overcome over-fitting from data scarcity. It approaches over-fitting from the root cause; that is, training data. It considers that raw data contains a load of information and can be expanded. Using data augmentation, new synthetic dataset can be created that follows the same distribution as the input data. It mainly preserves the features of the raw data, and only expands by varying the noise. In this paper, we have explored some of the existing classical data augmentation techniques as well as generative modeling-based data augmentation techniques. We have also proposed a new framework capable of over-coming the shortcomings of the existing techniques.

4.1 Classical Data Augmentation

A set of classical augmentation techniques, including blurring and noise, is used to generate new sets of augmented data. For blurring, two sets of data are generated with different sigma (s) values for Gaussian blur. Similarly for noise, two sets of augmented datasets are generated with different variance of random distribution for the Gaussian noise. The main constraint with classical augmentation is that image quality degrades with an increased number of output images. *With GAN, any number of similar images can be generated.*

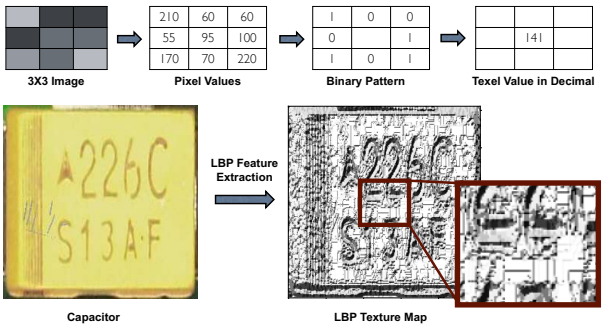


Fig. 2: Local Binary Pattern (LBP) feature map extraction

4.2 Generative Models

Generative modeling is a special branch of unsupervised learning where the model learns the regularities and patterns in the input data so that, once trained, it can generate new data points which plausibly could have been drawn from the original dataset [25]. It learns the true data distribution of the training set so that it can generate new data points with some variations. One such technique of generative modeling is the Generative Adversarial Network or GAN [26]. This type of training involves two networks – *generator* and *discriminator*. Generator models are unsupervised models that summarize the distribution of input variables used to create or generate new examples that plausibly belong to the input distribution. It takes a fixed-length random vector as input which is drawn randomly from a Gaussian distribution. The discriminator is the predictive model which discriminates or classifies the output of the generator as fake or real. The two models are trained together with the aim of achieving equilibrium. As the discriminator is updated to improve its ability to discriminate real and fake samples for the next round, the generator is likewise updated based on how well its generated samples fooled the discriminator. This technique uses a game theory approach to achieve Nash equilibrium between the two networks. After the training process, the discriminator model is discarded, and the generator model can be used to generate new, realistic data samples. There are a variety of GAN models available [27]. Out of all these models, two categories of the most advanced state-of-the-art models are described here: controlled GANs and uncontrolled GANs.

4.2.1 Uncontrolled GAN

The main purpose of using the uncontrolled GAN model is to multiply the input dataset. It generates synthetic data points following the same distribution as the training data. Two of the most advanced state-of-the-art uncontrolled GAN models are – Progressive GAN (ProGAN) [28] and StyleGAN [29]. ProGAN and StyleGAN produce high-quality, realistic images and offer superior control and understanding of the generated images. These qualities make it easier to generate believable fake images. For most GANs, generating realistic, high-resolution images is difficult because higher resolutions drastically amplify the gradient problem where the distributions do not have substantial overlap and the gradients point in random directions. Large resolutions also necessitate the use of smaller minibatches due to memory constraints, which further compromises training stability. However, ProGAN and StyleGAN address this gradient problem by training in an innovative way. Hence, they are among the first GANs that can generate high-resolution images: up to 1024 $\times$ 1024 resolution. Moreover, the models can adapt to any resolution and produce high-quality images.

4.2.2 Controlled GAN

Controlled GAN is a special type of generative model which learns pixel-level information like texture, color, etc. for each of the labels provided in *semantic maps* corresponding to the input images. Semantic Maps are image maps corresponding to a particular input image where each pixel of the input image is color coded in the semantic map and each color belongs to a particular label. It can then generate a realistic image using a user-created semantic map containing similar labels. The user has complete control in creating the semantic map of the output synthetic image. The only disadvantage is that to get an output image, the user needs to provide a semantic map as input. So, to create innumerable output images the user would need the same number of semantic maps. The controlled GAN model that we have used in our work is Pix2pix [30]. It learns the mapping between the semantic maps and the real images, provided as training input. The semantic maps contain separate colors for each label. Once trained, the model can generate

realistic synthetic images by using any provided semantic map with similar labels. The pros and cons of the controlled and uncontrolled GAN is illustrated in Table 2.

4.3 MaGNIFIES Data Augmentation Framework

The MaNGIFIES framework consists of an uncontrolled GAN in series with a controlled GAN. The reasoning behind this choice and the overall framework are discussed in this section.

4.3.1 Purpose of Controlled GAN

As discussed in Section 4, a controlled GAN mainly helps in creating a synthetic image based on a user-provided semantic map. The main purpose of using the controlled GAN model in the MaGNIFIES framework is to exploit this capability to boost the *variety* of synthetic golden (i.e., known authentic) and counterfeit images. Specifically, it can be used to create an assortment of counterfeit ICs based on domain knowledge of different types of package defects, commonly found in counterfeit ICs [8]. The user can create semantic maps with one or more package defects to create a synthetic counterfeit component image. Some examples of synthetic images of counterfeit components with defects, generated using our trained pix2pix model are shown in Figure 3. Sub-figures (a), (b), (c) and (d) shows resistors with different texture and color. These are examples of counterfeit components with package texture and color differences. Sub-figure (e) shows different styles of pins, which shows an example of pin defect. Also, we see different patterns of scratches in resistors (a), (b) and (e), as well as capacitor (h). Scratch marks on components are defects commonly found on recycled chip/component packages and fall under the scratch defect category. Also, random markings are generated in (f), (g) and (j) which are different from the original training data. This is an example of a marking defect. Figure 3(j) also shows the components with different colors, pins, and writings can even be created within the same image. This shows that components with multiple defects can also be generated using the same model even if it is not present in the training data. A proof-of-concept for creating structurally

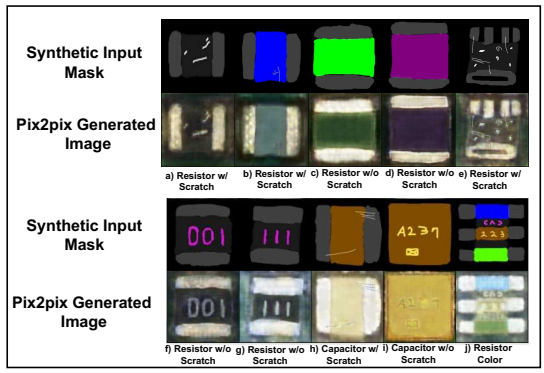


Fig. 3: An assortment of counterfeit data generated from semantic masks using Pix2pix.

dissimilar synthetic components and ICs with different marking and pin structures compared to the training set is shown in Figure 4. Here the model is trained on resistors and their semantic maps are similar to the images shown in Figures 4(a) and 4(b). But, when the trained model is provided with semantic map of a reference IC, as shown in Figure 4(d) and 4(c) respectively, it successfully generates the synthetic IC that looks similar to the original reference IC image, from which the semantic map is inspired, as shown in Figure 4(e). Note that these original reference ICs are not present in the training set. Nevertheless, since the textures of each of component in semantic map locations of the reference ICs are similar to that of the resistors in the training set, the pix2pix model was able to create reasonably good synthetic images from the semantic map provided by the user. Thus, without even using the original IC, MaGNIFIES was able to generate synthetic reference ICs. However, the main bottleneck of using the pix2pix model alone is that for each synthetic IC it needs one semantic map which the user needs to provide. In order to generate unlimited counterfeit IC images, one needs an unlimited source of semantic maps. Uncontrolled GANs can generate unlimited data following the same distribution as the input data. Hence, the generation of such masks is automated to some extent using an uncontrolled GAN.

4.3.2 Purpose of Uncontrolled GAN

The primary advantage of uncontrolled GAN is that it can multiply the input dataset. The only problem with uncontrolled GAN is that it cannot create an assortment of counterfeit components,

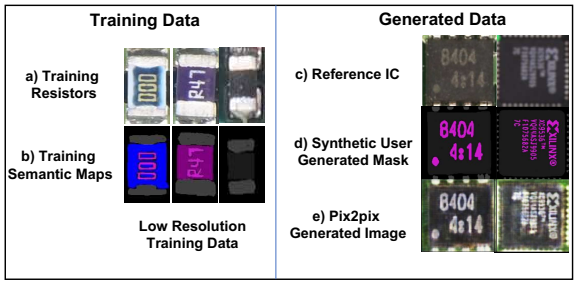


Fig. 4: Structurally different IC generated by training pix2pix with only resistors and capacitors. Reference IC not present in training set.

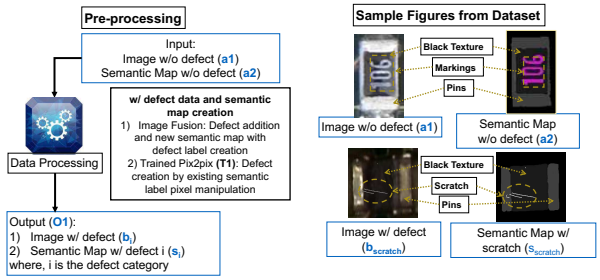


Fig. 5: Pre-processing of data: Defect insertion using image processing and/or controlled GAN

unlike the controlled GAN. But, if we have a small amount of assortment of counterfeit IC images, then uncontrolled GAN can be used to multiply the number of images. Hence, we have used an uncontrolled GAN in the MaGNIFIES framework as a *multiplier of semantic maps*. The uncontrolled GAN which we have integrated in our framework is StyleGAN.

4.3.3 Overall Framework

MaGNIFIES utilizes both controlled and uncontrolled GANs. The main goal of this framework is to generate unlimited counterfeit resistors and capacitors with one or more defects, given a set of golden ones and domain knowledge about the defect that needs to be inserted to generate the counterfeit component. The framework is a two-step method, involving pre-processing and two trainings asshown in Fig. 5, 6 and 7. The overall framework as shown in Figure 8. Each step is discussed in detail in the following sub-sections.

Step 1: This step involves pre-processing of the existing dataset and training the GAN models. In the pre-processing step, synthetic defects are added using two methods- image processing and/or controlled GAN. The pre-processing has two sub methods, semantic map creation and defect insertion based on domain knowledge.

- **Semantic Map Creation:** Semantic maps are generated for both the real and defective dataset, if available, using image processing. Since, this is the first work, no existing semantically labeled dataset was found. Each of the real data points are labeled manually. From these labeled datasets, the defective counterparts are generated using image processing. In future, this smaller labeled data can be used to automate the process of generating larger dataset with many defects and defect detection using uncontrolled GAN. This new semantic map dataset is one of the contributions of this work as well and plays a vital role in the automation of counterfeit resistors and capacitors. Although small in size, this can be considered as the first semantically labeled resistor and capacitor dataset.
- **Defect Addition:** As discussed in Section 3, there is a list of package defects that are commonly observed in counterfeit components. These defects can be either inserted in golden IC images using an image processing algorithm or using a trained controlled GAN. Image processing is mainly needed to add defects with new labels, i.e., the texture, color and/or pattern are not present in the input data and need some new label assignment. Controlled GAN is needed when existing labels are modified to create defects. For marking defects like letter mismatch, we can use the same semantic map label of the existing data and no new label is needed. However, if we want the counterfeit component to have markings with different color or scratch defect, which is not present in the real input dataset, we will need a new label for that, and the defect needs to be added externally on the component using image processing.
  The flow diagram along with some counterfeit components with scratches is shown in Figure 5. For the purpose of proof-of-concept and lack of original counterfeits, we have used only scratch defects, where random scratches are added to the data and a new defective dataset is generated. At the end of this step there is real golden dataset with two classes: defective and golden. Each of the images in this newly created dataset also has a semantic map.

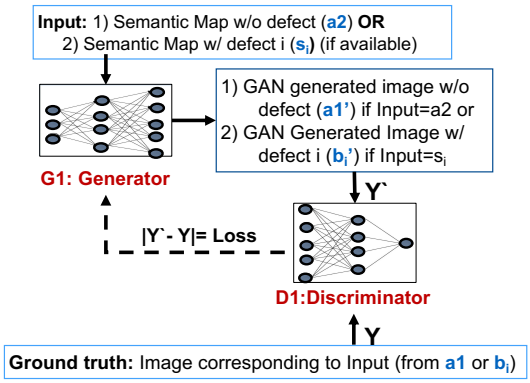


Fig. 6: Pix2pix Training (T1) (Controlled): Creation Using Controlled GAN

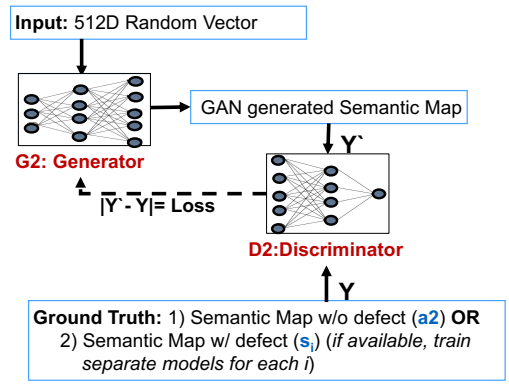


Fig. 7: StyleGAN Training (T2) Uncontrolled: Multiplication using uncontrolled GAN

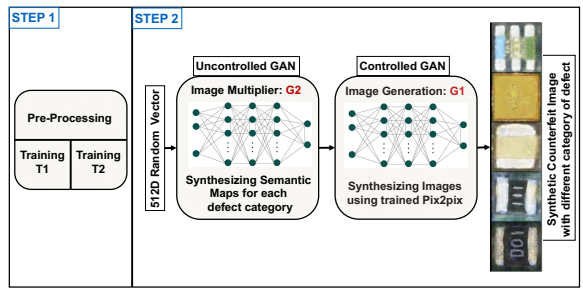


Fig. 8: MaGNIFIES Framework using controlled and uncontrolled GAN to generate ICs w/ and w/o defects commonly found in counterfeit components

- **Controlled GAN training:** Each class of the data with (w/) and without (w/o) defects along with their semantic maps are then used to train a separate controlled GAN architecture. This controlled GAN is then capable of generating more defective ICs as discussed in Sections 3 and 4.3.1 by providing the random semantic maps created by the user. In this paper, although we have used only the real, and scratch defect category for the next steps, it can be extended to any number of defect categories using trained controlled GANs in the future.
- **Uncontrolled GAN training:** Separate uncontrolled GAN models are trained with only the semantic maps of each of the defect and golden IC category. This uncontrolled GAN is used later as a semantic map multiplier as it is capable of generating any number of semantic maps similar to input semantic map distribution. The workflow of the uncontrolled GAN is shown in Figure 7.

Thus, we see defect insertion is taking place at two places in step 1. One at defect addition, where new types of defects are introduced and second by manipulating the input semantic to a controlled GAN. Uncontrolled GAN just multiplies the images by inserting randomnessin the semantic, for example creating randomness in the defect location, shape and size in the semantic. This semantic is then given asinput to the controlled GAN to generate a new data point.

Step 2: This step is essentially the multiplication part of the framework, combining the different modules obtained in step 1. From step 1, the trained controlled and uncontrolled GANs along with a classes w/ and w/o defect real IC dataset are obtained. This dataset is small, but each data point has a corresponding semantic map. This small dataset may have more categories of data. For N number of defect categories there can be N number of categories and one real data category. In this step, each of these categories is multiplied using an uncontrolled GAN trained in Step 1. Each of the trained models are used to generate unlimited semantic maps visually similar to the trained category. Since these semantics are segmentation maps, a little poor quality of the generated image would not create any problem, as long as all pixels belonging to the same category have the color as the training data. Once these synthetic semantic maps are generated using the uncontrolled GAN, they can be used in the trained controlled GAN model, also obtained from step 1, to generate more real and counterfeit component images. The entire MaGNIFIES workflow is given in Figure 8. An interesting fact about MaGNIFIES is that the generated data has both the image as

Table 2: Pros and Cons of the different data augmentation techniques in defective components generation.

Methods	Pros	Cons
Uncontrolled	Once trained can generate unlimited data	Can generate defective components similar to the one used in training
Controlled	Once trained can generate components with a variety of defects at chosen location	Number of data points is limited to the number of semantic maps provided
Classical Data Augmentation	Simple technique, no training data required	Degrades quality of the original image and can only generate limited amount of data
MaGNIFIES	Can generate unlimited data with variety of defects at different location	Complex architecture, combines both controlled and uncontrolled GAN

well as the semantic maps from which the data was generated. Hence, we plan to use this framework in the future, to generate a huge dataset which can be used to perform many segmentation tasks using deep network architectures and to detect defects. The pros and cons of each of the discussed data augmentation techniques is given in Table 2.

5 Defect Detection

To illustrate the power of data augmentation in defect detection and to find the most efficient data augmentation technique, different machine learning algorithms are trained with each of the synthetic datasets generated using different augmentation techniques, i.e., classical data augmentation, controlled, uncontrolled and MaGNIFIES, as discussed in Section 4, and tested on a set of real data. The different detection algorithms and their respective setup and parameters used are discussed in this section. These algorithms can be broadly classified under two categories. They are classical machine learning techniques and deep learning. The number of synthetic data points used for training is also varied to illustrate the power of data augmentation. A Convolutional Neural Network architecture is also proposed for defect detection in counterfeit components using deep learning.

5.1 Classical Machine Learning Techniques

The different classical machine learning algorithms used for defect detection and quality analysis can be grouped under three categories: linear, non-linear, and a basic neural network. The linear models used are Support Vector Machine (SVM) with linear kernel [31] and Logistic Regression [32]. The basic neural network approach used

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 199, 199, 32)	416
max_pooling2d (MaxPooling2D)	(None, 99, 99, 32)	0
conv2d_1 (Conv2D)	(None, 98, 98, 64)	8256
max_pooling2d_1 (MaxPooling2	(None, 49, 49, 64)	0
conv2d_2 (Conv2D)	(None, 48, 48, 128)	32896
max_pooling2d_2 (MaxPooling2	(None, 24, 24, 128)	0
flatten (Flatten)	(None, 73728)	0
dense (Dense)	(None, 256)	18874624
dropout (Dropout)	(None, 256)	0
dense_1 (Dense)	(None, 1)	257

Total params: 18,916,449
Trainable params: 18,916,449
Non-trainable params: 0

Fig. 9: Quantity Analysis: CNN architecture used for CNN

is Multi-layer Perceptron (MLP) [33] with 300 iterations. Local Binary Pattern (LBP), as discussed in Section 3, extracted from each of the images are used as input features representing the images, for each of these models.

5.2 Deep Learning

The main bottleneck of deep learning algorithms in detecting defects is necessity of large datasets. Even if the data is of very good quality, if the number of data points is not enough the model fails to learn all the features present in the data and leads to over-fitting. In such cases data augmentation plays an important role by spreading the data and increasing the amount of training data. The major utilization of the data augmentation techniques used in this paper are in the domain of deep learning. In this paper a simple and novel Convolutional Neural Network architecture is used for defect detection. The detail architecture of the model is given in the Figure 9. Each of the augmented sets of data, explained in Table 3, is used to train a CNN model, with batch size of 32 and 20 epochs. Each of the training set data is divided into 80% training data and 20% validation data. The training and validation curves for resistors and capacitors converges to an accuracy of 99.9% for both after 100 iterations. Once trained the models are tested using the same real test dataset which is used by the other machine learning methods and given in Table 4.

6 Experimental Setup and Results

6.1 Generative Adversarial Network Setup

In the following subsection, a brief introduction to the GAN architectures used is provided. The trade off in accuracy between using only unconditional GAN, to generate only one type of defective class, and our framework, capable of generating multiple class, is shown.

6.1.1 ProGAN and StyleGAN

In this paper, ProGAN and StyleGAN performances are first compared. ProGAN is an innovative way to train a GAN which involves training with low-resolution images, and then progressively increasing the resolution by adding layers to the networks. More details about this GAN model and the network parameters can be found in [28]. StyleGAN is a more advanced version of ProGAN, as its baseline configuration setup is the same, but its generator network has some modifications [29, 34]. More details of the architecture can be found in [29, 34]. As discussed in 3, StyleGAN and ProGAN are not capable of generating a variety of defects. Hence, we have combined StyleGAN with our framework and compared the trade off in accuracy.

Since the input capacitor and resistor images of the FICS-PCB Image Collection (FPIC) dataset [35] are low-resolution and vary in shape and size. Before training GANs, the images are resized to 256 $\times$ 256 for training. After training, both GANs generate synthetic capacitor and resistor images, when given a 512 dimensional vectors drawn randomly from a Gaussian distribution. The synthetic images have the same size as the training images: 256 $\times$ 256.

6.1.2 Pix2pix

The Pix2Pix GAN architecture involves the careful specification of a generator model, discriminator model, and model optimization procedure. Both the generator and discriminator models use standard Convolution-Batch Normalization-ReLU blocks of layers as is common for deep convolutional neural networks. The generator model is a modified U-Net model. Unlike the traditional GAN model that uses a deep convolutional neural network to classify images, the Pix2Pix model uses a PatchGAN as the discriminator. Details about the model and the architecture can be found in [30]. Unlike ProGAN and StyleGAN, a Pix2pix GAN is not capable of generating synthetic data without semantic. Hence it is used only in our framework.

6.2 Dataset

For training these models, we have used the resistor and capacitor images from FICS-PCB Image Collection (FPIC) dataset [18] with 424 capacitor images and 11,645 resistor images. For each set of augmented datasets, two classes of images are generated. They are 'with scratch' and 'without scratch'. Out of these images, 200 images are randomly selected from each of resistor and capacitor. The defective class of images with scratch is created from these images by superimposing scratches. Semantic maps are generated for each of these images, capacitor and resistors both, collectively referred as Real_Train set or (A) in this paper. Another set of 200 data points are randomly sampled from the remaining dataset. Scratches are again superimposed on them to create the 'scratch' class of images. This set of total 200 images for each class of with and without scratch, mutually exclusive of the Real_Train set, for each of capacitor and resistor is collected and named as Real_Test or (H) on which each of the models are tested. Parts of Real_Train set is used to train the uncontrolled GAN, as well as, to generate augmented images by adding blur and noise. The semantic labels for Real_Train are used to train the uncontrolled GAN of the MaGNIFIES framework and images of Real_Train along with the semantic maps are used to train the controlled GAN of the framework. After training the uncontrolled GANs (ProGAN and StyleGAN) with the Real_Train dataset, 10000 synthetic datapoints are generated using each of the models for each class. These new datasets are named as ProGAN10000 or (B) and StyleGAN10000 or (C). The labeled Real_Train dataset is also used to train different parts of the Magnifies Framework. Once trained, a set of 10000 datapoints are generated using the framework for each class and is referred as MaGNIFIES10000 or (D). Another set of 200 augmented data points are generated by

Table 3: Datasets used in Quality and Quantitative Analysis.

Datasets (resistor and capacitor each)	Size	Datasets (resistors and capacitor each)	Size
(A) Real_Train	200	(E) Classical Blur (s=1)	200
(B) ProGAN10000	10000	(F) Classical Blur (s=2)	200
(C) StyleGAN10000	10000	(G) Classical Noise (f=0.1)	200
(D) MaGNIFIES10000	10000	(H) Classical Noise (f=0.2)	200
(H) Real_Test	20		

adding blur to each class of 200 Real_Train dataset for each of Sigma values 1 and 2. They are referred as classical blur or (E) and (F). In detection methods, for increasing datapoints, (E) and (F) are combined and have commonly referred as Classical Blur. Similarly for noise two datasets, each of 200 datapoints for each class, are generated by adding random noise in the images using floating point values of 0.1 and 0.2, where the range of noise that can be applied is from zero to 1, 1 being the highest. These datasets are referred as (G) and (H) and in Detection methods they are mixed and referred as Classical Noise. All these datasets are also shown in Table 3. The final dataset for each type contains two subclasses - with and without scratch.

6.3 Data Augmentation and Quality Analysis of Generated Data

In this section, different no-reference quality metrics of the images generated using MaGNIFIES are compared with the quality of the real images as well as the synthetic image datasets given in Table 3. Each of the experiments is performed on capacitor and resistor datasets separately. No-reference image quality assessment (NR-IQA) is a class of IQA that is used to predict the quality of an image as perceived by human observers without using any pristine, reference images. GANs generate images from input data distributions and, hence, there are no particular data points to compare the images with. In such a situation, NR-IQA helps evaluate the quality of the output images without requiring any reference images. In this paper, the NR-IQA distributions of 100 randomly chosen images of each of the generated datasets are evaluated and compared. Among these no-reference image quality measures, there are two types of techniques. One of the techniques uses trained models which are trained on good and bad quality natural images and the score gives an idea about the image quality. The other technique is just to compare the different properties of the image like sharpness, brightness, blur, etc. For the technique with the trained model, we have used two methods Blind/Reference-less Image Spatial Quality Evaluator (BRISQUE) and Naturalness Image Quality Evaluator (NIQE), and for the technique which evaluates the property of an image, measurement of sharpness is used. Each of these measures has benefits and limitations, which are discussed in the following subsections.

6.3.1 Blind/Reference-less Image Spatial Quality Evaluator (BRISQUE)

Blind/Reference-less Image Spatial Quality Evaluator (BRISQUE) compares an image to a default model computed from images of natural scenes with similar distortions. Natural scene images refer to any images captured using a camera. The BRISQUE score is evaluated using a support vector regression (SVR) model trained on an image database with the corresponding differential mean opinion score (DMOS) values [36, 37]. The image database consists of several images along with mangled counterparts with known distortions such as compression artifacts, blurring, and noise. The image to be scored must have at least one of the distortions for which the model was trained. A smaller score indicates a higher image quality. The result obtained after evaluating the BRISQUE NR-IQA on 100 randomly chosen data points from each of the datasets created in this paper is shown in the box plots in Figure 10. It clearly illustrates how the addition of blur and noise degrades the quality of the images, whereas the quality of the images generated by the generative models is similar and sometimes better than the real images. Similar BRISQUE values would indicate that the generated synthetic images are similar to the real images, and the GAN is not causing unnatural distortions. Also, the better values indicate that the GAN models are able to remove noise and improve the quality of the generated images.

6.3.2 Naturalness Image Quality Evaluator (NIQE)

Naturalness Image Quality Evaluator (NIQE) also compares an image to a default model computed from images of natural scenes [38]. Similar to BRISQUE, a smaller score indicates a higher image quality. Here, the model tries to construct

a 'quality aware' collection of statistical features, based on a space-domain natural scene statistic (NSS) model. The primary difference between BRISQUE and NIQE is that NIQE is not trained on intentionally distorted images. Since NIQE features are derived from undistorted natural images, any type of measurable deviations from statistical regularities observed in natural images can be identified. This benefit comes with a trade-off, so NIQE's ability to find distortion is less accurate than BRISQUE's. Similar NIQE scores indicate similar NSS statistical deviation. The result obtained after evaluating the NIQE NR-IQA on 100 randomly chosen data points from each of the datasets created in this paper is shown in the box plots in Figure 10. As NIQE mainly identifies the statistical regularities (NSS) and performs less successfully than BRISQUE in identifying distortions, the range of NIQE values of all the datasets is almost similar. But when the mean value is observed, it is visible that the quality of the GAN-generated images is better.

6.3.3 Sharpness

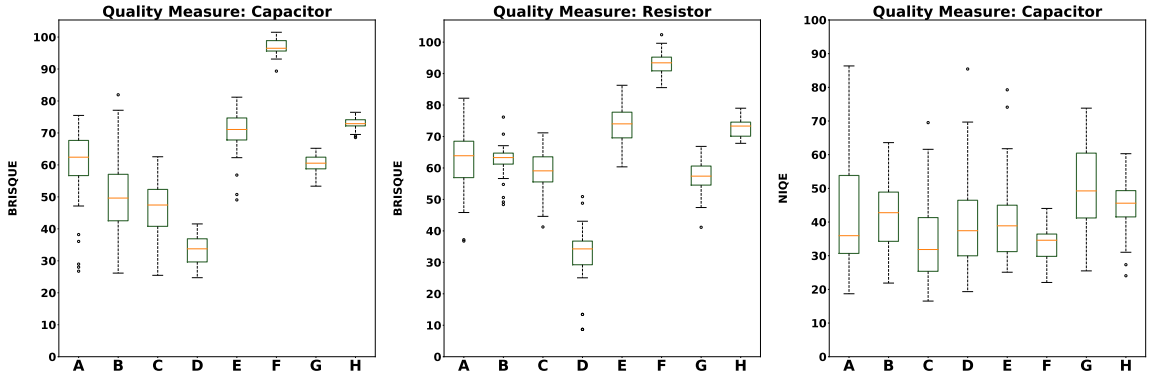
Image sharpness measures distortions by detecting blurriness, rather than training [39]. Although there are numerous ways to quantify sharpness, this paper uses a popular gradient-based approach [40]. In general, higher values indicate higher clarity. Very high sharpness values can indicate graininess or noise, which are other types of distortion. Ideally, the sharpness distribution of the augmented data is the same as that of the real data. The result obtained after quantifying the sharpness on 100 randomly chosen data points from each of the datasets created in this paper is shown in the box plots in Figure 10. It clearly illustrates how the addition of blur and noise degrades the quality of the images, whereas the quality of the images generated by the generative models is similar and sometimes better than the real images. Similar sharpness values would indicate that the generated synthetic images are similar to the real images, and the GAN is not causing unnatural distortions. In addition, the better values indicate that the GAN models are able to remove noise and improve the quality of the generated images.

6.4 Defect Detection and Quantitative Analysis

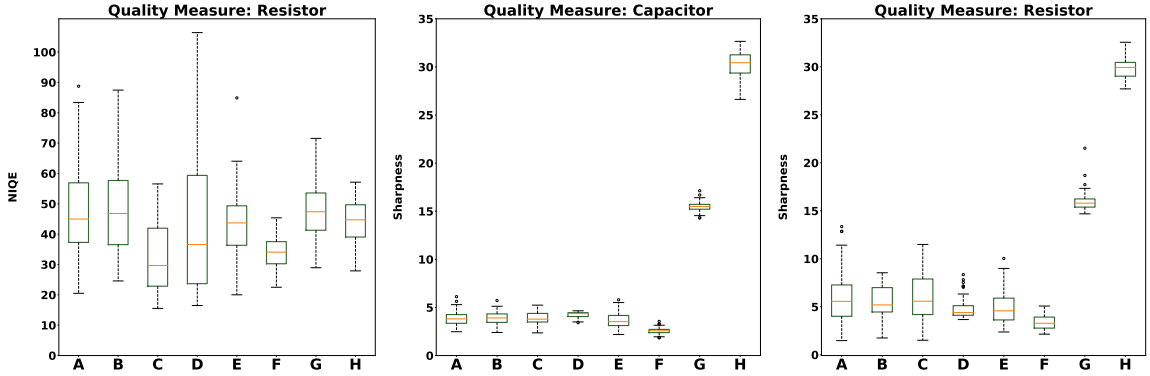
For quantitative analysis and to illustrate the utility of our proposed framework, MaGNIFIES, we have used different machine learning algorithms to illustrate how our framework, trained only on synthetic data, identifies defects in real data with significant accuracy. We have used both classical machine learning and deep learning methods for quality analysis. Experiments are performed on each and every dataset generated using each of the augmentation techniques to determine the most efficient data generation algorithm. The amount of datapoints used during training is also varied for each experiment for each dataset to illustrate how dataset size affects accuracy. Each set of experiments is performed on capacitor and resistor datasets separately. The total number of models trained with synthetic datasets generated using ProGAN, StyleGAN, and Magnifies is 16. The number of models trained with synthetic data generated by adding blur and noise is eight, because of a smaller number of generated datapoints. In addition to all these experiments, where each model is trained using synthetic data and tested on real data, Real_Test, one model of each type of detection algorithm is trained with Real_Train dataset and tested on the same Real_Test Dataset. This last set of 4 experiments, as shown in the last column of Table 4, is performed to illustrate how accuracy can be improved by augmenting data with respect to using only a small amount of real data. Each cell in the Table 4 represents one set of experiments where the model mentioned in the row-head is trained with the dataset mentioned in the column-head and is tested with the Real_Test set. The number of randomly chosen training samples is also mentioned in the same row under the training data size column. For the machine learning algorithms, number of random trials performed for each set of experiments is 10, where testing is carried out on the Real_Test dataset.

6.4.1 Observations

There is a clear trend of an increase in accuracy with the increase in data for both classical machine learning techniques and deep learning techniques. Overall, the accuracy of the deep learning methods is much higher than the classical



(a) BRISQUE image quality Capacitor images (b) BRISQUE image quality Resistor images (c) NIQE image quality Capacitor images



(d) NIQE image quality Resistor images (e) Sharpness Capacitor images (f) Sharpness Resistor images

Fig. 10: BRISQUE and Sharpness image quality comparison. The x-axis represents the datasets given in Table 3

Table 4: Quantitative Analysis Results for Capacitors(C) and Resistors(R): Each cell represents a model trained on the dataset mentioned in column-head and algorithm mentioned in row-head. Each of these models are tested using the Real_Test Capacitor data.

Col:Data Generation Technique Row: Detection Algorithm	Training Data Size for Each Class	ProGAN C/R	StyleGAN C/R	MaGNIFIES C/R	Blur C/R	Noise C/R	RealvsReal C/R
SGD Log Reg	**200**	69.09/67.27	67.49/66.36	65/66.06	66.13/66.02	62.9/62.81	68.18/65.90
	400	69.36/67.18	67.5/67.18	66.96/66.81	66.81/66.81	63.81/63.45	
	1000	69.54/67.72	68.18/67.50	67.45/67.42			
	10000	72.72/68.09	69.54/68.9	71.81/70.36			
SVM Linear	**200**	69.54/69.5	62.95/68.63	59.39/69.54	62.72/69.31	60.18/65.45	69.54/69.54
	400	69.09/67.27	63.45/68.86	66.81/69.84	62.93/69.65	59.54/65.72	
	1000	72.72/70.00	69.77/69.31	67.09/70.38			
	10000	73.99/70.27	72.72/72.72	71.36/71.18			
MLP	**200**	60.9/60.00	61.36/59.77	60.6/58.48	61.13/58.29	59.27/57.09	61.36/63.63
	400	60.45/62.72	59.31/61.81	60.75/59.99	61.93/60.11	62.36/57.81	
	1000	61.99/59.54	61.81/62.27	61.06/59.39			
	10000	65.45/64.54	62.04/65.45	61.33/60.15			
CNN	**200**	99/96.24	96.49/93.25	83.25/89.75	79.25/51.25	68.48/61	93.75/97
	400	99.75/99.50	99.25/98.5	99.50/93.75	95.74/94.49	69.99/85.25	
	1000	100/99.5	99.75/99.00	99.75/95.99			
	10000	100/99.75	100/99.5	99.00/98.5			

methods. Hence, it is evident if the data bottleneck for deep learning methods is solved, then much better accuracy can be obtained. Another interesting fact observed in the result is that, in some cases, the model trained on synthetic data outperforms the model trained on real data. One reason for this is the real data is of very low resolution and extremely noisy. Noise affects the quality of the images to a great extent as we have seen in quality analysis. The GAN-generated images are relatively less noisy. The quality of the images generated by all three generative models is similar to or better than the real images in terms of noise. This may have resulted in increasing the accuracy to some extent. Also, in most cases, the number of data points used to train the models is more than the model trained with only real data, which has also resulted in better accuracy.

7 Discussion

This paper illustrates a preliminary synthetic counterfeit component data generation method using different generative models as well as the proposed MaGNIFIES framework. The paper also proves how data generated by each of the proposed techniques, i.e., StyleGAN, ProGAN, and MaGNIFIES generally improves the accuracy of scratch detection in components. Not only in terms of volume, but our proposed framework MaGNIFIES can also generate different categories of counterfeit data by incorporating domain knowledge of counterfeit components. Quantity and quality analyses of the synthetic data also show that it is as good as the real data, sometimes performing better because of less noise and/or more quantity and helps in increasing the accuracy of counterfeit detection. There is room for improvement and expansion in future work. In this paper, because of the lack of high-resolution real data, the synthetic images generated by both our framework as well as the StyleGAN and ProGAN are of equally low resolution. To generate high-quality images, the framework would need some number of high-resolution images. The dataset in this paper lacks pin and lead data to properly capture ICs which can be added in the future. Many other counterfeit data types and defects will also be created. In this work we have used data which has very less variation, i.e, addressed only one category of capacitor or resistor, and addressed only one category of defect. To address all categories of data with all the defects, we would require a larger dataset with sample subset of each category and defect. Also, since each of the output images has a semantic map associated with it, the paired huge dataset can be used in advanced computer vision and image analysis algorithms, such as automated segmentation, that require such semantic maps for training. Finally, our promising results coupled with the sheer quantity of synthetic images that the generative models can produce shall enable deep-learning based classification in future work, *which cannot currently be employed due to the scarcity of real data.*

8 Conclusion

Lack of data is a long-standing bottleneck in the automated detection of counterfeit IC using machine learning and deep learning techniques. Manually creating datasets by taking and labeling images is time-consuming, requires expensive image collection setups, and requires one to buy vast quantities of electronic chip/component samples. Also, such counterfeit electronics are not always readily available. In this paper, to alleviate such problems of data acquisition, different data augmentation techniques are explored. The cons of each of the methods are then addressed using our proposed framework, that uses generative models along with domain knowledge of the counterfeits to generate synthetic chip/component images having properties similar to the actual authentic and counterfeit chips/components. As a proof-of-concept, a dataset of 10,000 images each of capacitors and resistors with and without scratch defects was generated using our framework. The quality analysis of the generated synthetic images and the real images showed that the generated images are as good as the real ones. To prove the utility of uncontrolled GANs and the MaGNIFIES framework, quantitative analysis is also performed using different machine learning algorithms. The results clearly show how an increase in data using GANs increases detection accuracy. In the future, we plan to use higher-resolution real images to illustrate the generation of other types of counterfeit defects using this framework.

9 Declaration

9.1 Ethical Approval

Not Applicable

9.2 Competing interests

Not Applicable

9.3 Authors' contributions

Dr. Domenic Forte and Dr. Damon L. Woodard provided supervision throughout the work. Data labeling done by Pallabi Ghosh and Gijung Lee. Coding and data generation done by Pallabi Ghosh, Gijung Lee and Mengdi Zhu. Pallabi Ghosh wrote the manuscript. Olivia P. Dizon-Paradis provided supervision throughout the writing process and contributed text in the literature reviews to the paper. All authors reviewed the paper.

9.4 Funding

The work was supported by National Science Foundation (NSF) Awards – 1821780 and 2131480.

9.5 Availability of data and materials

Dataset and Code will be made available in Github after acceptance of the paper.

References

[1] Dhwani Mehta, Hangwei Lu, Olivia P Paradis, Mukhil Azhagan MS, M Tanjidur Rahman, Yousef Iskander, Praveen Chawla, Damon L Woodard, Mark Tehranipoor, and Navid Asadizanjani. The big hack explained: detection and prevention of pcb supply chain implants. *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, 16(4):1–25, 2020.

[2] Paul Karazuba. Combating counterfeit chips. *https://semiengineering.com/combating-counterfeit-chips/*, 2020.

[3] Frontier economics. The economic impacts of counterfeiting and piracy. In *Report prepared for BASCAP and INTA*. International Chamber of Commerce, 2016.

[4] Brett Daniel. Counterfeit electronic parts: A multibillion-dollar black market. *https://www.trentonsystems.com/*, 2020.

[5] Bill Cardoso. The dark side of the chip shortage: Counterfeits. *X-ray News*, 2021.

[6] Ujjwal Guin, Daniel DiMase, and Mohammad Tehranipoor. Counterfeit integrated circuits: Detection, avoidance, and the challenges ahead. *Journal of Electronic Testing*, 30(1):9–23, 2014.

[7] Ujjwal Guin, Domenic Forte, and Mohammad Tehranipoor. Anti-counterfeit techniques: From design to resign. In *2013 14th International workshop on microprocessor test and verification*, pages 89–94. IEEE, 2013.

[8] Pallabi Ghosh and Rajat Subhra Chakraborty. Recycled and remarked counterfeit integrated circuit detection by image-processing-based package texture and indent analysis. *IEEE Transactions on Industrial Informatics*, 15(4):1966–1974, 2019.

[9] Pallabi Ghosh and Rajat Subhra Chakraborty. Counterfeit ic detection by image texture analysis. In *2017 Euromicro Conference on Digital System Design (DSD)*, pages 283–286, 2017.

[10] Pallabi Ghosh, Aritra Bhattacharya, Domenic Forte, and Rajat Subhra Chakraborty. Automated defective pin detection for recycled microelectronics identification. *Journal of Hardware and Systems Security*, 3(3):250–260, 2019.

[11] Pallabi Ghosh, Ulbert J Botero, Fatemeh Ganji, Damon Woodard, Rajat Subhra Chakraborty, and Domenic Forte. Automated detection and localization of counterfeit chip defects by texture analysis in

infrared (ir) domain. In *2020 IEEE Physical Assurance and Inspection of Electronics (PAINE)*, pages 1–6. IEEE, 2020.

[12] Pallabi Ghosh, Domenic Forte, Damon L Woodard, and Rajat Subhra Chakraborty. Automated detection of pin defects on counterfeit microelectronics. In *ISTFA 2018: Proceedings from the 44th International Symposium for Testing and Failure Analysis*, page 57. ASM International, 2018.

[13] Pallabi Ghosh, Fatemeh Ganji, Domenic Forte, Damon L Woodard, and Rajat Subhra Chakraborty. Automated framework for unsupervised counterfeit integrated circuit detection by physical inspection, 2019.

[14] Navid Asadizanjani, Mark Tehranipoor, and Domenic Forte. Counterfeit electronics detection using image processing and machine learning. In *Journal of physics: conference series*, volume 787, page 012023. IOP Publishing, 2017.

[15] Kaleel Mahmood, Pedro Latorre Carmona, Sina Shahbazmohamadi, Filiberto Pla, and Bahram Javidi. Real-time automated counterfeit integrated circuit detection using x-ray microscopy. *Applied Optics*, 54(13):D25–D32, 2015.

[16] Sina Shahbazmohamadi, Domenic Forte, and Mark Tehranipoor. Advanced physical inspection methods for counterfeit ic detection. In *ISTFA 2014: Conference Proceedings from the 40th International Symposium for Testing and Failure Analysis*, page 55. ASM International, 2014.

[17] Chia-Wen Kuo, Jacob D Ashmore, David Huggins, and Zsolt Kira. Data-efficient graph embedding learning for pcb component detection. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 551–560. IEEE, 2019.

[18] Hangwei Lu, Dhwani Mehta, Olivia P Paradis, Navid Asadizanjani, Mark Mohammad Tehranipoor, and Damon L Woodard. Ficspcb: A multi-modal image dataset for automated printed circuit board visual inspection. *IACR Cryptol. ePrint Arch.*, 2020:366, 2020.

[19] Nathan Jessurun, Olivia P Dizon-Paradis, Jacob Harrison, Shajib Ghosh, Mark M Tehranipoor, Damon L Woodard, and Navid Asadizanjani. Fpic: A novel semantic dataset for optical pcb assurance. *arXiv preprint arXiv:2202.08414*, 2022.

[20] Nathan Jessurun, Daniel E. Capecci, Olivia P. Dizon Paradis, Damon L. Woodard, and Navid Asadizanjani. Semi-Supervised Semantic Annotator (S3A): Toward Efficient Semantic Labeling. In Meghann Agarwal, Chris Calloway, Dillon Niederhut, and David Shupe, editors, *Proceedings of the 21st Python in Science Conference*, pages 7 – 12, 2022.

[21] Yehonatan Fridman, Matan Rusanovsky, and Gal Oren. Changechip: A reference-based unsupervised change detection for pcb defect detection. In *2021 IEEE Physical Assurance and Inspection of Electronics (PAINE)*, pages 1–8. IEEE, 2021.

[22] Prashanth Ganapathy and Amit Gupta. Defect detection and classification in manufacturing using Amazon Lookout for Vision and Amazon Rekognition Custom Labels | AWS Machine Learning Blog, July 2021. Section: Amazon Lookout for Vision.

[23] Weibo Huang and Peng Wei. A pcb dataset for defects detection and classification. *arXiv preprint arXiv:1901.08204*, 2019.

[24] Namratha Karanth. PCBexperiment, 2022.

[25] Yuichiro Anzai. *Pattern recognition and machine learning*. Elsevier, 2012.

[26] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

[27] Avinash Hindupur. The gan zoo, 2018.

[28] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans

for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017.

[29] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8110–8119, 2020.

[30] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017.

[31] William S Noble. What is a support vector machine? *Nature biotechnology*, 24(12):1565–1567, 2006.

[32] Raymond E Wright. *Logistic regression.*, pages 217—-244. American Psychological Association, 1995.

[33] Hassan Ramchoun, Youssef Ghanou, Mohamed Ettaouil, and Mohammed Amine Janati Idrissi. Multilayer perceptron: Architecture optimization and training. *International Journal of Interactive Multimedia and Artificial Intelligence*, 2016.

[34] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. *CoRR*, abs/1812.04948, 2018.

[35] Mukhil Azhagan Mallaiyan Sathiaseelan, Olivia P Paradis, Dhwani Mehta, Hangwei Lu, Sudarshan Agrawal, Alexandra Roberts, Nathan Jessurun, Damon L Woodard, Praveen Chawla, Mark Tehranipoor, and Navid Asadi. Pcb images. *TrustHub,https://trusthub.org/#/data/PCB-Images*, 2016.

[36] Anish Mittal, Anush K Moorthy, and Alan C Bovik. Blind/referenceless image spatial quality evaluator. In *2011 conference record of the forty fifth asilomar conference on signals, systems and computers (ASILOMAR)*, pages 723–727. IEEE, 2011.

[37] Anish Mittal, Anush Krishna Moorthy, and Alan Conrad Bovik. No-reference image quality assessment in the spatial domain. *IEEE Transactions on image processing*, 21(12):4695–4708, 2012.

[38] Anish Mittal, Rajiv Soundararajan, and Alan C Bovik. Making a "completely blind" image quality analyzer. *IEEE Signal processing letters*, 20(3):209–212, 2012.

[39] Rony Ferzli and Lina J Karam. A no-reference objective image sharpness metric based on the notion of just noticeable blur (jnb). *IEEE transactions on image processing*, 18(4):717–728, 2009.

[40] Tolga Birdal. Sharpness estimation from image gradients. *MATLAB Central File Exchange*, 2021.