



Article

PCB Component Detection Using Computer Vision for Hardware Assurance

Wenwei Zhao *, Suprith Reddy Gurudu, Shayan Taheri *, Shajib Ghosh, Mukhil Azhagan Mallaiyan Sathiaseelan and Navid Asadizanjani

Electrical and Computer Engineering Department, University of Florida, Gainesville, FL 32611, USA; sgurudu@ufl.edu (S.R.G.); shajib.ghosh@ufl.edu (S.G.); mukhil.mallaiyan@ufl.edu (M.A.M.S.); nasadi@ece.ufl.edu (N.A.)

* Correspondence: wenwei.zhao@ufl.edu (W.Z.); shayan.taheri@ufl.edu (S.T.)

Abstract: Printed circuit board (PCB) assurance in the optical domain is a crucial field of study. Though there are many existing PCB assurance methods using image processing, computer vision (CV), and machine learning (ML), the PCB field is complex and increasingly evolving, so new techniques are required to overcome the emerging problems. Existing ML-based methods outperform traditional CV methods; however, they often require more data, have low explainability, and can be difficult to adapt when a new technology arises. To overcome these challenges, CV methods can be used in tandem with ML methods. In particular, human-interpretable CV algorithms such as those that extract color, shape, and texture features increase PCB assurance explainability. This allows for incorporation of prior knowledge, which effectively reduces the number of trainable ML parameters and, thus, the amount of data needed to achieve high accuracy when training or retraining an ML model. Hence, this study explores the benefits and limitations of a variety of common computer vision-based features for the task of PCB component detection. The study results indicate that color features demonstrate promising performance for PCB component detection. The purpose of this paper is to facilitate collaboration between the hardware assurance, computer vision, and machine learning communities.

Keywords: artificial intelligence; automated optical inspection; computer vision; feature extraction; feature selection and analysis; PCB component detection; physical inspection and assurance



Citation: Zhao, W.; Gurudu, S.R.; Taheri, S.; Ghosh, S.; Mallaiyan Sathiaseelan, M.A.; Asadizanjani, N. PCB Component Detection Using Computer Vision for Hardware Assurance. *Big Data Cogn. Comput.* **2022**, *6*, 39. <https://doi.org/10.3390/bdcc6020039>

Academic Editors: Antonio Fernández-Caballero, Byung-Gyu Kim and Hugo Pedro Proença

Received: 15 February 2022

Accepted: 30 March 2022

Published: 8 April 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Modern electronic systems ranging from personal computers and mobile devices to critical government, military, and medical infrastructures use printed circuit boards (PCB) as functional blocks that connect different electrical components, traces, and vias [1]. With the advancement of semiconductor industries, the design of these PCBs is getting highly complex, having multiple layers with hidden vias and embedded passive components to meet the requirements of advanced systems. Incorporating such complexities creates a great opportunity for the potential attackers to maliciously modify the design and thus expose the PCB supply chain to vulnerabilities [2]. Moreover, due to the dominant trend of outsourcing the PCB manufacturing process, a wide variety of vulnerabilities such as tampering, recycling, cloning, etc., are being introduced to the hardware assurance community now more than ever. Therefore, from a hardware assurance perspective, it is of paramount significance to involve on-site verification under trusted conditions.

Over the years, physical inspection in the optical domain has become a popular approach within the community due to its mostly non-contact and non-destructive nature [3]. Traditionally, a subject matter expert (SME) performs visual inspections of PCBs under certain controlled conditions. They analyze PCBs not only for defects, but also for maliciously added components and Trojans. However, this process is time-consuming and error-prone with increasing number of PCBs. Therefore, researchers in the community

have proposed different methods of image processing, computer vision, and machine learning for automatic visual inspection. However, many of these approaches rely on the existence of golden PCB, which is not always available in practice.

Application of an automatic Bill of Materials extraction method to tackle this problem was introduced in [3]. The proposed framework of materials extraction for PCB assurance (as shown in Figure 1) involves two major steps: (1) imaging modality, and (2) image analysis. In this paper, we focus on the image analysis step where the goal is to detect and identify the components in a PCB with high accuracy.

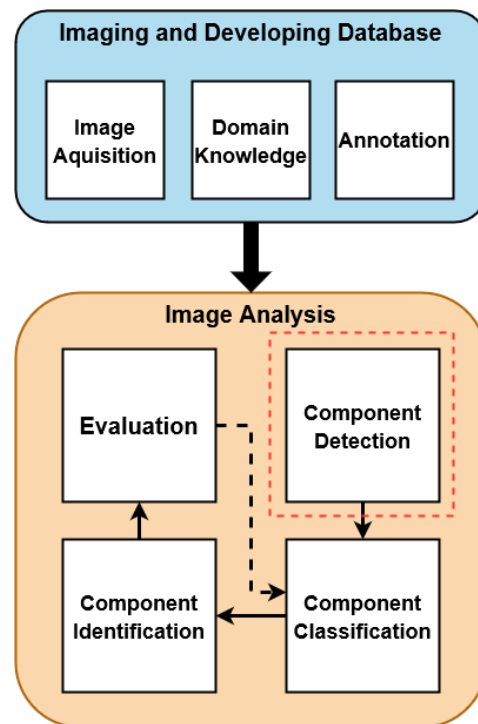


Figure 1. The framework for bill of materials extraction for PCB assurance as proposed in [3].

While developing an efficient system for detecting components automatically, it should be kept in mind that the method needs to be fast and highly accurate to meet the requirements of critical applications (e.g., military and biomedical applications) [3]. In addition, there are a wide variety of factors and challenges which needs special attention while developing an automated PCB assurance system. Though the majority of PCBs are monochromatic and consist mostly of standard commercial off-the-shelf components, there are an increasing variety of new and/or custom components as the state of technology is constantly advancing. Additionally, in the case of foreign, competitor, and/or malicious technologies, the components may be intentionally obfuscated with uncommon designs, camouflaging, and misleading or absent silkscreen labels. Moreover, factors related to image acquisition systems can also significantly affect the performance of developed component detection and identification algorithm [3]. Unfortunately, such challenges are very difficult to overcome with existing traditional computer vision (CV) and image understanding methods for component detection and identification, alone [4].

Though different machine learning (ML) and deep learning (DL) based approaches have shown significant progress in case of object detection and localization in other domains [5], it has not shown much progress in the PCB assurance domain. Though ML and DL based methods may outperform the traditional CV methods in terms of performance, there are a variety of challenges in the PCB domain that must be overcome. For example, ML and DL based methods tend to require massive amount of labeled data, which can be expensive, tedious, and time-consuming to obtain. In addition, such methods can be difficult to interpret and adapt with the technological progresses. To overcome these

challenges, we propose the benefits of CV methods which can be used in tandem with ML and DL methods.

This article provides an overview on the development of a novel system in the area of physical assurance for the problem of PCB component detection based on applying optimized computer vision techniques using semantic data. It is considered as one of the few works in the literature that introduces the concepts and the methods from computer vision into the domain of hardware security, specifically for PCB assurance. The utilized data is nicely different from the related works since the samples are in semantic format in here, while they use the bounding box information. The semantic attributes can positively impact classification of the components, tracking the pins, and extraction of the netlist. Considering a computing system with the hierarchy of (a) application and data; (b) computational flow, framework, system architecture; (c) system entities; and (d) computations of entities, then the most significant novelty of the proposing system in this work is at the levels of “a” and “b”. The review studies various techniques in the major computing steps/units within a conventional PCB component detection system that are: data collection, feature extraction, feature selection, and feature classification. So, our contribution at lower levels (“c” and “d”) is described as implementation of different computer vision methods within the context of PCB component detection as well as their further optimization using relevant machine learning procedures. Explicitly, the review includes a comprehensive introduction, technical evaluation, and conclusive discussion on three feature types of color, shape, and texture for this application. Such features enhance the explainable property of PCB assurance system and allow for incorporation of PCB domain knowledge, which leads to constriction of learning model, reduction in the required amount of data for training and testing, and finally achieving highly accurate results. The researchers and engineers from academia, industry, and government can find this review paper very useful due to: (1) better understanding the problems, challenges, and vacancies for this application in hardware security; and (2) realizing opportunities to develop interesting ideas and initiate new research directions on how to benevolently and maliciously leverage the processes from computer vision and artificial intelligence. Therefore, this work serves as a great structure to facilitate collaboration between hardware security, computer vision, and artificial intelligence research communities.

The remainder of the paper is structured as follows. In Section 2, we present related works and relevant existing techniques. In Section 3, we explain our methodology of analyzing different features using image analysis for PCB component detection. Sections 4–6 contain detailed discussion on the benefits, limitations, and uses of color, shape and texture feature descriptors in hardware assurance, respectively. Section 7 shows quantitative results and analysis of the features’ importance for the task of PCB component detection. Finally, in Section 8, we provide an overall discussion and future research directions for our work followed by the concluding remarks in Section 9.

2. Related Works

In comparison to previous efforts in creating effective feature extraction methods, there has been an increased focus in deploying feature selection approaches in object identification tasks [6]. Despite the fact that numerous studies have been done to investigate the relevance of feature extraction and feature selection in machine learning-based objection detection tasks, there are few studies on applying them in the PCB assurance domain. Feature saliency, which is dependent on the size of the dataset and the difficulty of the task, determines the effectiveness of a machine learning model for object identification, segmentation, and so on. As a result, feature selection techniques are incredibly important.

According to [7], a feature can be classified as (i) very relevant, (ii) somewhat relevant but not redundant, (iii) irrelevant, or (iv) redundant. While analyzing the reasons for Convolutional Neural Networks (CNNs) remarkable performance on complicated perceptual tasks such as object recognition, it was discovered that image texture features are more significant than object forms [8]. In [9], it was demonstrated that when color

features are mixed with traditional form features, state-of-the-art results for object detection could be obtained. Due to the fact that a range of external elements can considerably impair the efficacy of a component detection method for PCB images, a variety of image attributes (e.g., color, shape, and texture) should be considered [3]. The color normalization technique [3] will help in employing an appropriate feature selection strategy for PCB component detection. In [10], a hybrid strategy for extracting color and shape features for object detection was proposed. In contrast, recent research indicates that, in addition to color and shape, texture plays a critical role in detecting objects in images [11]. As demonstrated, there are a variety of examples where CV-based features can benefit ML and DL models.

A deep learning model will perform well if sufficient data is available to train on. In the case of a limited training dataset, a deep neural network may overfit and be unable to generalize. Since a DNN has millions of parameters, each with complicated inter-relationships, manually tuning the model's parameters would be incredibly challenging and computationally expensive. However, in certain cases, a similar performance could be obtained by simpler methods such as basic color thresholding, which uses small amount of data. Certain issues can be solved using less sophisticated, and time-consuming traditional computer vision-based solutions [12]. Conventional CV-based approaches are completely transparent, but deep learning models are often criticized for being opaque and difficult to understand. This is even more challenging in the field of PCB assurance due to the variety of components, continuous changes in technology, so giving rise to more added features to be considered. As DL models often suffer from a critical phenomenon known as the curse of dimensionality, dimensionality reduction is imperative to perform to limit the memory storage requirements and the computational overhead associated with data analysis. Two main components of dimensionality reduction, such as feature extraction and feature selection, offer improved learning performance, increased computing efficiency, decreased memory storage, and the development of more robust generalization models [13]. Therefore, in our work, we have evaluated the importance of extracted features and have proposed methods of selecting the optimal set of features strictly relevant to PCB component detection for hardware assurance.

3. Methodology

To prepare the data for PCB detection, we first have collected PCB images and performed color correction and region windowing. Later, color, shape, and texture features have been extracted from each windowed region. Finally, feature selection have been used to determine the importance of each feature. Importance values can be used to reduce the number of features to only the most salient ones, which can be used in tandem with ML or DL models to improve their efficiency and generalization. The process of our workflow is summarized in Figure 2. We introduce the process in this section.

3.1. Data

In this study, we used 15 images of 10 different PCB samples, of which 5 PCBs possessed components on both the front side and the backside, so both sides were imaged. The PCBs were purchased online or disassembled from a variety of different devices such as servers, computer hard drive controllers, and audio amplifiers. The images were taken under ambient laboratory light using a Nikon D850 digital single-lens reflex (DSLR) camera. This camera was set to take a 2 s delay and then a 2 s exposure shot to reduce noise during image collection due to camera shake. After, color correction was used during preprocessing to effectively normalize the data in a controlled manner [3]. The dataset used here is derived from a previous study in [3].

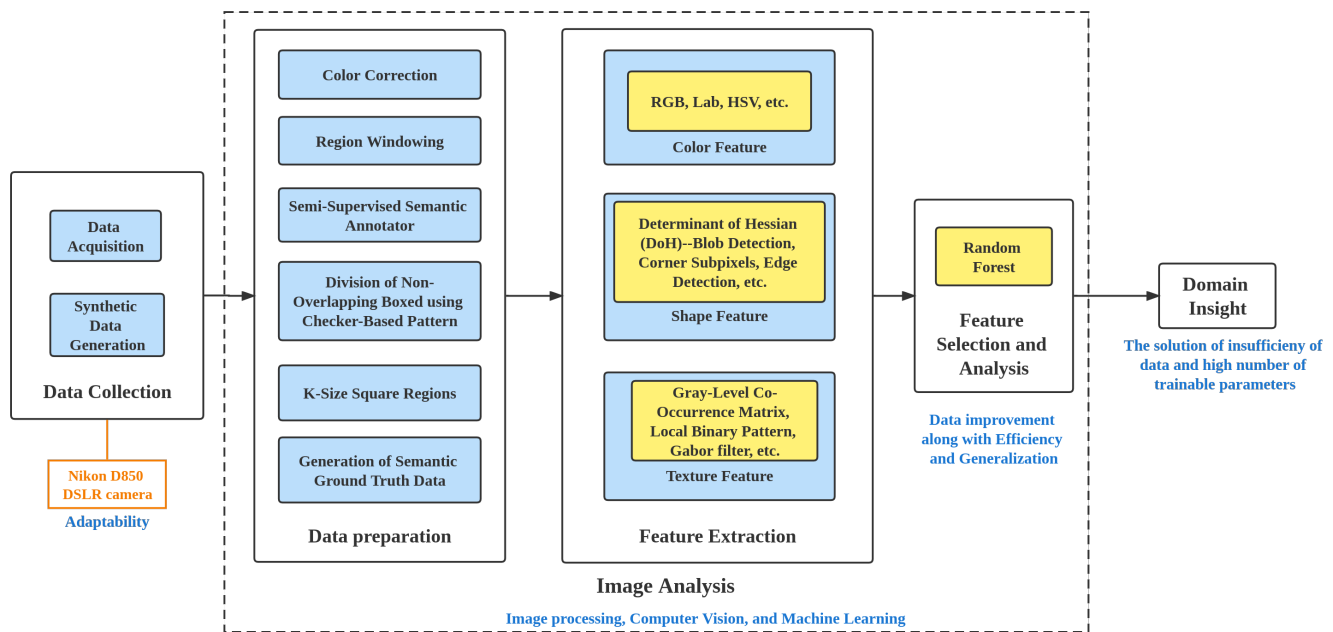


Figure 2. The processing workflow.

Then, PCB components (e.g., resistors, capacitors, and integrated circuits (ICs)) were annotated with semantic boundaries by human SMEs. In this study, the semi-supervised semantic annotator (S3A) have been used for annotation. By dividing each image into non-overlapping boxes in a checker-board pattern, the image data have been divided into square regions of different kernel sizes (ksizes). Since the ideal ksize is unknown, we have performed feature extraction on each divided region for ksizes of 5, 10, 15, 20, and 25 pixels.

Since this study is concerned with PCB component detection, ground truth data have been generated as follows. As feature extraction have been performed on each of the non-overlapping box regions (bboxes) in the image, the ground truth needed to be in terms of the bboxes. Therefore, the bbox ground truth have been generated based on the semantic ground truth data, which consist of pixel-level labels of 1s and 0s for component and background, respectively. Each bbox ground truth region have been assigned values of 0–10 for the percentage of the region’s pixels that correspond to a component (e.g., a region with a value of 0 does not have any component pixels in the semantic ground truth, whereas a region with a value of 10 completely consist of component pixels in the semantic ground truth). A visual example of the bbox label mask and semantic ground truth can be seen in Figure 3.

3.2. Feature Extraction

Feature extraction of the image is the key to the image recognition process. Selecting appropriate image features for different components can improve the efficiency of components recognition. In our study, we have extracted features of PCB board images from three aspects: color feature, shape feature, and texture feature.

Color is intuitively an important feature for detecting components, as many components are distinct in color from the PCB board. For example, a black SMD voltage resistor is distinct in color from a monochromatic green PCB, but not a black board. Color features can show good stability in different shapes and placement directions. Here, 13 different methods to extract color features have been implemented, of which 3 methods will be discussed in detail in Section 4.

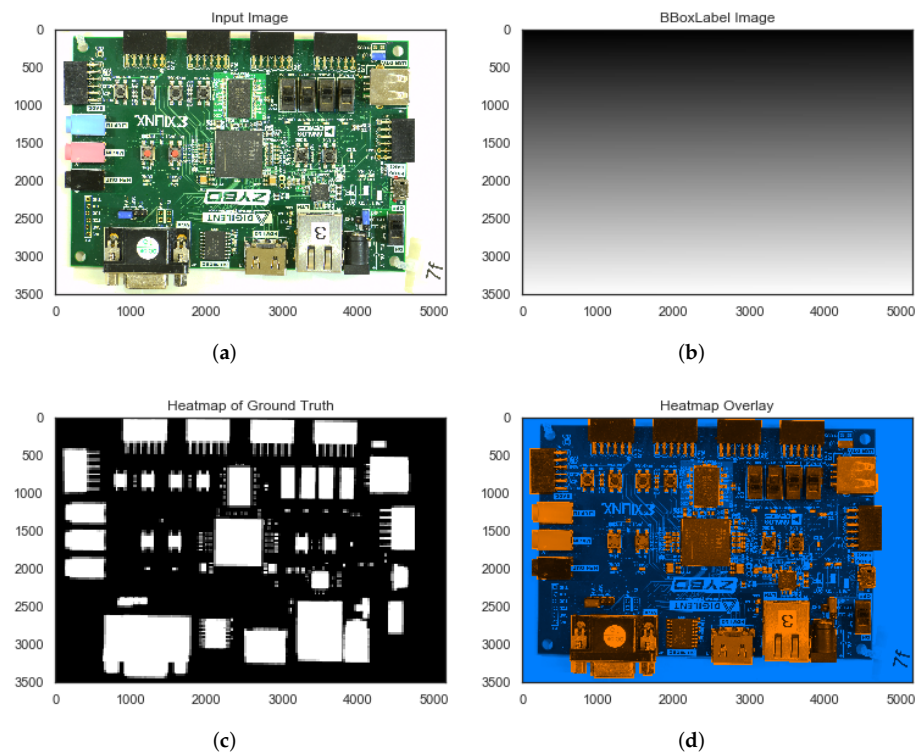


Figure 3. (a) The original PCB image; (b) the corresponding bbox labels; (c) the bbox ground truth heatmap for this PCB image; (d) and the heatmap overlay on the PCB image.

In addition, shape features are also intuitively with an important feature type for detecting PCB components, as many components consist of regular shapes. For example, many SMD resistors are rectangular, while many vias on the PCB appear circular from above. Shapes are crucial for humans to distinguish different objects, so they are regarded as very critical and important in computer vision and pattern recognition. There are many ways to express the shape of an object in a computer: local/boundary shape features and global shape features. Local shape features require prior segmentation, such as area, perimeter, etc. Global shape features do not require segmentation and can be computed on the entire image, such as edge, corner, and blob detection. Since this study concerns PCB component detection, i.e., prior segmentation is not known, we have used global shape features. Three feature extraction methods will be discussed in Section 5.

Finally, texture features are also intuitively beneficial for detecting PCB components, as the components are often composed of different materials which can possess different patterns when imaged. For example, plastic packaging materials on an IC tend to appear rougher and less reflective than the head of certain ceramic capacitors. The texture feature is an image feature that reflects the spatial distribution of pixels, and it is usually characterized by local irregularities and macroscopic regularities [14]. Different image grayscale pixel arrangements will produce different texture features to different components for distinction. Statistical methods and signal processing methods are the two main texture feature methods. Here, 9 texture feature extraction approaches have been implemented, 3 of which will be discussed in Section 6.

3.3. Feature Selection and Analysis

Feature selection is used to extract the optimal subsets for PCB component detection [15]. It can effectively eliminate irrelevant features, reduce data dimensions, and improve the accuracy and efficiency of classification models. In this step, we use the feature selection algorithm to rank the importance of the 1200 features extracted in the previous step.

Commonly-used feature selection algorithms are broadly categorized into filters, wrappers, and embedding methods [16]. At present, the feature selection mechanisms are mainly based on Information Theory, Neural Networks, Support Vector Machine (SVM) [17,18], etc.

Breiman proposed the Random Forest (RF) algorithm in 2001. This method operated by building a large number of decision trees and belongs to an ensemble algorithm [19]. Random forests can give the importance of features by calculating the average impurity reduction based on all decision trees in the forest. It won't need to make any assumptions about the linear separability of data [20].

In the process of splitting decision tree nodes in a random forest, we define the Gini importance [21] i as:

$$i = 1 - \sum_j p^2 \quad (1)$$

The proportion of samples marked j in the node is $p(j)$. After the split, the importance of Gini is:

$$\Delta i = i_{parent} - (p_{left} \cdot i_{left} + p_{right} \cdot i_{right}) \quad (2)$$

Among them, the sample proportions of the left child node and the right child node are p_{left} and p_{right} . i_{parent} , i_{left} , and i_{right} , respectively, represent the Gini importance of the parent node, the left child node, and the Gini importance of the right child node. For any feature X_i , the decreasing sum of impurities in all decision trees is the Gini importance of X_i :

$$\alpha \Delta I = \sum_k \Delta i_k \quad (3)$$

Based on this equation, this value indicates the importance of each feature.

4. Color Features

To better express the color feature of PCBs, we have used 13 methods, including RGB [22], RGB_CIE [23], HSV [24], HLS [25], LAB [26], LUV [27], YCrCb [28], YDbDr [29], YPbPr [30], XYZ [31], YIQ [32], YUV [33], and HED [34]. In this section, we discuss 3 kinds of color features, which are RGB, HSV and Lab color features. These three color features play an important role in image feature extraction computations.

4.1. RGB

The RGB color space is the most common representation for the color of pixels. The RGB color space uses the superposition of three primary colors in physics to form the principle of producing various colors. In RGB color space, the attributes of the three components of R, G, and B are independent, respectively representing the color of red, green, and blue. The gray-scale images of three channels in RGB color space and the original image are shown in Figure 4.

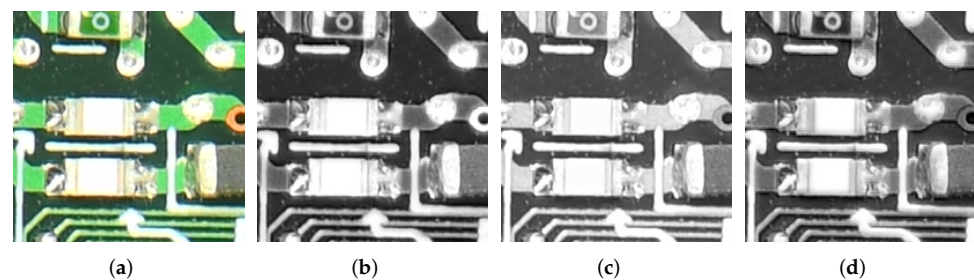


Figure 4. (a) The Original PCB Image; (b) R Channel of the Image; (c) G Channel of the Image; (d) B Channel of The Image.

4.1.1. Benefits

RGB is the most commonly used and most basic way of expressing color characteristics, and the expression of different components on the PCB is more intuitive and easy to understand.

4.1.2. Limitations

RGB color space is the most common hardware-oriented model, which is usually used in imaging and display systems, and is rarely used in image processing and feature extraction [22]. The color components of RGB space may be affected by light and the environment. The three color components are highly correlated, and the brightness will change with the transformation of the three components of R, G and B. Moreover, when one of the color components changes, it will also affect the other two color components to a certain extent. Two colors with the same chromaticity might be mistaken for each other when their intensities change even by the slightest amount. The challenging cases include distinguishing surface-mount resistors and inductors on a PCB.

4.2. HSV

A. R. Smith created the HSV color space in 1978 based on the intuitive characteristics of colors, and it is also called Hexcone Model [35]. This HSV model, respectively, represents hue (H), saturation (S), and value (V). To express color on the image, the Hue refers to an image's relative lightness and darkness. Saturation shows the proximity between color to the spectral color. The value indicates how bright the color is. HSL is similar to HSV in the color space and they are both related to the concept of the human visual system [24]. However, they are slightly different in conceptual expression [36]. Figure 5 shows the PCB image represented on the HSV color space.

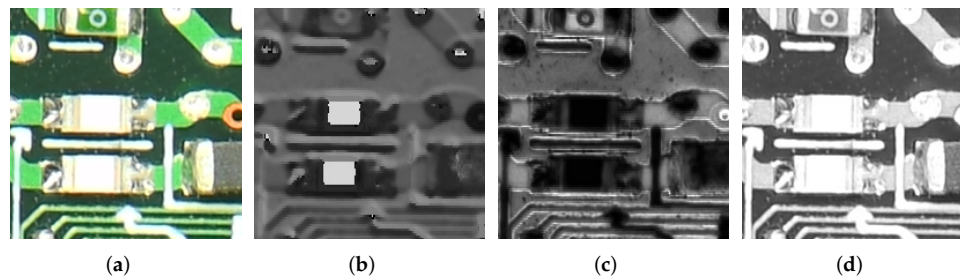


Figure 5. (a) The original PCB image; (b) H channel of the image; (c) S channel of the image; (d) V channel of the image.

4.2.1. Benefits

Each attribute of HSV directly corresponds to the basic color concept, which makes it conceptually simple and easy to understand. HSV can eliminate the influence of intensity components from the color information carried in color images. When the external illumination environment fluctuates slightly (as is frequently the case when optical images of PCBs are acquired), hue values vary less than RGB values. For instance, two colors of red may have comparable hue values yet vastly dissimilar RGB values. Thus, when differentiating identical components on a PCB under varying lighting conditions, the HSV color space may produce a more intuitive result.

4.2.2. Limitations

HSV is not suitable for use in illumination models. Many luminous mixed operations and luminous intensity operations cannot be implemented directly using HSV. A major disadvantage of the HSV spaces is that white, black, and gray do not have a distinct chromaticity; consequently, these colors are treated as singularities, making them difficult to detect. So, the components, e.g., resistors, inductors, diodes and ICs with the surfaces of these colors will be difficult to deal with using this color space.

4.3. Lab

The International Eclairage Committee (CIE) developed the Lab color model in 1931, which is improved in 1976 and named as a international standard color mode for color measurement. The L component represents the lightness of the pixel. Respectively, **a** and **b** represent the color range from red to green, and from yellow to blue. Figure 6 indicates the PCB image represented in Lab color space.

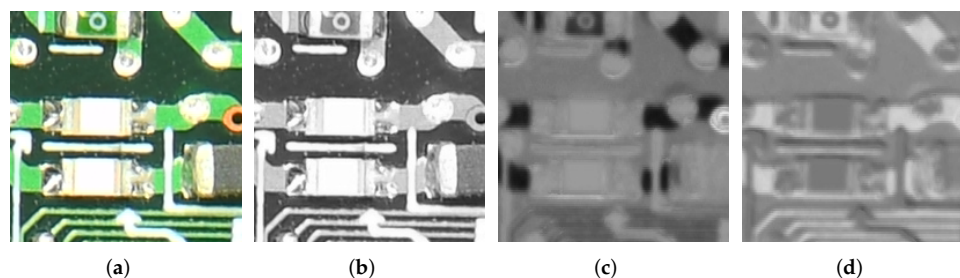


Figure 6. (a) The original PCB image; (b) L channel of the image; (c) A channel of the image; (d) B channel of the image.

4.3.1. Benefits

This color space can directly compare and analyze different colors by using the geometric distance in the color space. Certain kinds of components in PCB don't have visible differences to the naked eye, like the capacitors and the resistors, or different kinds of chips. Lab feature has a wide color gamut, so it can be effectively and conveniently used to measure slight color differences, such as surface-mount resistors and inductors.

4.3.2. Limitations

The creation of Lab space is relatively complicated, and the Lab color space generated is not as natural and understandable for humans as RGB or other perceptual color spaces [26]. This color space suffers from the same singularity issue as discussed in the limitations of the HSV color space.

In addition to the above-discussed color features (RGB, HSV, and Lab), the shape and texture features will help to detect the components of PCBs more efficiently that are discussed in the Section 3.2. We will converse about few significant shape features in-depth in the following section.

5. Shape Features

In our study, there are 11 types of shape features: Histogram of Gradients (HOG) [37], Scale Invariant Feature Transform (SIFT) [38], Oriented FAST and Rotated BRIEF (ORB) [39], Hough Line Transform, Hough Circle Transform [40], Determinant of Hessian (DoH)—Blob Detection [41], Fourier Transform [42], Connected Components [43], Corner Subpixels [44], Local Peak Maxima [45], and Edge Detection [46]. In this section, we will discuss the following three shape features: Determinant of Hessian (DoH)—Blob Detection, Corner Subpixels Detection, and Edge Detection features. These three shape features are significant in the extraction of image features.

5.1. Determinant of Hessian (DoH)—Blob Detection

The blob objects are generally bright on dark regions or dark on bright regions on an image, and can be extracted using three algorithms. One algorithm computes Laplacian of Gaussian (LoG) [47] with consecutively increasing standard deviation and piles them as a cubic structure. Local maxima of the cube are considered as blobs. This procedure executes very slowly on extracting larger blobs due to a high number of convolutions and only bright objects are detected in dark regions. An alternate method to LoG, Difference of Gaussian (DoG) [47], works on approximation by blurring the image during the convolution, so the difference between two consecutive blurred images are piled up as a cubic structure.

Unfortunately, this algorithm fails to detect larger blobs efficiently. The third algorithm, Determinant of Hessian (DoH), detects blobs using the DoH matrix by computing maxima. This method uses box filters instead of convolutions which removes the dependency of blob sizes in execution. Both bright and dark blobs are detected through this procedure.

5.1.1. Hyperparameters

We have set the hyperparameters by tuning them to get appropriate results as shown in Figure 7. The parameters are: minimum and maximum σ or standard deviation for Gaussian kernel, the threshold which is lower bound for scale-space maxima, overlap value that determines area limit of two blobs overlapped, and log scale set to default value, i.e., False.

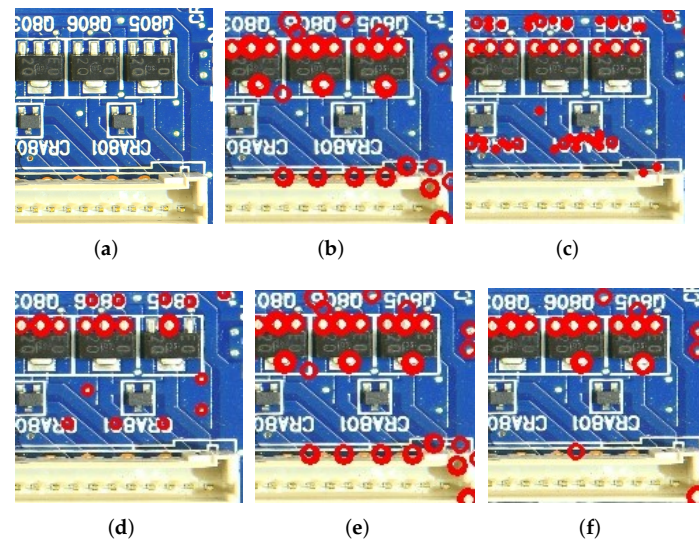


Figure 7. Determinant of Hessian–Blobs feature images with different label mask k-size. (a) Original image patch, and (b–f) respective experimental results from 25 to 5 image mask size. These six images are different images.

5.1.2. Benefits

Any circular and non-circular curvy components such as oscillators, transistors, and a number of diodes are easily detected using DoH blob features from PCB images.

5.1.3. Limitations

This algorithm will not be able to detect small blobs accurately such as vias on a PCB.

5.2. Corner Subpixels

We can represent corners as the pixel points with huge intensity variation from all directions around the pixel [48]. According to Harris and Stephens [49], the corners are identified based on the difference in the intensity for a displacement of (u, v) in all directions as shown below:

$$E(u, v) = \sum_{x,y} w(x, y) [I(x + u, y + v) - I(x, y)]^2 \quad (4)$$

The window function $w(x, y)$ could be either a Gaussian window or rectangular window used to add weights to pixels. To detect a corner, we need to maximize the $E(u, v)$ function. In the process of maximizing, we can apply Taylor expansion to the above equation that produces the below equation:

$$E(u, v) \approx [u \quad v] M \begin{bmatrix} u \\ v \end{bmatrix} \quad (5)$$

$$\text{where } M = \sum_{x,y} w(x,y) \begin{bmatrix} I_x I_x & I_x I_y \\ I_x I_y & I_y I_y \end{bmatrix}$$

In the above equation, I_x and I_y are derivatives of the image in x and y directions, respectively. We can determine whether a point is a corner or not by computing the scoring function R as below. If the value of R is large, it implies λ_1 and λ_2 are large and equivalent, then it is a corner.

$$R = \det(M) - k(\text{trace}(M))^2 \quad (6)$$

$$\text{where } \det(M) = \lambda_1 \lambda_2, \text{ trace}(M) = \lambda_1 + \lambda_2$$

In contrast to Harris corner algorithm, Shi-Tomasi [50] proposed a different approach of calculating the scoring function (R) as below, which produces better results compared to previous algorithms.

$$R = \min(\lambda_1, \lambda_2) \quad (7)$$

If λ_1 and λ_2 are greater than the minimum threshold R , it can be considered as a corner. Later, we apply refinement to the detected corners using the corner subpixel algorithm.

5.2.1. Hyperparameters

In this experiment, we have two primary functions: Shi-Tomasi's *GoodFeaturesToTrack* and *CornerSubpix*. The first function takes the number of corners, quality level, minimum distance, block size, and gradient size as the hyper-parameters that vary accordingly to the label mask's k -size. Similarly, the second function takes three parameters, window size, zero zone, and criteria, that are constant irrespective of the label mask's k -size. The results are shown in Figure 8.

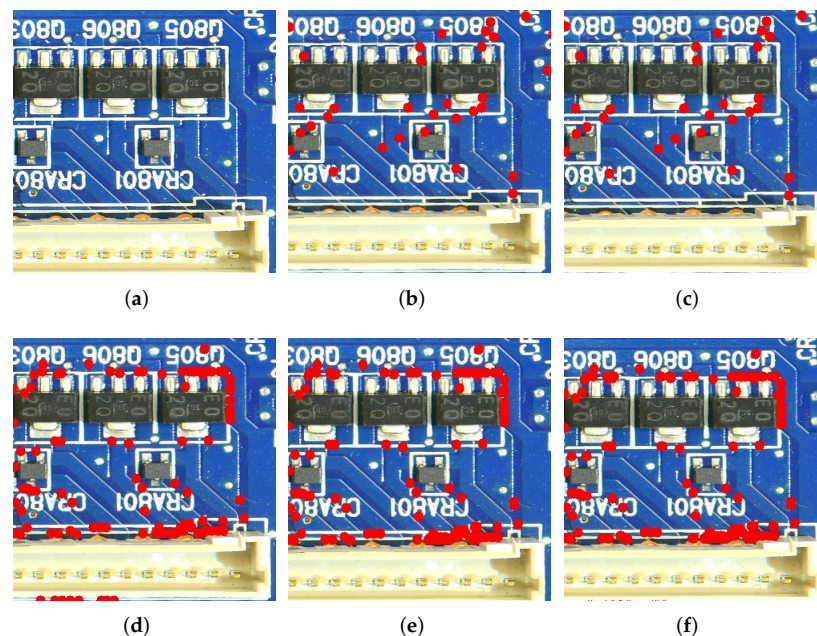


Figure 8. Corner Subpixel feature images with different label mask k -size. (a) Original image patch, and (b–f) respective experimental results from 25 to 5 image mask size. These six images are different images.

5.2.2. Benefits

Similar to Connected Components features, these features are scale, rotation, and illumination invariant. Using these corner features, we can easily identify the location of the component on the PCB.

5.2.3. Limitations

These features consume more memory space to process the algorithm due to redundancies and are not very robust in a complex image, for example, the image of a PCB with a high density of components or reference designators.

5.3. Edge Detection

Edge detection is the process of finding consecutive points of a sudden change in brightness that forms an edge in an image. The major steps involved in edge feature detection are gray-scaling, bilateral filtering for noise removal, edge detection using Canny algorithm [51,52], identifying contours around the detected edges, computing statistics such as the number of contours, maximum contour area, etc., as features. The Canny algorithm finds the gradients of the blurred image and utilizes the non-maximum suppression along with the hysteresis for removing spurious edges and weak edges from the detections, respectively. The Canny algorithm has been used due to its adaptability to variations in the images.

5.3.1. Hyperparameters

There are three different hyperparameters for each step. The diameter is set to 7, and sigma color and sigma space are set to 50 for the bilateral filtering. For the Canny edge algorithm, the lower threshold is set to (mean of gray intensities -25% of the mean), the upper threshold is set to (mean of gray intensities $+25\%$ of the mean), and *L2 Gradient* to false. Contour finding expects two significant parameters, i.e., retrieval mode is set to *RETR_EXTERNAL* and approximation method to *CHAIN_APPROX_NONE*.

5.3.2. Benefits

This edge feature detects both smaller and larger components on PCBs such as ICs, diodes, transistors, etc. The Canny edge detector aids in the detection of lines, which is advantageous for trace analysis in PCBs.

5.3.3. Limitations

These edge features could get biased towards horizontal and vertical edges in the images. There is a likelihood of wrong approximations of symmetry on rotations that are common in most PCBs, especially for ICs with square shapes.

Although the shape features play a significant role in image feature extraction, they are alone not sufficient enough to uniquely identify and detect the components on the PCB. As we have mentioned that texture indicates the spatial distribution of pixels, the texture features along with shape and color can improve the accuracy of detecting the PCB components. In the next section, we will explore the texture features in detail.

6. Texture Features

For texture features, we have used nine kinds of texture feature extraction methods: Gabor filter [53], Gray-level co-occurrence matrix (GLCM) [54], Local binary pattern (LBP) [55], Gray-level run length matrix (GLRLM) [56], Tamura [57], Law's Texture Energy Measures (LTEM) [58], Gray-level difference statistics [59], Autocorrelation function, and Segmentation-based fractal texture analysis (SFTA) [60]. Among these nine methods, GLCM, LBP, and Gabor filters have a more obvious effect of distinguishing components, and they are also the most common feature extraction methods for image texture features. In this section, we will introduce and analyze these three methods.

6.1. Gabor Filter

To better describe the texture information of the image, the first method we choose is Gabor filter. It's usually used in signal processing. To describe the local frequency information of the image signal, the Gabor kernel adds a window function to the signal in the frequency domain [61]. Gabor filter kernel is similar to the receptive field of vertebrate

visual cortex [62], which provides a satisfactory result for texture representation and discrimination [63].

6.1.1. Hyperparameters

λ represents the wavelength of the filter function. The longer the wavelength, the greater the interval between black and white stripes in the Gabor kernel image. θ represents the tilt angle of the kernel function image, which can be used for effective feature extraction for textures in different directions. ψ determines the phase shift. When ψ is 0, the kernel center is a white stripe; when ψ is 180, the kernel center is a black stripe. σ is the standard deviation of the Gaussian function, which reflects the effective size of the kernel. γ is the spatial aspect ratio, which determines the ellipticity of the filter kernel function [63].

After testing, when $\lambda = 14$, $\psi = 0$, $\sigma = 5$, $\gamma = 1$, the texture feature extraction effect is the best. Considering that the components on the PCB may have different placement directions, we set six values for θ : 0° , 30° , 60° , 90° , 120° , 150° . The output images after applying Gabor filtering are shown in Figure 9.

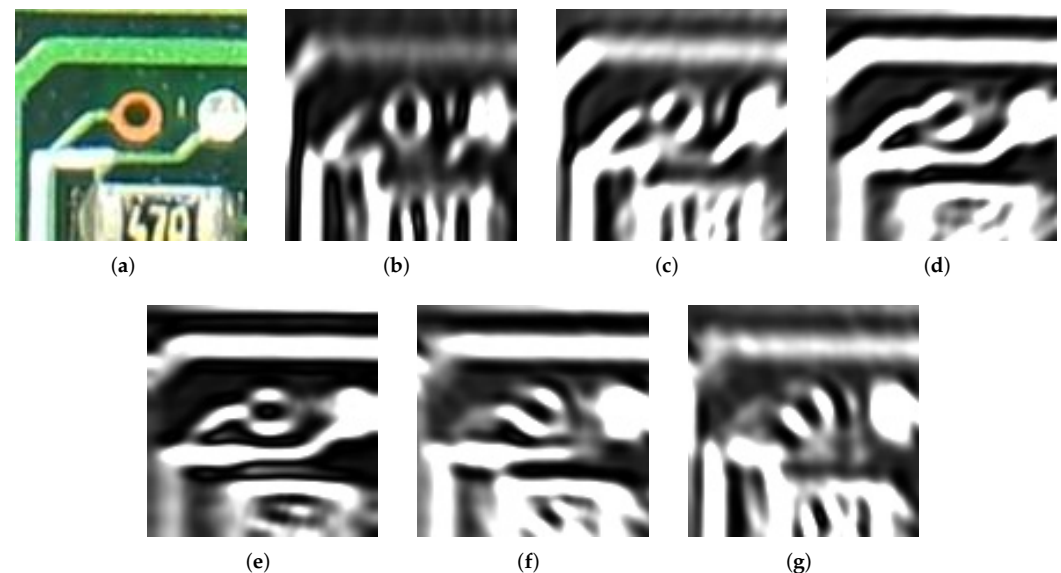


Figure 9. (a) A part of the original PCB image; (b) filtered image when $\theta = 0^\circ$; (c) filtered image when $\theta = 30^\circ$; (d) filtered image when $\theta = 60^\circ$; (e) filtered image when $\theta = 90^\circ$; (f) filtered image when $\theta = 120^\circ$; and (g) filtered image when $\theta = 150^\circ$

6.1.2. Benefits

For the kernels, we have chosen six directions, which can basically cover the direction of all components on the PCB board, so the Gabor filter has the characteristic of rotation invariance. For a certain degree of image rotation and distortion, Gabor filter can still provide better results. In addition, Gabor filter is insensitive to light changes. If the light at each position on the PCB is not exactly the same, it is able to provide a satisfactory adaptability.

6.1.3. Limitations

The limitation of the Gabor filter is that it is non-orthogonal, which will result in different proportions of redundant features [64]. In addition, due to the high frequency response at the image edge, a “ring” effect may occur [65]. This has potential to create problems while detecting vias on a PCB.

6.2. Gray-Level Co-Occurrence Matrix

Gray-level co-occurrence matrix (GLCM) was proposed by Haralick et al. in 1973, which refers to a simple way to describe textures by studying the distribution of different

gray levels corresponding to different textures in space [66]. That is, the spatial correlation in grayscale images [67].

6.2.1. Hyperparameters

We select four angle directions of 0° , 45° , 90° , 135° to calculate GLCM. With a small ksize, the best step size should be set to one pixel, otherwise the output images will become blurry. We have also changed the 8-bit pixel to 4-bit (which means the gray-level is equal to 16) to improve the computational efficiency. The output images when the angle is 0° is shown in Figure 10.

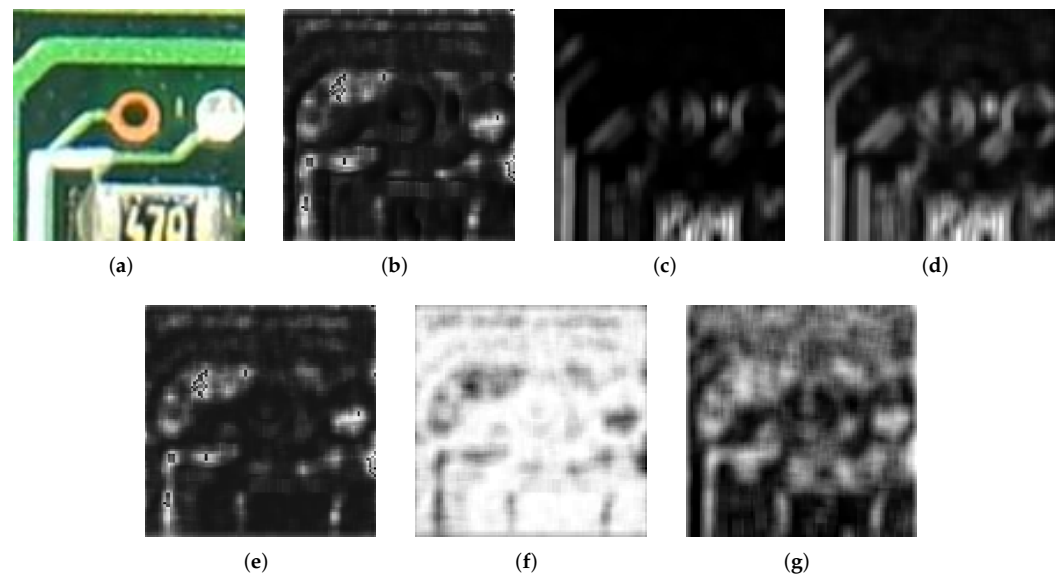


Figure 10. (a) A part of the original PCB image; (b) ASM image when $\theta = 0^\circ$, (c) contrast image when $\theta = 0^\circ$; (d) dissimilarity image when $\theta = 0^\circ$; (e) energy image when $\theta = 0^\circ$, (f) entropy image when $\theta = 0^\circ$; and (g) homogeneity image when $\theta = 0^\circ$. These six images are different images.

6.2.2. Benefits

GLCM has strong adaptability and robustness. Its features can be produced for a single orientation or for a group of orientations, making GLCM direction independent, which may effectively cover the orientation of all components on the PCB.

6.2.3. Limitations

As a statistical method for texture feature extraction, GLCM has less correlation with human visual models and lacks the use of global information. With high computational complexity, GLCM has long execution time. This technique is not suitable for distinguishing between different text fonts that is why this feature faces difficulty when detecting component markings and reference designators on a PCB.

6.3. Local Binary Pattern

LBP reflects the texture changes around the image pixels. We define a 3×3 window as the LBP operator. For the center pixel in the window, compare the pixel values with its neighboring 8 pixels. Mark the surrounding positions as 0 or 1 according to the compared result. Greater than the central pixel value is recorded as 1, otherwise it is recorded as 0. In this case, we get an 8-bit binary number, and then we convert it to a decimal number. Use the obtained decimal value to reflect the texture information of the 3×3 area, which is also the LBP value of the center pixel of the window [68].

The LBP at this time is available to represent texture feature, but not rotation-invariant. As the image rotates, the pixel position changes, and the LBP value will change accordingly, and the given feature values will be very different. Maenpaa et al. proposed a method to

achieve Rotated Local Binary Pattern (LBP) operation: continuously rotate the circular field to obtain all possible initial defined LBP values, and then take the minimum value as the value of the field [69,70].

Next, in order to further improve statistical capabilities, it is necessary to solve the problem of excessive binary patterns. Ojala et al. came up with a unified mode adaptation to reduce the dimensionality of the LBP operator mode type, which is called the unified local binary mode (ULBP). When the cyclic binary number corresponding to a certain LBP changes from 0 to 1 or from 1 to 0 at most two transitions, the binary corresponding to the LBP is called a uniform pattern class.

In here, we have combined the RLBP and the ULBP to form the most powerful rotated uniform LBP feature.

6.3.1. Hyperparameters

In the calculation process of this algorithm, a sliding window is needed. If the window is too small, there will be mis-segmentation within the same texture; while when the analysis window is too big, there will be mis-segmentation in the texture boundary area. Since the region size is relatively small and the window size is 3×3 , the testing result shows that the texture effect has the best form. We can see the output image in Figure 11.

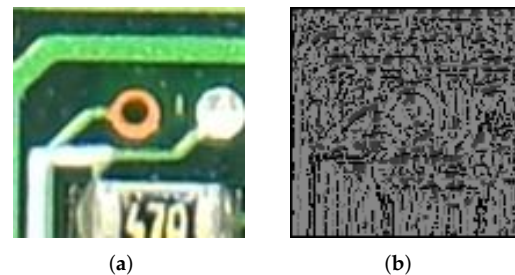


Figure 11. (a) A part of the original PCB image; (b) The output image after RLBP operators and ULBP operators.

6.3.2. Benefits

The RLBP and the ULBP are combined to retain the most effective feature value, and this method has rotation invariance, gray scale invariance, and invariance to monotonic illumination changes [66]. This helps with the same type of components with different orientations on a PCB in variant lighting conditions.

6.3.3. Limitations

The calculation time is positively correlated with to the number of pixels in the image. As the image grows larger, it will take longer to perform calculations [71]. Moreover, noise, blurring, and any other disturbing effect, all have noticeable impacts on the method performance. So, in case of blurry markings or having reference designators and incorporated noise due to image acquisition, LBP texture feature might not be suitable for PCB assurance.

In the next section, we will compare the different results from the feature extraction experiment to recognize the importance of few features based on their impact on PCB component detection.

7. Results

Among the employed 34 feature extraction methods, 13 of them are color, 12 for shape, and nine are for texture. Overall, we have used these 34 feature extraction methods to obtain a total of 1200 PCB board image features. The methods used for each type of feature and the number of generated features are shown in Table 1. After that, we have used random forest to give the importance of each features. The values of “feature importance” parameter show the features with the highest influence over the classification and regression results. We

have obtained the output data of the importance of different features with different ksizes. Different sizes of the region also have a certain impact on the feature extraction results.

Table 1. The methods used for each type of feature and the number of generated features.

Feature Types	Methods	Number of Features
Color Feature	RGB	12
	RGB_CIE	12
	HSV	12
	LAB	12
	LUV	12
	YCrCb	12
	YDbDr	12
	YPbPr	12
	XYZ	12
	YIQ	12
	YUV	12
	HED	12
	HLS	12
Shape Feature	Histogram of Gradients (HOG)	36
	Scale Invariant Feature Transform (SIFT)	384
	Oriented FAST and Rotated BRIEF (ORB)	320
	Hough Line Transform	6
	Hough Circle Transform	9
	Determinant of Hessian (DoH) - Blob Detection	6
	Fourier Transform	36
	Connected Components	3
	Corner Subpixels	10
	Local Peak Maxima	2
	Edge Detection	3
Texture Feature	Gabor filter	24
	Gray-level co-occurrence matrix (GLCM)	24
	Local binary pattern (LBP)	10
	Gray-level run length matrix (GLRLM)	44
	Tamura	3
	Law's Texture Energy Measures (LTEM)	60
	Gray-level difference statistics	12
	Autocorrelation function	4
	Segmentation-based fractal texture analysis (SFTA)	48

The box plot is used to depict the center and spread of the data distribution, so here we use the box plot to represent the feature data distribution under different ksizes, as shown in Figure 12. We also give the statistics values of this data distribution correspond to the boxplot in Figure 12, which is shown in Table 2.

Table 2. Statistics for different ksizes generated according to the boxplot in Figure 12.

Ksize	Count	Mean	Std	Min	25%	50%	75%	Max
Ksize 5	1200	0.005833	0.013386	0	0	0	0.007345	0.134498
Ksize 10	1200	0.011667	0.021662	0	1.24×10^{-7}	0.000733	0.018082	0.216028
Ksize 15	1200	0.012500	0.020859	0	0.000031	0.003261	0.018928	0.218144
Ksize 20	1200	0.012500	0.019310	0	0.000136	0.004696	0.017373	0.017373
Entry 25	1200	0.012500	0.017986	0	0.000715	0.005582	0.017005	0.185656

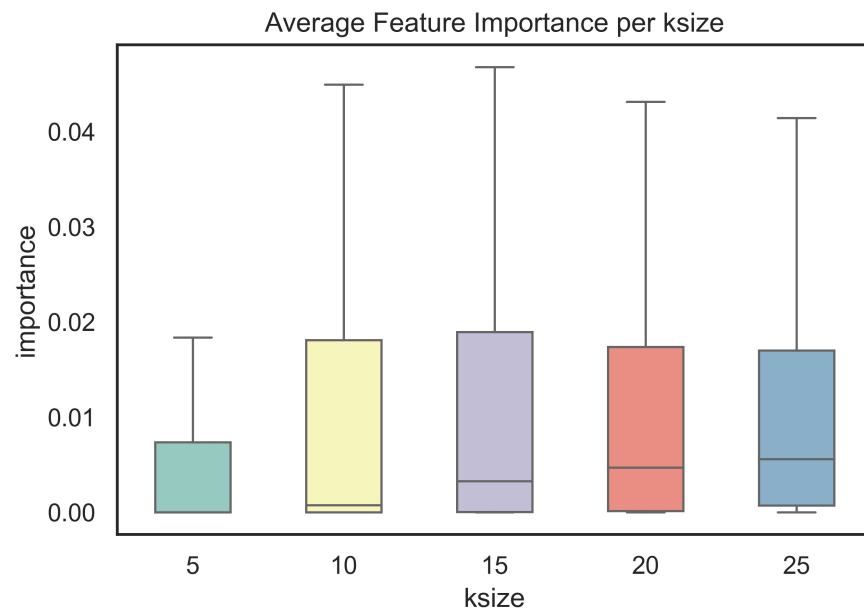


Figure 12. The boxplot for different ksizes. It indicate that ksize 25 has the higher median and shorter distance, so ksize 25 is the best among the 5 different ksizes.

In the box plot, each group of ksize data is composed of the sum of corresponding features in 15 PCB board images. The two ends of the box are the upper and lower quartiles, which are the median of numbers larger than the overall median, and the median of numbers smaller than the overall median. The height of the box, which is the distance between first quartile and the third quartile, reflects the degree of data fluctuation. So, the distribution of the data characteristics for ksize15 is more scattered, while for ksize5 is more concentrated. Although the overall eigenvalues of ksize5 are close to 0, so it is not a proper choice.

According to the higher median and shorter distance shown in the boxplot of Figure 12, ksize of 25 has the highest significance among the 5 different ksizes. This result is explained based on the fact that the smaller ksizes are very unfavorable for extraction of the shape and texture features. For example, it is difficult for a small region in the shape feature to reflect the lines and corners, while it is difficult to reflect the repeatability among pixels in the texture feature.

The distribution of the importance of the color feature values is generally higher than shape and texture features. Since the regional features of the image are related to the entire shape area, the shape features in a single area of the PCB board image after bounding box segmentation are not ideal. The distribution of feature types are displayed in Figure 13. The corresponding statistic values are shown in Table 3.

Through sorting the features based on their importance, we have selected the five most important features, which are “HLS_2_mean”, “LAB_1_med”, “LAB_1_mean”, “HED_1_med”, and “HED_1_mean”. Note that in this study, color features has been identified by their color space, channel, and the function used to compute a single feature value from each region. For example, the most important feature, “HLS_2_mean”, indicates the mean of the second channel of each region when the image was converted to HLS color space was considered the most important. The distribution of these feature values in 15 images are shown in Figure 14. Those five significant features belong to the color type. We can also find the statistic values of the importance of these five features in Table 4.

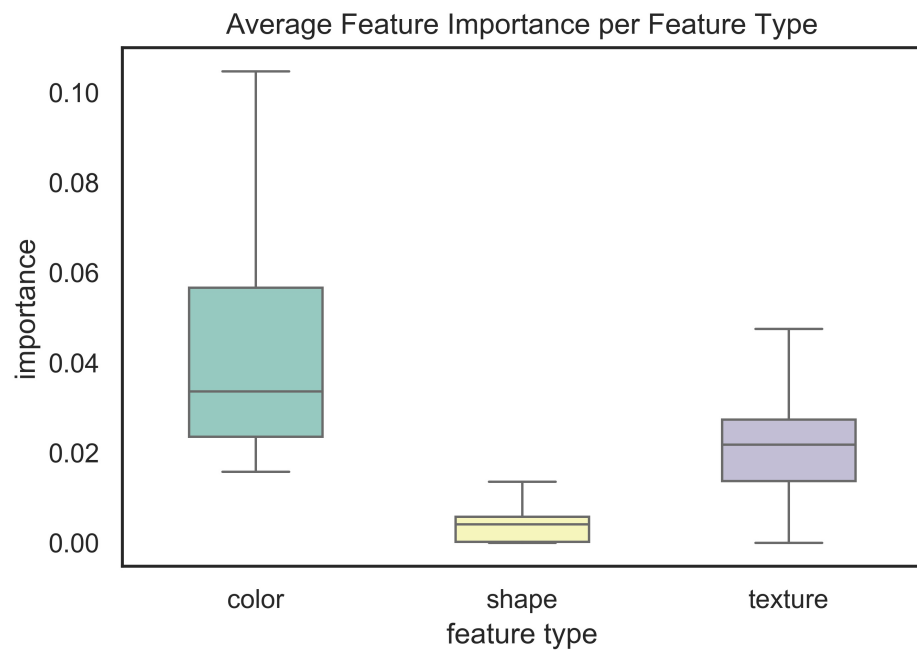


Figure 13. The boxplot for different feature types in different images. The distribution of the color feature in the box plot shows that it is the most effective feature among the three types of features.

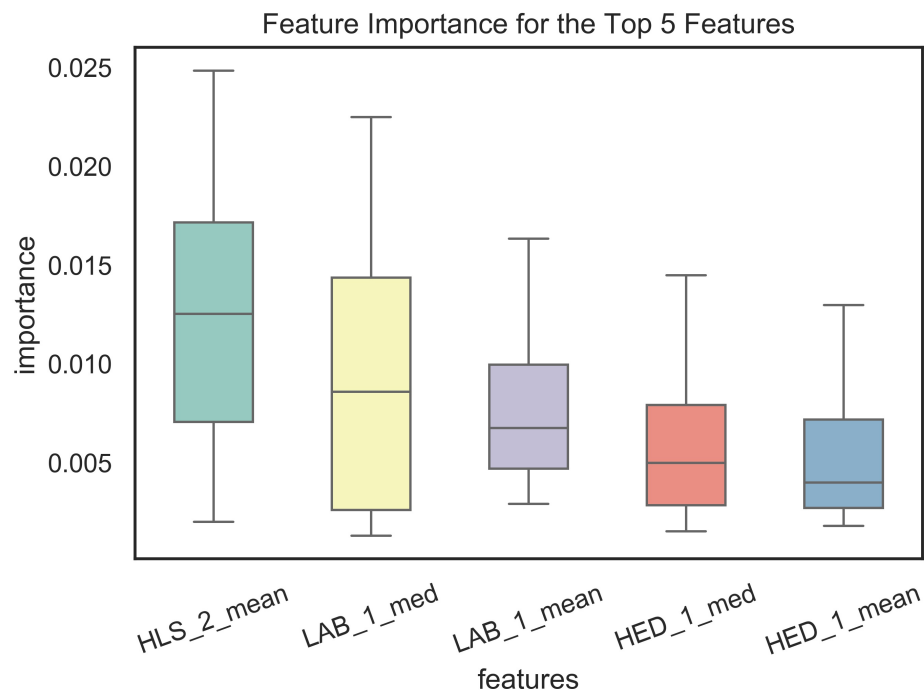


Figure 14. The boxplot for the five most important feature types in different images. The top five important features all come from color features, this also shows that the color feature is the most important among the three types of features.

Table 3. Statistics for different feature types generated according to the boxplot in Figure 13.

Ksize	Count	Mean	Std	Min	25%	50%	75%	Max
Color	156	0.043102	0.026751	0.015766	0.023544	0.033614	0.056612	0.185656
Shape	815	0.003625	0.003406	0	0.000186	0.004119	0.005780	0.014832
Texture	229	0.023240	0.011621	0	0.013738	0.021807	0.027370	0.058496

Table 4. Statistics for the five most important feature methods generated according to the boxplot in Figure 14.

Feature	Count	Mean	Std	Min	25%	50%	75%	Max
HLS_2_mean	15	0.012377	0.007098	0.002031	0.007076	0.012545	0.017180	0.024853
LAB_1_med	15	0.009547	0.007315	0.001328	0.002635	0.008607	0.014378	0.022514
LAB_1_mean	15	0.007753	0.004246	0.002942	0.004711	0.006776	0.009983	0.016350
HED_1_med	15	0.006981	0.005926	0.001548	0.002860	0.005006	0.007932	0.021296
HED_1_mean	15	0.006846	0.006392	0.001821	0.002721	0.004022	0.007194	0.023355

8. Discussion

Overall, the results of this study demonstrate color features that are completely informative and useful for PCB component detection. As shown in Section 7, color features for most cases demonstrated higher levels of importance in PCB component detection than shape and texture features. In other words, color features are generally more informative for the task of PCB component detection. The higher strength of color features is due to monochrome property (a single base color) of PCB boards and the fact that they process a distinct color from the components. Therefore, we consider color features as great candidates for modeling the PCB boards with satisfactory performance. With respect to execution time, color features tend to be the fastest to get extracted because no parameter tuning is necessary and many of the algorithm operations are vectorised and this trait of color features represents a promising prospect for real-time PCB assurance.

Table 5 represents a comparative analysis between existing works and our study. It is evident from the table that our work demonstrates a novel high-performance approach to investigate the impact of different features in recognizing certain components on the PCB using semantic image datasets. Compared with [72], which is also aimed at detection, we use the color, shape, and texture in the basic features of the image. However, their work only focuses on the ORB method, a shape feature descriptor that is not convincing for extracting the best feature. In addition, our work has proved that color feature is more important than shape feature when detecting PCB components. Different from our work, Mahalingam et al. [73,74] are all based on using neural networks to do analysis and detection. These methods cannot completely explain the reasons behind the falsely detected components and the impact of the features of the image on those wrong predictions, but our work focuses on analyzing the features and their impact on detection results. Our work uses semantic PCB images as the datasets, which classify and localize every pixel in an image so that we can localize all the components in an image to facilitate feature analysis. This approach will help the component detection algorithms explain their decisions and provide insight into the impact of adversarial examples on the detection results. As for bounding box and bare PCB images used in [72,73,75,76], they are all more suitable for analysis of the overall PCB image.

Table 5. Comparative analysis between existing works and our study.

Papers	Dataset	Use Cases	Method	Result
[75]	CAD files of the PCB and bare PCB image datasets	PCB inspection	LIF (Learning Inspection Features) and OLI(On-line Inspection)	Detection accuracy exceeded 97%.
[72]	Bounding box PCB image datasets	Detecting specific PCBs and recognizing mainboards	ORB features and Random Forest	The PCB recognition accuracy is 98.6% and the classification accuracy is 83%.

Table 5. Cont.

Papers	Dataset	Use Cases	Method	Result
[73]	Bounding box PCB image datasets	Component analysis, IC detection and localization	YOLO, Faster-RCNN, Retinanet-50	The mean average precision of these 3 techniques are: 0.698, 0.783 and 0.833.
[74]	Semantic PCB image datasets	PCB element detection	SSD neural network	The mean average precision of normal, enhanced, and ideal images are 0.9209, 0.9272, and 0.9510.
[76]	Bare PCB image datasets	Defect detection and classification	Image processing and flood fill operation	Classified up to 7 defects and the defects are identified successfully.
Our work	Semantic PCB image datasets	Analyzing a variety of common computer vision-based features for the task of PCB component detection	34 feature extraction methods for color, shape, and texture; Random Forest	For most of the cases, color features demonstrated higher levels of importance in PCB component detection than shape and texture features.

9. Conclusions

In this study, we have found color features are faster in overall to extract and more accurate for PCB component detection than shape and texture features. It is important to note that this is a controlled experiment, as all PCB images used in this study were obtained using similar lighting conditions. Since color features are sensitive to such conditions (e.g., a brown capacitor imaged under dim light could appear similar to a black resistor), a color normalization technique and preprocessing algorithms should be utilized prior to feature extraction to ensure better generalizability.

Though shape and texture features have been generally slower to extract and less accurate for PCB component detection than color features, they could still be very useful for PCB assurance. For example, after the components have been detected, local shape features such as size, aspect ratio, and shape complexity could be helpful for classifying the different PCB components, as many component types have a distinct footprint (e.g., 3-prong transistors vs. 8-pin DIP ICs). In addition, texture features such as regional smoothness and variance could be helpful for detecting visual defects such as scratches, spurious copper, and mousebites. Since both shape and texture features are sensitive to lighting conditions as well as imaging resolution, both color and shape normalization techniques and preprocessing algorithms should be utilized prior to feature extraction to ensure better generalizability.

Our work and data will be applied to do feature selection for PCB component detection using semantic data. In future works, the tradeoffs between using more and diverse data and as well as transferring knowledge in hardware assurance applications should be comprehensively explored. Employing large high number of varied data samples would be effective for common cases for which we have lots of examples (such as off-the-shelf

components), but such data is time-consuming to collect and can be very difficult to acquire on competitor or foreign custom components, legacy devices, and malicious modifications. On the other hand, leveraging strong prior knowledge would increase the explainability of the system and reduce the amount of data needed for difficult-to-obtain cases. However, domain knowledge can be difficult to translate into an algorithm for certain data types. Since the hardware assurance domain is a complex and constantly evolving field, there is a great demand for both approaches. All in all, this work can be considered as a new direction and motivation for the artificial intelligence and computer vision communities to get involved in hardware security studies and vice versa.

Author Contributions: Conceptualization, W.Z. and S.R.G.; methodology, W.Z. and S.R.G.; validation, W.Z. and S.R.G.; formal analysis, W.Z. and S.R.G.; investigation, W.Z. and S.R.G.; resources, W.Z. and S.R.G.; writing—original draft preparation, W.Z., S.R.G., S.T., S.G. and M.A.M.S.; writing—review and editing, W.Z., S.T. and S.G.; visualization, W.Z., S.R.G. and S.G.; supervision, S.T. and N.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: Olivia Paradis, for collecting the data, conceptualizing the experiment, and writing code.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Lu, H.; Mehta, D.; Paradis, O.P.; Asadizanjani, N.; Tehranipoor, M.; Woodard, D. FICS-PCB: A Multi-Modal Image Dataset for Automated Printed Circuit Board Visual Inspection. *IACR Cryptol. ePrint Arch.* **2020**, *2020*, 366.
2. Mehta, D.; Lu, H.; Paradis, O.P.; MS, M.A.; Rahman, M.T.; Iskander, Y.; Chawla, P.; Woodard, D.L.; Tehranipoor, M.; Asadizanjani, N. The Big Hack Explained: Detection and Prevention of PCB Supply Chain Implants. *J. Emerg. Technol. Comput. Syst.* **2020**, *16*, 1–25. [[CrossRef](#)]
3. Paradis, O.P.; Jessurun, N.T.; Tehranipoor, M.; Asadizanjani, N. Color normalization for robust automatic bill of materials generation and visual inspection of PCBs. In Proceedings of the ISTFA 2020, ASM International, Pasadena, CA, USA, 15–19 November 2020; pp. 172–179.
4. Botero, U.J.; Wilson, R.; Lu, H.; Rahman, M.T.; Mallaiyan, M.A.; Ganji, F.; Asadizanjani, N.; Tehranipoor, M.M.; Woodard, D.L.; Forte, D. Hardware Trust and Assurance through Reverse Engineering: A Tutorial and Outlook from Image Analysis and Machine Learning Perspectives. *J. Emerg. Technol. Comput. Syst.* **2021**, *17*, 1–53. [[CrossRef](#)]
5. Zhao, Z.Q.; Zheng, P.; Xu, S.-T.; Wu, X. Object Detection With Deep Learning: A Review. *IEEE Trans. Neural Netw. Learn. Syst.* **2019**, *30*, 3212–3232. [[CrossRef](#)] [[PubMed](#)]
6. Sun, Z.; Bebis, G.; Miller, R. Object detection using feature subset selection. *Pattern Recognit.* **2004**, *37*, 2165–2176. [[CrossRef](#)]
7. Jović, A.; Brkić, K.; Bogunović, N. A review of feature selection methods with applications. In Proceedings of the 2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), Opatija, Croatia, 25–29 May 2015; pp. 1200–1205. [[CrossRef](#)]
8. Geirhos, R.; Rubisch, P.; Michaelis, C.; Bethge, M.; Wichmann, F.; Brendel, W. ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. *arXiv* **2019**, arXiv:1811.12231.
9. Khan, F.; Anwer, R.M.; van de Weijer, J.; Bagdanov, A.D.; Vanrell, M.; López, A.M. Color attributes for object detection. In Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, 16–21 June 2012; pp. 3306–3313.
10. Diplaros, A.; Gevers, T.; Patras, I. Combining color and shape information for illumination-viewpoint invariant object recognition. *IEEE Trans. Image Process.* **2006**, *15*, 1–11. [[CrossRef](#)]
11. Bansal, M.; Kumar, M.; Kumar, M. 2D Object Recognition Techniques: State-of-the-Art Work. *Arch. Comput. Methods Eng.* **2020**, *28*, 1147–1161. [[CrossRef](#)]
12. O'Mahony, N.; Campbell, S.; Carvalho, A.; Harapanahalli, S.; Hernandez, G.V.; Krpalkova, L.; Riordan, D.; Walsh, J. Deep Learning vs. Traditional Computer Vision. In Proceedings of the 2019 Computer Vision Conference (CVC), Las Vegas, NV, USA, 2–3 May 2020; pp. 128–144.

13. Li, J.; Cheng, K.; Wang, S.; Morstatter, F.; Trevino, R.P.; Tang, J.; Liu, H. Feature Selection: A Data Perspective. *ACM Comput. Surv.* **2017**, *50*, 1–45. [[CrossRef](#)]
14. Chen, C.H. *Handbook of Pattern Recognition and Computer Vision*; World Scientific: Singapore, 2015.
15. Duda, R.O.; Hart, P.E.; Stork, D.G. *Pattern Classification and Scene Analysis*; Wiley: New York, NY, USA, 1973; Volume 3.
16. Guyon, I.; Elisseeff, A. An introduction to variable and feature selection. *J. Mach. Learn. Res.* **2003**, *3*, 1157–1182.
17. Liu, H.; Motoda, H. *Feature Selection for Knowledge Discovery and Data Mining*; Springer Science & Business Media: Cham, Switzerland, 2012; Volume 454.
18. Kononenko, I. Estimating attributes: Analysis and extensions of RELIEF. In Proceedings of the European Conference on Machine Learning, Catania, Italy, 6–8 April 1994; pp. 171–182.
19. Ma, L.; Fan, S. CURE-SMOTE algorithm and hybrid algorithm for feature selection and parameter optimization based on random forests. *BMC Bioinform.* **2017**, *18*, 169. [[CrossRef](#)] [[PubMed](#)]
20. Raschka, S. *Python Machine Learning*; Packt Publishing Ltd.: Birmingham, UK, 2015.
21. Breiman, L. Random forests. *Mach. Learn.* **2001**, *45*, 5–32. [[CrossRef](#)]
22. Pitas, I. *Digital Image Processing Algorithms and Applications*; John Wiley & Sons: New York, NY, USA, 2000.
23. Martínez, J.; Pérez-Ocón, F.; García-Beltrán, A.; Hita, E. Mathematical determination of the numerical data corresponding to the color-matching functions of three real observers using the RGB CIE-1931 primary system and a new system of unreal primaries $X' Y' Z'$. *Color Res. Appl.* **2003**, *28*, 89–95. [[CrossRef](#)]
24. Wen, C.Y.; Chou, C.M. Color image models and its applications to document examination. *Forensic Sci. J.* **2004**, *3*, 23–32.
25. Setiawan, N.A.; Seok-Ju, H.; Jang-Woon, K.; Chil-Woo, L. Gaussian mixture model in improved hls color space for human silhouette extraction. In Proceedings of the International Conference on Artificial Reality and Telexistence, Hangzhou, China, 29 November–1 December 2006; pp. 732–741.
26. Chavolla, E.; Zaldivar, D.; Cuevas, E.; Perez, M.A. Color spaces advantages and disadvantages in image color clustering segmentation. In *Advances in Soft Computing and Machine Learning in Image Processing*; Springer: Cham, Switzerland, 2018; pp. 3–22.
27. Kekre, H.; Thepade, S.; Sanas, S. Improving performance of multileveled BTC based CBIR using sundry color spaces. *Int. J. Image Process.* **2010**, *4*, 620–630.
28. El Baf, F.; Bouwmans, T.; Vachon, B. A fuzzy approach for background subtraction. In Proceedings of the 2008 15th IEEE International Conference on Image Processing, San Diego, CA, USA, 12–15 October 2008; pp. 2648–2651.
29. Sahdra, G.S.; Kailey, K.S. Detection of Contaminants in Cotton by using YDbDr color space. *Int. J. Comput. Technol. Appl.* **2012**, *3*, 1118–1124.
30. Campadelli, P.; Lanzarotti, R.; Lipori, G.; Salvi, E. Face and facial feature localization. In Proceedings of the International Conference on Image Analysis and Processing, Cagliari, Italy, 6–8 September 2005; pp. 1002–1009.
31. Kekre, H.; Sonawane, K. Comparative study of color histogram based bins approach in RGB, XYZ, Kekre's LX_Y and L' X' Y' color spaces. In Proceedings of the 2014 International Conference on Circuits, Systems, Communication and Information Technology Applications (CSCITA), Mumbai, India, 4–5 April 2014; pp. 364–369.
32. Liu, Z.; Liu, C. A hybrid color and frequency features method for face recognition. *IEEE Trans. Image Process.* **2008**, *17*, 1975–1980.
33. Sudhir, R.; Baboo, L.D.S.S. An efficient CBIR technique with YUV color space and texture features. *Comput. Eng. Intell. Syst.* **2011**, *2*, 85–95.
34. Birchfield, S. *Color, in Image Processing and Analysis*, 1st ed.; Cengage Learning: Boston, MA, USA, 2018; pp. 401–442.
35. Smith, A.R. Color gamut transform pairs. *ACM Siggraph Comput. Graph.* **1978**, *12*, 12–19. [[CrossRef](#)]
36. Ibraheem, N.A.; Hasan, M.M.; Khan, R.Z.; Mishra, P.K. Understanding color models: A review. *ARPN J. Sci. Technol.* **2012**, *2*, 265–275.
37. Dalal, N.; Triggs, B. Histograms of oriented gradients for human detection. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA, 20–25 June 2005; Volume 1, pp. 886–893. [[CrossRef](#)]
38. Lowe, D. Object recognition from local scale-invariant features. In Proceedings of the Proceedings of the Seventh IEEE International Conference on Computer Vision, Kerkyra, Greece, 20–27 September 1999; Volume 2, pp. 1150–1157. [[CrossRef](#)]
39. Rublee, E.; Rabaud, V.; Konolige, K.; Bradski, G. ORB: An efficient alternative to SIFT or SURF. In Proceedings of the IEEE International Conference on Computer Vision, Barcelona, Spain, 6–13 November 2011; pp. 2564–2571. [[CrossRef](#)]
40. Duda, R.O.; Hart, P.E. Use of the Hough Transformation to Detect Lines and Curves in Pictures. *Commun. ACM* **1972**, *15*, 11–15. [[CrossRef](#)]
41. Lindeberg, T. Image Matching Using Generalized Scale-Space Interest Points. In Proceedings of the International Conference on Scale Space and Variational Methods in Computer Vision, Leibnitz, Austria, 2–6 June 2013; pp. 355–367.
42. Zheng, Y.; Meng, F.; Liu, J.; Guo, B.; Song, Y.; Zhang, X.; Wang, L. Fourier transform to group feature on generated coarser contours for fast 2D shape matching. *IEEE Access* **2020**, *8*, 90141–90152. [[CrossRef](#)]
43. Häfner, M.; Uhl, A.; Wimmer, G. A novel shape feature descriptor for the classification of polyps in HD colonoscopy. In Proceedings of the International MICCAI Workshop on Medical Computer Vision, Nagoya, Japan, 26 September 2013; pp. 205–213.

44. Lucchese, L.; Mitra, S.K. Using saddle points for subpixel feature detection in camera calibration targets. In Proceedings of the Asia-Pacific Conference on Circuits and Systems, Denpasar, Indonesia, 28–31 October 2002; Volume 2, pp. 191–195.
45. Brieu, N.; Schmidt, G. Learning size adaptive local maxima selection for robust nuclei detection in histopathology images. In Proceedings of the 2017 IEEE 14th International Symposium on Biomedical Imaging (ISBI 2017), Melbourne, VIC, Australia, 18–21 April 2017; pp. 937–941.
46. Canny, J. A computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **1986**, *PAMI-8*, 679–698. [[CrossRef](#)]
47. Lowe, D.G. Distinctive Image Features from Scale-Invariant Keypoints. *Int. J. Comput. Vis.* **2004**, *60*, 91–110. [[CrossRef](#)]
48. Kenney, C.S.; Zuliani, M.; Manjunath, B.S. An axiomatic approach to corner detection. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA, 20–25 June 2005; Volume 1, pp. 191–197. [[CrossRef](#)]
49. Harris, C.; Stephens, M. A combined corner and edge detector. In Proceedings of the Alvey Vision Conference, Manchester, UK, 31 August–2 September 1988.
50. Jianbo Shi.; Tomasi. Good features to track. In Proceedings of the 1994 Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 21–23 June 1994; pp. 593–600. [[CrossRef](#)]
51. Rong, W.; Li, Z.; Zhang, W.; Sun, L. An improved Canny edge detection algorithm. In Proceedings of the 2014 IEEE International Conference on Mechatronics and Automation, Tianjin, China, 3–6 August 2014; pp. 577–582. [[CrossRef](#)]
52. Abdusalomov, A.; Mukhiddinov, M.; Djuraev, O.; Khamdamov, U.; Whangbo, T.K. Automatic Salient Object Extraction Based on Locally Adaptive Thresholding to Generate Tactile Graphics. *Appl. Sci.* **2020**, *10*, 3350. [[CrossRef](#)]
53. Mehrotra, R.; Namuduri, K.; Ranganathan, N. Gabor filter-based edge detection. *Pattern Recognit.* **1992**, *25*, 1479–1494. [[CrossRef](#)]
54. Haralick, R.M.; Shanmugam, K.; Dinstein, I. Textural Features for Image Classification. *IEEE Trans. Syst. Man, Cybern.* **1973**, *SMC-3*, 610–621. [[CrossRef](#)]
55. He, D.C.; Wang, L. Texture Unit, Texture Spectrum, And Texture Analysis. *IEEE Trans. Geosci. Remote Sens.* **1990**, *28*, 509–512.
56. Galloway, M.M. Texture analysis using gray level run lengths. *Comput. Graph. Image Process.* **1975**, *4*, 172–179. [[CrossRef](#)]
57. Tamura, H.; Mori, S.; Yamawaki, T. Textural Features Corresponding to Visual Perception. *IEEE Trans. Syst. Man Cybern.* **1978**, *8*, 460–473. [[CrossRef](#)]
58. Laws, K.I. Rapid texture identification. In Proceedings of the International Society for Optics and Photonics, SPIE, San Diego, CA, USA, 29 July–1 August 1980; Volume 238, pp. 376–381. [[CrossRef](#)]
59. Baraldi, A.; Pannigiani, F. An investigation of the textural characteristics associated with gray level cooccurrence matrix statistical parameters. *IEEE Trans. Geosci. Remote Sens.* **1995**, *33*, 293–304. [[CrossRef](#)]
60. Costa, A.F.; Humpire-Mamani, G.; Traina, A.J.M. An efficient algorithm for fractal analysis of textures. In Proceedings of the 2012 25th SIBGRAPI Conference on Graphics, Patterns and Images, Ouro Preto, Brazil, 22–25 August 2012; pp. 39–46. [[CrossRef](#)]
61. Zhou, J.; Fang, X.; Tao, L. A sparse analysis window for discrete Gabor transform. *Circuits Syst. Signal Process.* **2017**, *36*, 4161–4180. [[CrossRef](#)]
62. Stork, D.G.; Wilson, H.R. Do Gabor functions provide appropriate descriptions of visual cortical receptive fields? *JOSA A* **1990**, *7*, 1362–1373. [[CrossRef](#)] [[PubMed](#)]
63. Pavlovičová, J.; Oravec, M.; Osadský, M. An application of Gabor filters for texture classification. In Proceedings of the ELMAR-2010, Zadar, Croatia, 15–17 September 2010; pp. 23–26.
64. Teuner, A.; Pichler, O.; Hosticka, B.J. Unsupervised texture segmentation of images using tuned matched Gabor filters. *IEEE Trans. Image Process.* **1995**, *4*, 863–870. [[CrossRef](#)] [[PubMed](#)]
65. Moraru, L.; Obreja, C.D.; Dey, N.; Ashour, A.S. Dempster-shafer fusion for effective retinal vessels' diameter measurement. In *Soft Computing Based Medical Image Analysis*; Elsevier: Amsterdam, The Netherlands, 2018; pp. 149–160.
66. Öztürk, Ş.; Akdemir, B. Application of feature extraction and classification methods for histopathological image using GLCM, LBP, LBGLCM, GLRLM and SFTA. *Procedia Comput. Sci.* **2018**, *132*, 40–46. [[CrossRef](#)]
67. Zhang, X.; Cui, J.; Wang, W.; Lin, C. A study for texture feature extraction of high-resolution satellite images based on a direction measure and gray level co-occurrence matrix fusion algorithm. *Sensors* **2017**, *17*, 1474. [[CrossRef](#)]
68. Tabatabaei, S.M.; Chalechale, A. Noise-tolerant texture feature extraction through directional thresholded local binary pattern. *Vis. Comput.* **2019**, *36*, 967–987. [[CrossRef](#)]
69. Mehta, R.; Egiazarian, K.O. Rotated Local Binary Pattern (RLBP): Rotation invariant texture descriptor. In Proceedings of the 2nd International Conference on Pattern Recognition Applications and Methods, ICPRAM 2013, Barcelona, Spain, 15–18 February 2013; pp. 497–502.
70. Ojala, T.; Pietikainen, M.; Maenpaa, T. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Trans. Pattern Anal. Mach. Intell.* **2002**, *24*, 971–987. [[CrossRef](#)]
71. Sharma, M.; Ghosh, H. Histogram of gradient magnitudes: A rotation invariant texture-descriptor. In Proceedings of the 2015 IEEE International Conference on Image Processing (ICIP), Quebec, QC, Canada, 27–30 September 2015; pp. 4614–4618.
72. Pramerdorfer, C.; Kampel, M. A dataset for computer-vision-based PCB analysis. In Proceedings of the 2015 14th IAPR International Conference on Machine Vision Applications (MVA), Tokyo, Japan, 18–22 May 2015; pp. 378–381. [[CrossRef](#)]
73. Mahalingam, G.; Gay, K.M.; Ricanek, K. PCB-METAL: A PCB image dataset for advanced computer vision machine learning component analysis. In Proceedings of the 2019 16th International Conference on Machine Vision Applications (MVA), Tokyo, Japan, 27–31 May 2019; pp. 1–5. [[CrossRef](#)]

74. Li, D.; Xu, L.; Ran, G.; Guo, Z. Computer Vision Based Research on PCB Recognition Using SSD Neural Network. *J. Phys.* **2021**, *1815*, 012005. [[CrossRef](#)]
75. Chen, T.Q.; Zhang, J.; Zhou, Y.; Murphey, Y.L. A smart machine vision system for PCB inspection. In Proceedings of the 14th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems, IEA/AIE 2001, Budapest, Hungary, 4–7 June 2001.
76. Harshitha, R.; Apoorva, G.C.; Ashwini, M.C.; Kusuma, T.S. Components Free Electronic Board Defect Detection and Classification Using Image Processing Technique. *Int. J. Eng. Res. Technol.* **2018**, *6*, 1–6.