DEGREE PROJECT IN TECHNOLOGY,
SECOND CYCLE, 30 CREDITS
STOCKHOLM, SWEDEN 2020

Machine Learning and Computer Vision for PCB Verification

CHEN YANG

KTH ROYAL INSTITUTE OF TECHNOLOGY
ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

Authors

Chen Yang <cheyan@kth.se>
Information and Communication Technology
KTH Royal Institute of Technology

Place for Project

Ericsson
Mobilvägen 12, Lund, Sweden

Examiner

Markus Flierl
KTH Royal Institute of Technology, Stockholm, Sweden

Supervisor

Raihan Rafique
Ericsson, Lund, Sweden

Abstract

Digitizing printed circuit boards (PCB) from images with computer science techniques is efficient in analyzing the PCB circuit. This automatic optic processing could help electronic engineers have a faster and more in-depth insight into complex multi-layer PCB. This automatic optic processing could help electronic engineers have a faster and more in-depth insight of complex multi-layer PCB.

In this thesis, multiple machine learning and computer vision methods for extracting PCB circuits are investigated, designed, and tested with real-world PCB data. PCB image dataset is collected by professional de-layer engineers, that consist of every layer of PCB and X-ray 3D models of the whole PCB. Region of interest (RoI) cropping and image alignment are applied firstly as in the pre-process stage. Detection and localization of electronic components are implemented with deep learning networks (Faster R-CNN), unsupervised machine learning clustering (XOR-based K-means), and multiple template matching, their accuracy results are 71.2%, 82.3% and 96.5%, respectively. For the multi-layer circuit extraction, the metallic print circuit is segmented in YCbCr color space, then the connection of every circuit net is obtained.

Keywords

PCB, Machine learning, Computer vision, Circuit extraction

Abstract

Digitalisering av tryckta kretskort (PCB) från bilder med datavetenskapstekniker är effektivt för att analysera PCB: s kretsar. Denna automatiska optiska bearbetning kan hjälpa elektroniska ingenjörer att få en snabbare och mer djupgående inblick i komplexa flerlagers PCB.

I denna avhandling undersöks, designas och testas flera maskininlärnings- och datorvisionsmetoder för att extrahera PCB-kretsar med verkliga PCB-data. PCB-bilddataset samlas av professionella de-layer-ingenjörer, som består av varje lager av PCB och röntgen 3D-modeller av hela PCB. Beskärning av region av intresse (RoI) och bildjustering tillämpas först som i förprocessstadiet. Upptäckt och lokalisering av elektroniska komponenter implementeras med djupinlärningsnätverk (Faster R-CNN), utan tillsyn av maskininlärningskluster (XOR-based K-means) och flera mallmatchningar. För extraktion med flera lager kretsar är den metalliska utskriftskretsen segmenterad i YCbCr-färgutrymme, då erhålls anslutningen av varje kretsnät.

Nyckelord

PCB, maskininlärning, datorsyn, kretsextraktion

Acknowledgements

This master thesis project is hosted by Ericsson, which is a beautiful experience for me to work and study in this great place. The topic of PCB circuit extraction is very interesting because it could improve a lot for traditional industry with new artificial intelligence techniques.

First of all, I want to thank my supervisor Raihan Rafique, my co-supervisors Nicklas Ekstrand and Maomao Chen Larsson, for their constant guidance and instructive advice. Second, my sincere thanks go to my manager Lars Peter Künkel, for offering me this master thesis position and a lot of work support. I also would like to express my heartfelt gratitude to my colleagues Göran Pehrsson, Ola Guldbrand, Jens Kongstad, Swati Sawant, Häkan Bygdö and Muhammad Waqar Hussain, for their thoughtful kindness and useful suggestions. Besides, I want to thank Kulle Tomas from Sigma, for his perfect prototype work and help.

I am grateful to my examiner Markus Flierl to help improve my thesis work and give me insightful comments. Last but not least, I would like to appreciate my parents and friends for their encouragement and support, especially in this pandemic period.

Lund, November 2020
Chen Yang

Acronyms

PCB Printed Circuit Board
AOI Automated Optical Inspection
SURF Speeded Up Robust Features
RANSAC Random Sample Consensus
SIFT Scale-invariant Feature Transform
Faster R-CNN Faster Region-based Convolutional Neural Network
XOR Exclusive Or
CNN Convolutional Neural Network
IC Integrated Circuit
CT Computed Tomography
YOLO You Only Look Once
RPN Region Proposal Network
DBSCAN Density-Based Spatial Clustering of Applications with Noise
DOG Difference of Gaussian
GPP Group Pyramid Pooling
SSD Single Shot Detection
PE Printed Electronics
CCL Connected Component Labeling
RoI Region of Interest
SVD Singular-value Decomposition
TP True Positive
FP False Positive
TN True Negative
FN False Negative
mAP Mean Average Precision
NMS Non-Maximum Suppression

Contents

1 Introduction **1**

1.1 Background 1

1.2 Problem 2

1.3 Purpose 2

1.4 Goal 2

1.5 Methodology 3

1.6 Delimitations 3

1.7 Outline 4

2 Background **5**

2.1 Print Circuit Manufacture 5

2.1.1 PCB Structure 5

2.1.2 PCB Composition 6

2.2 Object Detection 8

2.2.1 Machine Learning for Object Detection 8

2.2.2 Image Processing for Object Detection 10

2.3 PCB Automated Optical Inspection 12

3 Method **13**

3.1 Data Acquisition 13

3.2 Image Pre-processing 15

3.2.1 Multi-layer Alignment 15

3.2.2 PCB RoI Segmentation 17

3.3 Component Detection with Faster R-CNN 19

3.3.1 Data Preparation 19

3.3.2 Network Design 20

3.3.3 Evaluation Metrics 22

3.4 Component Detection with K-Means Clustering 23
3.4.1 Localization of Image Difference 23
3.4.2 Clustering and Segmentation 24
3.4.3 Component Pad Detection 25
3.5 Component Detection with Template Matching 26
3.5.1 Multiple Template Matching 26
3.5.2 Image Correction 27
3.6 Metal Segmentation 28
3.6.1 Color Space Segmentation 28
3.6.2 Multi-channel Image Enhancement 29

4 Implementation 31
4.1 PCB Component Detection 31
4.1.1 K-means Detection 32
4.1.2 Faster R-CNN Detection 32
4.1.3 Multi-template Matching 35
4.2 PCB Inner Connection Extraction 35

5 Results and Evaluation 37
5.1 Faster R-CNN Detection 37
5.2 K-means Detection 41
5.3 Multi-template Matching 43
5.4 Inner Circuit Connection 44

6 Conclusions and Future Work 47
6.1 Conclusion 47
6.2 Limitation 48
6.3 Future Work 49

References 50

Chapter 1

Introduction

Machine learning powered image processing techniques have been applied for Printed Circuit Board (PCB) verification in recent years. In this master thesis, we investigate automatic computing solution of component detection and inner metal connection for our practical multi-layer PCB.

1.1 Background

Computer vision based inspection technology is widely applied in many industries, transforming the physical model to a digital representation [28]. Specifically, for the research and development of a PCB, an automatic computer vision system could help designers and engineers to accelerate and optimize their work. Meanwhile, it could be easier to find out faults, defects as well as hardware Trojans. Facing to hundreds of, even thousands of, components on a scaled PCB with millimeter or even micron level metal wires prints, it is hard for manual inspection.

With the development of image processing and machine learning in computer science, automatic recognition methods are available to help engineers in the PCB design and analysis[32]. For example, the Automated Optical Inspection (AOI) technique has been using to find out the defects of PCB [14]. For inspection of silicon-based chips, circuit netlist can be extracted with the electronic microscope and advanced detection algorithms [7].

1.2 Problem

This project's research questions can be summarized as follows: How to collect images for our industrial PCB and what condition and constraint should be followed? With limited practically real PCB data, how do we pre-process and labeling these images? How can we augment more data to enhance the machine learning model? What methods are most helpful to accelerate and improve the detection and classification processes? What structure of our neural network is most efficient, and how to evaluate it? The system robustness is another crucial figure to assess the entire algorithm.

1.3 Purpose

The purpose of this project is to investigate automatic methods in machine learning and image processing for circuit information extraction from physical print circuit boards, which should include components information, wires connectivity, and inter-layer relationship. It can help electronics engineers to boost and accelerate PCB inspection and verification processes. The thesis discusses both deep learning and classical image processing methods for electronic components detection; it studies multi-layer alignment and circuit connection; it illustrates the components detection and localization performance, the circuit of multi-layer PCB.

1.4 Goal

This project aims to extract the circuit information for PCB with high accuracy and low cost of time and computation. This project is divided into the following three sub-goals:

1. Designing algorithms for RoI cropping and multi-layer PCB alignment.
2. Implementing a deep learning algorithm for components detection and localization.
3. Designing classical image processing algorithms to enhance and complement components detection.
4. Designing an image segmentation method for inner layer metal detection.
5. Designing a process for extracting circuit element connection information.

6. Evaluating performance of component detection.

1.5 Methodology

There are three main parts to this project: the first part is PCB image pre-processing, in which stage, every layer of PCB is aligned, and RoIs are cropped out for the next process; the second part is electronic component detection and localization; the second part is inner layer metal segmentation and connection information extraction.

For the first part, Speeded Up Robust Features (SURF) [3] and Random Sample Consensus (RANSAC) [9] is used to find the homography matrix between two adjacent PCB layers. Scale-invariant Feature Transform (SIFT) is also attempted to feature points[21]. PCB Region of Interest (RoI) is cropped from the raw image by the color-indexed and edge-density segmentation.

For component detection and localization, we apply three different methods to ensure a high accuracy result for circuit analysis. The first method is Faster Region-based Convolutional Neural Network (Faster R-CNN) objection detection; the second is our innovative Exclusive Or (XOR) image based K-means [17] detection with a simple Convolutional Neural Network (CNN) classifier; The last process is multi-template matching as a manually controlled complement.

We utilize both standard digital image and X-ray slice for PCB inner circuit connection to enhance the conductive wire image. Connection information is extracted by overlapping aligned layers.

1.6 Delimitations

This thesis project research on PCB component detection and inner circuit connection separately does not study how to combine these two parts to generate a complete circuit. For component detection, we only classify components into *Resistor*, *Capacitor*, *Integrated Circuit (IC) Package*, and *Metal Pad*. We are no going to discuss the specific model of these components.

We concentrate on specific 7 PCBs brought from Sparkfun[1], which is the pilot research model for complex multi-layer PCBs. The motivation is that Sparkfun offers circuit

[1]https://www.sparkfun.com/

diagram and layout information of PCB, besides it is a economic option. The dataset for machine learning is prepared by ourselves; thus, it is limited and may not apply for other cases. Due to physical composition processing, PCB image quality is not ideal, which causes a specific problem of information loss.

1.7 Outline

This thesis is organized as following: Chapter 2 presents the background information and previous researches about electronic component detection and PCB netlist extraction; Chapter 3 introduces the techniques which are used in this project and gives solutions for our specific problems; Chapter 4 shows our workflows and the implementation of the project; In chapter 5, our algorithms' performance and results are evaluated for component detection and circuit extraction, respectively; Chapter 6 discusses this project's contribution and suggests its challenges and future work.

Chapter 2

Background

In this chapter, background knowledge and relative work are introduced for this thesis project. Section 2.1 gives a brief description of print circuit technology and PCB composition techniques. Section 2.2 describes the objection detection techniques in both machine learning and image processing fields. Meanwhile, reasons and explanations are given for techniques we applied in this project.

2.1 Print Circuit Manufacture

2.1.1 PCB Structure

A printed circuit board connects electronic components using conductive tracks, pads. In modern manufacturing, a PCB has multiple layers, and between every layer they are connected with conductive material touching straightly or by drills. On its surface, components are generally soldered onto the PCB to both electrically connect and mechanically fasten them to it.

PCB contains an insulating substrate for every layer, which usually is a fiberglass composite and is attached etched copper as the print circuit. As the conductive material, copper may be present on both sides of the substrate. Inner layers are manufactured by bonding multiple substrates together, as Figure 2.1.3 illustrated. We treat every copper layer is isolated, while it is connected by contacted copper of adjacent layers.

On the top of PCB, it is an insulating surface, it connects conductive copper circuit

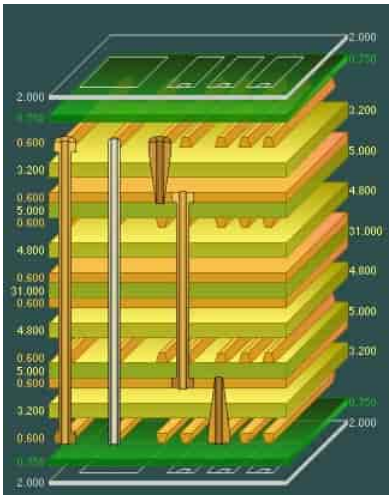


Figure 2.1.1: Multiple Layer Structure [31]

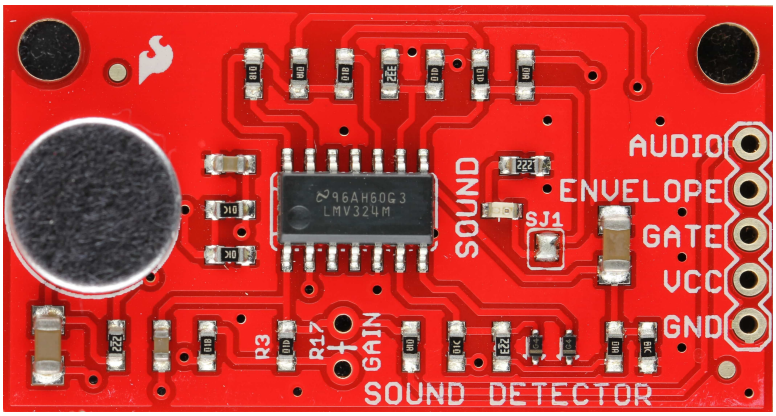


Figure 2.1.2: Soldermask and Silkscreen of PCB Surface

and components by soldermask, which defines pads for attaching components with inner copper. Silkscreen is another important of the surface, it is a mask of ink traces used to identify components, conductive points,logos and marks. In our project, the soldermask is unwanted background for component detection, while its contrast could help us to segment targets. The silkscreen causes troubles for solder and metal component detection, as you can see from Figure 2.1.2.

2.1.2 PCB Composition

For PCB composition, prototype designer usually uses physical methods to remove soldermask with tools, like sandpaper and fiberglass scratch brush. With the development in manufacturing, chemical removal and laser ablation techniques are widely used [13].

After removing soldermask, the next process is delayering, which is to access the

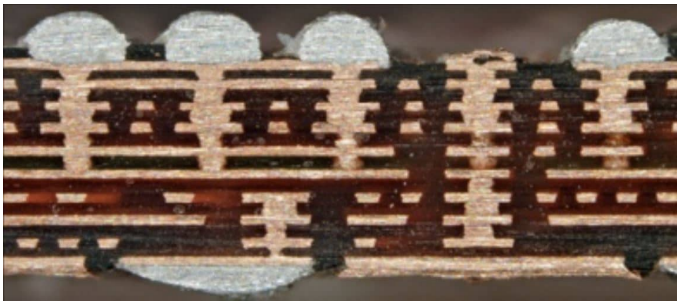
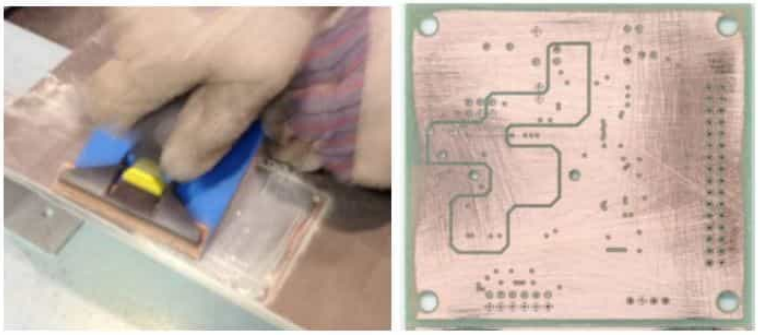
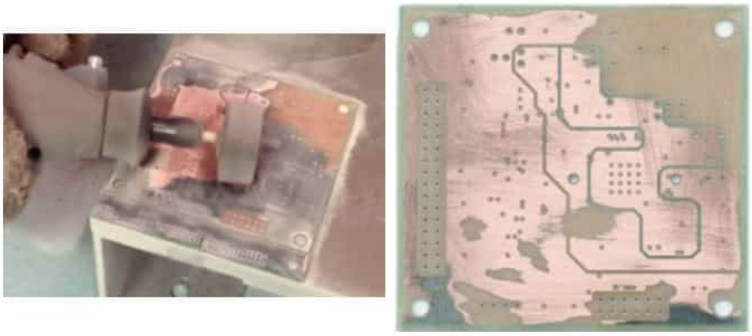


Figure 2.1.3: Cross-section of multi-layer PCB [13]

inner copper layers of a multi-layer PCB by way of physical, destructive delayering. Sandpaper is a traditional and reliable way to remove layers, as Figure 2.1.4 shown. Dremel tool is used in a variety of applications, including cutting, grinding, drilling, routing, polishing, and sanding.



(a) Sandpaper method



(b) Dremel method

Figure 2.1.4: Results of PCB Delayering [13]

As we can see from the result of physical delayering, there are a lot scratch and broken for PCB's copper integrity, which would lead difficulties to our project. In additional, a Computed Tomography (CT) method is applied by using X-ray scan to reconstruction 3D model. [16] introduced the mathematics principle to rebuild CT images. With

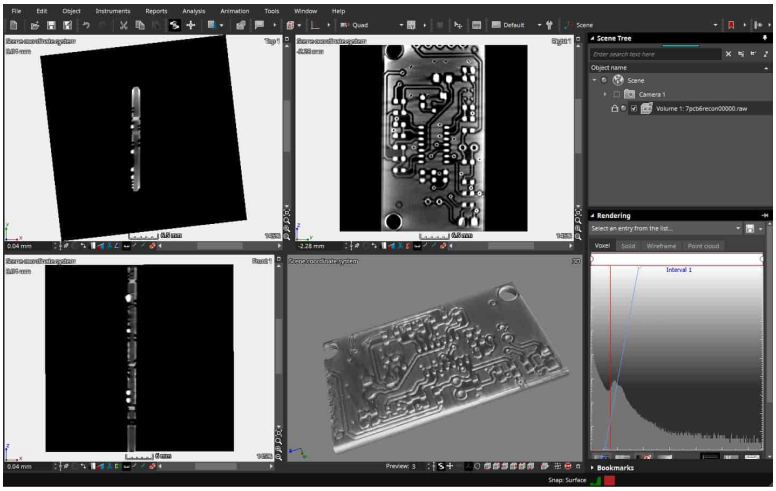


Figure 2.1.5: PCB 3D Model in VGStudio

advanced scanning machine and professional software, inner structure of copper can be visualized in a less lossy way. Figure 2.1.5 presents work interface of software **VGStuido** [1], which gives X, Y and Z cross-sectional views of a PCB.

2.2 Object Detection

2.2.1 Machine Learning for Object Detection

Supervised Machine Learning

Recently, Convolutional Neural Network (CNN) [18] makes a great improvement in computer version, this deep learning architecture performs excellently in object recognition. In CNN classification design, it mainly contain three layers: convolutional layer creates a feature map to predict the class probabilities for each feature by applying a filter that scans the whole image; pooling layer scales down the amount of information the convolutional layer; Finally, flattened matrix goes through a fully connected layer to classify the images.

Besides image target classification, target localization is required in a lot object detection tasks, especially in our PCB component project. R.Girshick et al. proposed R-CNN[11] in 2014, which exploits search selective method [30] to find out the RoIs in image and then put these part to the traditional convolution neural network. As the name shows, faster R-CNN is a improved algorithm from R-CNN, while there is another intermediate version called fast R-CNN [10]. In fast R-CNN, input image

[1]https://www.volumegraphics.com/en/products/vgstudio.html

is sent to a pre-trained CNN model, instead of applying $2,000$ times CNN to RoIs. The search selective algorithm is applied with feature map which is the result of CNN model. In the most advanced version, faster R-CNN uses Region Proposal Network (RPN) to replace the search selective process.

There are some other neural-network-based object detection approaches, like YOLO and RetinaNet. Redmon et al. [24] proposed You Only Look Once (YOLO) uses a single convolutional neural network for simultaneous prediction of location box and their class probabilities. YOLO is fast, because it treat detection task as a regression problem. Lin et al. proposed RetinaNet [19] is single network architecture which consists of a backbone network and two task-specific networks. The backbone network computes a feature map for the whole image, and the two mini-networks are in charge of object classification and bounding box location regression.

For this project specific topic, components are small objects on PCB, and component detection is the first step of circuit information extraction, the accuracy rate is more important than time cost. In [23](2020), it made an evaluation of different deep learning methods for small object detection. From that research we found that, YOLO is a fast method for small object; RetinaNet is a good one with balance of speed and accuracy;While faster R-CNN is well-performed in accuracy if ignoring the speed of processing, especially it could deal different datasets on many contexts of objects including multiple scale objects. Therefore, faster R-CNN is the best choice for electronic components detection in this thesis project.

Unsupervised Machine Learning

Anderberg proposed the K-means method [1] in 1973, which is a technique employed for partitioning a set of objects into k groups such that each group is homogeneous with respect to certain attributes based on the specific criterion. The main idea here is that the algorithm tries to build the groups so that two pixels from the same cluster are as similar as possible, while two points from two different clusters are as diverse as possible.

K-means is applied with Lloyd algorithm [20] with an iteration of clustering, it assigns data to k clusters, which are defined by their centroids. The algorithm is working as follow process:

1. Select k points from the data as the initial cluster centers;

2. Calculate the distance of each cluster object to the cluster center for dividing;
3. Calculate each cluster center again;
4. Calculate the measurement function. If the method reaches the maximum number of iterations, it will stop, otherwise, it will continue the operation.

Density-Based Spatial Clustering of Applications with Noise (DBSCAN), introduced by Ester et al. [8], is density based clustering technique. DBSCAN belongs to the third category of clustering methods and assumes that clusters are areas with high density in the data space.

Compared with non-density based clustering methods, the DBSCAN algorithm performs excellently for detecting different shapes and sizes objects. Besides, when there is a considerable density without any preliminary information about the groups present in the data set, DBSCAN can find these clusters well [29].

For our PCB component detection task, K-means clustering is based on color feature, while the DBSCAN utilizes the density of pixel data distribution. DBSCAN may bring out precise detection result, while it requires a large computational cost and memory. In specific case, K-means clustering could ensure a robust and reliable performance and it computes faster with lower cost.

2.2.2 Image Processing for Object Detection

Image Registration

Scale-invariant feature transform (SIFT) was proposed by Lowe [21] in 1999. The SIFT algorithm mainly has five steps:

1. Scale-space extreme detection;
2. Accurate key points localisation;
3. Orientation assignment;
4. Generation of key points descriptors;
5. SIFT key points matching.

SIFT uses Difference of Gaussian (DOG) to approximate a blob detector, which can be obtained by subtracting two images blurred by Gaussian kernel with different σ.

This process is done for different octaves of the image in the Gaussian pyramid. Then comparisons are made between one pixel and its eight neighbors in the current layer as well as 9 pixels in the above and previous layers. Local extremes are selected as potential interest points.

Besides, speed up and robust feature (SURF) is an improvement of SIFT, which is less complex with faster computation. SURF uses an integer approximation of the determinant of Hessian blob detector, which can be computed with 3 integer operations using a precomputed integral image. Its feature descriptor is based on the sum of the Haar wavelet response around the point of interest.[3]

SIFT and SURF both denote how to calculate correspondence between images. In particular, SURF could compute faster, while SIFT shows a stronger and more precise feature point detection in image alignment.

Template Matching

Template-based approaches allow the computation of the most probable positions of a single template image within a larger image with negligible manual annotation and at minimal computational cost [4].

However, using a single template, the detection capacity is limited, as the algorithm searches for a single intensity pattern which might not generalize well to objects with different perspectives or characteristics.

In [27], it proposed a multi-template matching method for microscopy image's object detection and localization. For multi-template matching, it searches by translation of template. The manually prepared dataset provides a set of templates to be searched (additional object perspectives, scales). Then, simultaneous object-detection for different categories can be performed as an iteration operation above.

For our electronic component detection, most components in one class are totally same in the color and shape (like resistors, capacitors) , they are in a repetition form presented in a PCB image. Compared with complex feature-based matching methods, multiple template matching could compute faster and show a reliable performance.

2.3 PCB Automated Optical Inspection

Recently, there are many research in PCB automated optical inspection (AOI) using computer vision techniques. For object detection, Huang et al. [15] (2019) detected PCB defects by image difference comparison and convolutional neural network. They take picture of real-world PCB, and then create defects by *PhotoShop*. By image matching and comparing template and test (defect) images, defects location are founded. In final, CNN classifier is applied to recognize 6 different type defects.

Tang et al. [26] (2019) established a new dateset of synthetical PCB defects and proposed an advanced object detection neural network with Group Pyramid Pooling (GPP) module. Compared with Single Shot Detection (SSD), YOLO and Faster R-CNN, their proposed methods show a better performance in PCB defect detection. These researches use simulative PCB defect dataset, which is much easier in data cleaning and image processing. Cheoing et al. [5] (2019) applied convolutional neural network on real PCB components and defects. However, they only did classification for electronic components with CNN and ignored the localization.

For PCB inner circuit extraction, Erozan et al. [7] (2020) extracted netlist for synthetical Printed Electronics (PE). After detecting circuit patches, the Connected Component Labeling (CCL) algorithm is applied for automatical annotation in netlist extraction. Navid et al. [2] used nondestructive X-ray tomography of PCB to extract components and connections. Yin et al. [33] proposed a PCB inner metal segmentation method based on gradient, which is helpful to coupe difficult of thin wire segmentation.

Our thesis work deals with real-world PCB and focuses on the component detection and inner circuit connection. Compared with PCB defect, component detection has various and multiple targets on one image. These targets are located densely, challenging for supervised machine learning to localize and classify components accurately. Therefore, we are going to combine various detection methods to guarantee accuracy. Component-removed images could be as the template for with-component images. In this way, component location will be found by comparing the difference between test and template images. PCB's inner circuit could be extracted by segmenting conductive metal from X-ray slices. Besides, we can utilize the alignment and overlapping condition to investigate the connection between adjacent layers.

Chapter 3

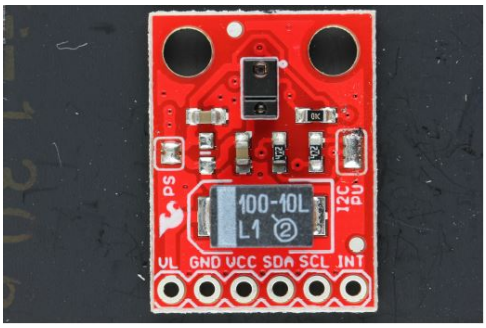
Method

This chapter describes the methods and techniques we used in the PCB circuit extraction project. We first introduce the data acquisition and image pre-processing, then present our supervised, unsupervised machine learning methods and template matching method for electronic component detection and localization. In the final, we introduce metal segmentation for inner layer inspection.

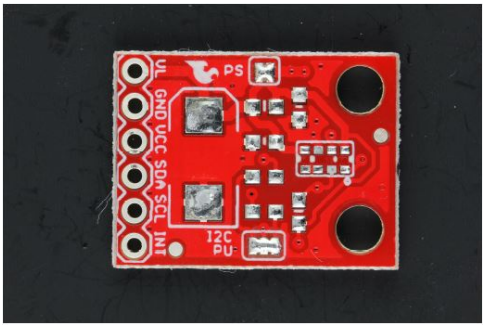
3.1 Data Acquisition

Data collection is the first step of our project. In this project, PCBs are public data that we bought online, and they are decomposed for every layer imaging by professional electronic engineers.

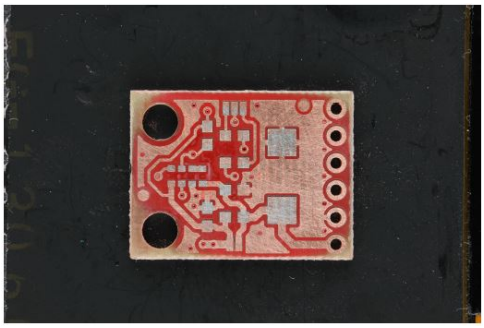
The image acquisition system is built with a high-resolution digital camera -Canon EOS 550D, desk-fixed lighting equipment. We output both *JPG* and *CR2* format images for 7 different sensor PCBs. *JPG* is the most commonly used digital image format with limited lossy compression and relatively good quality, *CR2* is Canon raw version format, which is larger data with more detail information for post-processes of different industry requirements. In this project, we choose *JPG* images as our basic data, because it contains sufficient information and is faster for computation. Each image is set up with 5184×3456 pixel-resolution, a shutter speed of $1/60$ second, an aperture of 8, and an ISO of 200.Figure 3.1.1 shows the PCB images we got with digital camera and delayering processing, where 3.1(c) and 3.1(d) are delayered PCB images.



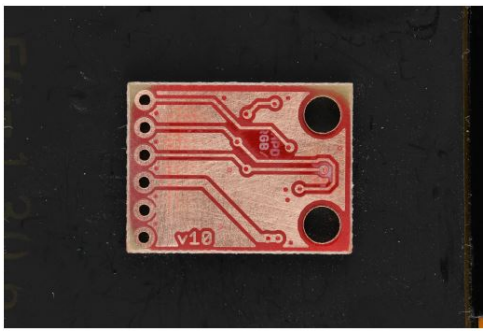
(a) Surface image with component



(b) Surface image component removed



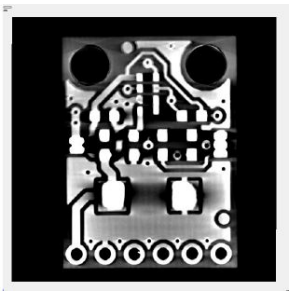
(c) Top layer color image



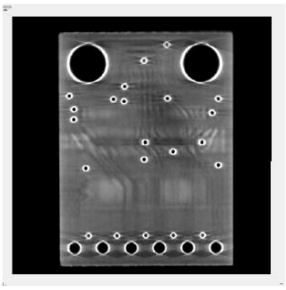
(d) Bottom layer color image

Figure 3.1.1: Normal PCB image collection

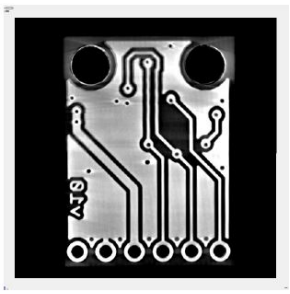
In order to have a better insight into multi-layer PCB, computed tomography (CT) technique is utilized to scan the X-ray 3D model of PCB. The CT machine we used is YXLON Cougar EVO [1], which could generate xxx models. It could clearly show the in-depth structure of the metal in every layer, which is the critical information to the representative inner circuit of PCB. The PCB 3D model is analyzed and extract image slices by software **VGSTUDIO**, which is an industrial application for CT data visualization. We extract slices from PCB's 3D model, which representative every layer information. The X-ray images for PCB layers are present as Figure 3.1.2.



(a) Top layers slice



(b) Middle layer slice



(c) Bottom layer slice

Figure 3.1.2: X-ray Slices

[1]https://www.yxlon.com/en/products/x-ray-and-ct-inspection-systems/yxlon-cougar-evo

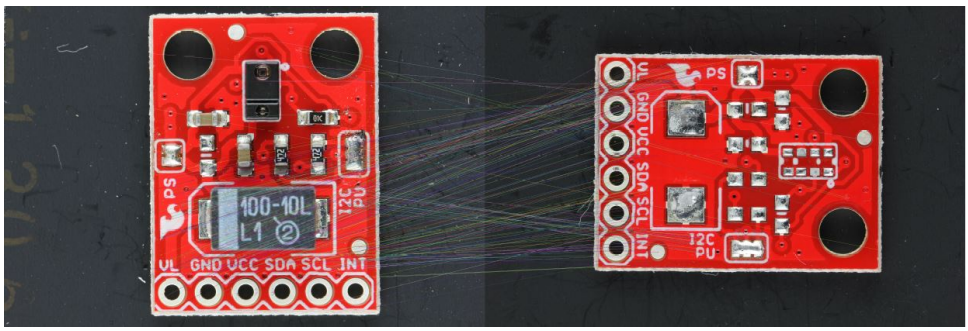


Figure 3.2.1: SIFT matches between two images

3.2 Image Pre-processing

In the section, we find the standard feature points between adjacent layer images to correct the pixel-to-pixel mapping relationship by using image registration techniques. Then we crop the RoI out for subsequent detection.

3.2.1 Multi-layer Alignment

Every PCB layer images are not precisely aligned due to the manual disassembly and imaging processes. Therefore, we need to correct all layer images to aligned positions. Image registration techniques, like SIFT, SURF, and RANSAC, are utilized to achieve this goal.

In the first step, it extracts the interest points and find the matches between consecutive pictures based on feature matching implementation. Image feature detection algorithm SIFT is applied in this section to seek scale invariant interest points. We then use **OpenCV** SIFT function to get the descriptors of interest points for later use in the matching step. Interest points between two images are matched based on Euclidian distance between their descriptors. We calculate both the best match and the second closest-match. Then we calculate the ratio, as function 3.1, to exclude mismatched points.

$$MatchRatio = \frac{bestDist}{secondBestDist} \tag{3.1}$$

After finishing all the steps above, we can get the coordinates of the matched points between two consecutive images, as Figure 3.2.1 shown.

After that, we then take one step further and try to find out the fundamental matrix

based on matched points. RANSAC is one of the most well-known methods to find fundamental matrix. It can exclude the outliers and calculate the matrix using only inliers, thus it can guarantee a higher accuracy. RANSAC can be done in following procedure:

1. Randomly select several matched points and assume these points are inliers.
2. Fit the model using these point pairs.
3. Calculate the number of points which are inliers. Sampson distance is used in this step
4. Repeat step 1 to 3 and update fundamental matrix for several iterations
5. Matrix with highest inlier ratio is selected as fundamental matrix

We set the image with components as template, and then align the component-removed image. The homograph transformation matrix H is compute by applying RANSAC filtered matches with Singular-value Decomposition (SVD) [12], as Equation 3.2 and 3.3 given.

$$[U, W, V] = svd(X) \tag{3.2}$$

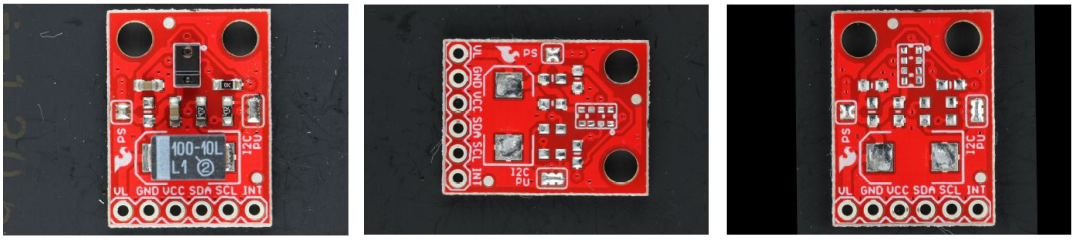
$$H = U * W^{1/2} \tag{3.3}$$

Here, X is the matrix of corresponding point matches, W is a diagonal matrix which reserves the one-dimensional eigenvalues, and U and V are orthogonal matrices in the left and right side of W, respectively.

After we computed the homograph matrix H, we could use perspective warp projection 3.4 to correct the test image [6], as Figure 3.2.2 illustrated.

$$I' = HX = \begin{bmatrix} h_{00} & h_{01} & h_{02} \\ h_{10} & h_{11} & h_{12} \\ h_{20} & h_{21} & h_{22} \end{bmatrix} I \tag{3.4}$$

Where, I is the test image, I' is the aligned test image with homography matrix H.



(a) Template image (b) Test image (c) Aligned test image

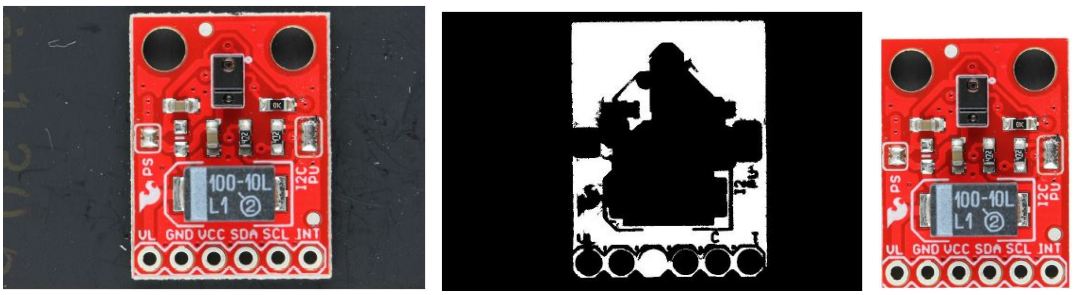
Figure 3.2.2: Alignment Result

3.2.2 PCB RoI Segmentation

Optical color images and X-ray images collected from 3.1 usually contain our PCB target as well as useless background information. In order to cropping region of interest out, we use color space segmentation, edge density segmentation to analyze the feature of PCB.

For the X-ray slice image, RoI is the metal board, which is with high illumination (energy) representative, and the target is usually in the center of the image. We compute the sum pixel value for each row/column from margins to the image center. Once the sum exceeds a threshold value, it would be defined as a bordering of RoI. The threshold is set as a fixed proportion of the average value of the whole image. While, for some PCB layers, they're maybe just a few metal dots that could not represent border information, we can solve this problem by choosing the most similar location and applying it to minorities.

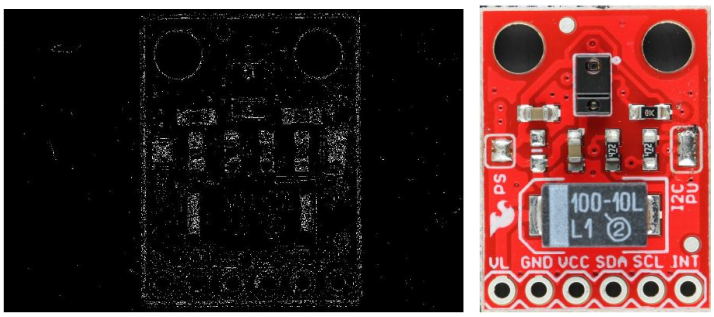
For a color image, we take the same idea as dealing with X-ray images, while we need to find out the PCB area's feature first to weaken the influence of background (usually is a plastic board in our project). Firstly, we convert the RGB image into an indexed color space. In indexed color space, masks are generated with different color based clusters. Therefore we can split RoI and background with limited index setting, as Figure3.2.3 shown.



(a) Original image (b) Color mask (c) Cropped RoI

Figure 3.2.3: Color Index Based RoI Segmentation

However, it is not easy to segment accurately when the background color is close to RoI. In a particular case, the background is usually the baseboard of a multi-layer PCB. It would make the indexed color segmentation failed. In order to prevent this situation, we set prior knowledge to test the result of previous segmentation. In this project, the width-height-ratio of PCB is the prior condition, which can be computed from X-ray segmentation results. Once the color segmentation does not satisfy this condition, the edge density segmentation would be implemented. PCB layer contains complicated circuits that are always highly dense edges. Thus we applied Sober edge detection for the entire image, then, as above processing, detection the large sum of every row and column to define the PCB RoI, the result of image 3.3(a) is illustrated as Figure 3.2.4 .



(a) Sobel edge detection (b) Cropped RoI

Figure 3.2.4: Sobel Edge Based RoI Segmentation

3.3 Component Detection with Faster R-CNN

3.3.1 Data Preparation

Components and pads were labeled separately with a powerful Python tool **LabelImg**[2]. For components, we divide them into 3 classes (resistor, capacitor and IC), labeled with box and class name. In order to improve the detection accuracy, we detect the metal pad with no-component PCB images, which manually removed electronic components before image acquisition. **LabelImg** exports *Pascal VOC* format label file, then we use an online platform **Roboflow**[3] for transforming raw images into a trained computer vision model. **Roboflow** can transform the images and label file to *TFRecord* and *CSV* file for Tensorflow object detection. When generating the dataset, we divide 70% into train dataset,15% into validation dataset, and rest into test dataset. The label image is shown as Figure3.3.1

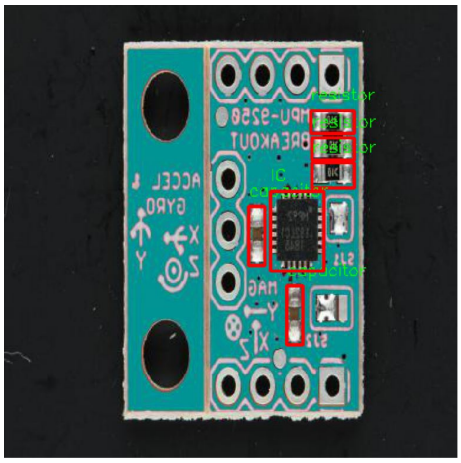


Figure 3.3.1: Labeled components in the train dataset

With limited physical PCB, data augmentation is necessary for a reliable deep learning system. Firstly, we change the board color, which is *red* for our real PCB, to *sky blue*, *grey*, *navy*, *indigo*, *green*, *tan* and *white*. This change is implemented by replacing the color value in the RGB space, as shown as Figure3.3.2.

[2]https://github.com/tzutalin/labelImg
[3]https://roboflow.ai/

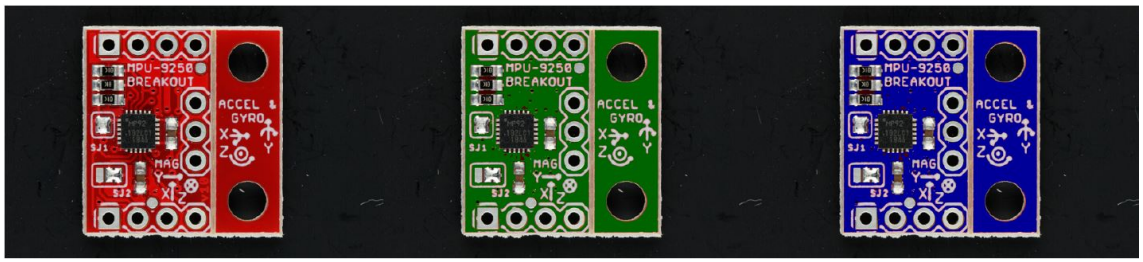


Figure 3.3.2: Synthesize new images by changing board color

Then, rotation is applied to create more, while practical experience, the imaging of PCB is always keep the rectangle outline, therefore we only rotate them with 90, 180 and 270 degrees. Furthermore, we also flip them both in vertical and horizon directions. In the last step, we add Gaussian noise for a random selection of generate images talked about above.

3.3.2 Network Design

Figure3.3.2 presents the complete architecture of faster R-CNN algorithm. First of all, the input images with bounding box labels information are passed through a pre-trained CNN model. For this step, we use transfer learning technology to boost and accelerate the training process with a limited dataset. Transfer learning provides it experienced weights that were trained with a huge dataset. We can treat the pre-trained model as an intermediate layer in our custom CNN model: the softmax layer connects with pre-trained CNN's output, then it exports the number of classes we defined. For example, VGG-16 is a popular pre-trained CNN model, its structure is as Figure 3.3.3 shown.

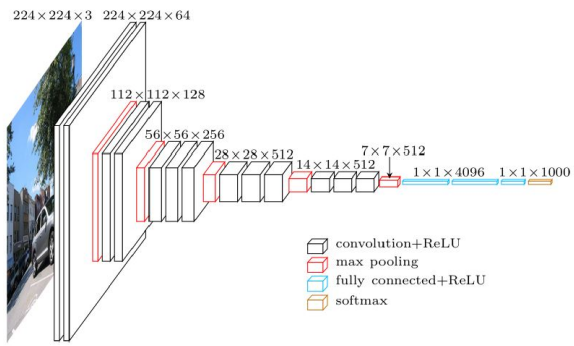


Figure 3.3.3: Architecture of VGG-16 [25]

In our case, the input of VGG-16 is $224 \times 224 \times 3$ tensor, and then it goes through these

convolutional layers. Finally, output a $256 \times 50 \times 37$ dimensions feature map. The transformation can be represented as Figure 3.3.4.

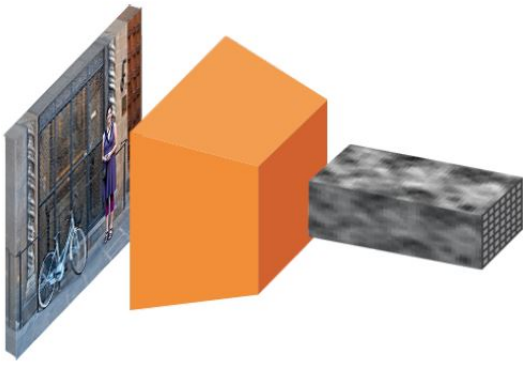


Figure 3.3.4: From image to convolutional feature map [25]

Secondly, RPN is computed with the output feature map of CNN. From our previous example, there are 50×37 points in each frame of VGG-16's feature map. We consider each point as an anchor, and each anchor has different shape ratios as well as size (we can define them). The RPN is implemented efficiently in a fully convolutional way, as Figure3.3.5. From the RNP's architecture, we can find that we can get two predictions for each anchor: one is for binary classification (foreground or background), the other is for bounding box regression (the actual object).

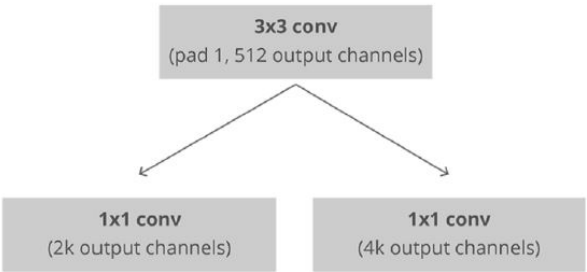


Figure 3.3.5: RPN's convolutional implementation

After the RPN step, we have some object proposals without a class labeled. Fixed-size convolutional feature maps are cropped out by RoI pooling as Figuer3.3.6. Then it is flattened with fully connected layers. Finally, the box location is fixed by a softmax function for classification and linear regression as Figure3.3.7, actually this step implements R-CNN with fixed-sized feature maps.

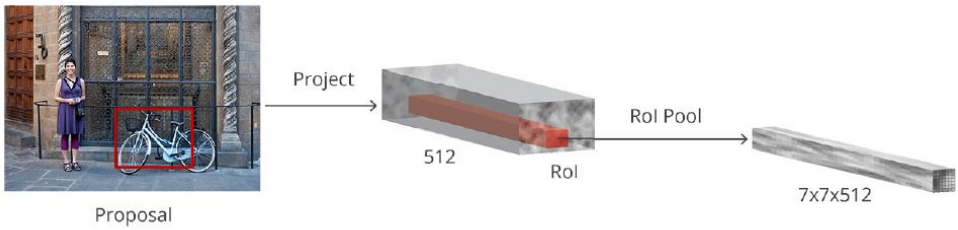


Figure 3.3.6: Region of Interest Pooling [25]

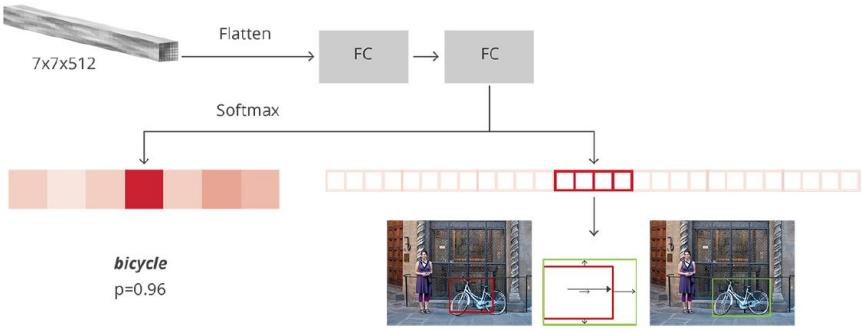


Figure 3.3.7: Architecture of R-CNN [25]

3.3.3 Evaluation Metrics

Significant figures are used for evaluation to our algorithms. Accuracy and Recall measures are basic indicator for machine learning, thay are defined by *True Positive (TP)*, *False Positive (FP)*, *True Negative (TN)* and *False Negative (FN)*, as formulas 3.5, 3.6 and 3.7

$$Accuracy = \frac{TP + TN}{TP + FN + FP + TN} \tag{3.5}$$

$$Recall = \frac{TP}{TP + FN} \tag{3.6}$$

$$Precision = \frac{TP}{TP + FP} \tag{3.7}$$

To evaluate test dataset, the Mean Average Precision (mAP) is introduced to measure the model performance. The Average Precision (AP) is defined by the precision rate

and the recall rate, as Equation 3.8:

$$AP = \int p(r)dr = \sum P(k)\Delta r(k) \tag{3.8}$$

Where, AP is the area under the precision-recall curve, *p(r)* is the precision at recall *r*, while *P(k)* and *r(k)* are the discrete expression.

The mAP is the primary indicator for evaluating model performance. For C different component classes, mAP represents the mean value of all classes result as Equation 3.9 shown.

$$mAP = \frac{\sum_{i=1}^{C} AP_i}{C} \tag{3.9}$$

3.4 Component Detection with K-Means Clustering

The supervised machine learning method needs to prepare a massive amount of data, which is vital to the learning model's performance. In our project, with limited PCB images, Faster R-CNN is challenging to detect all the components, especially IC packages have various forms and appearances. In this section, an unsupervised machine learning is introduced to detect the IC packages and other components by utilizing the difference between an image with components and an image without component.

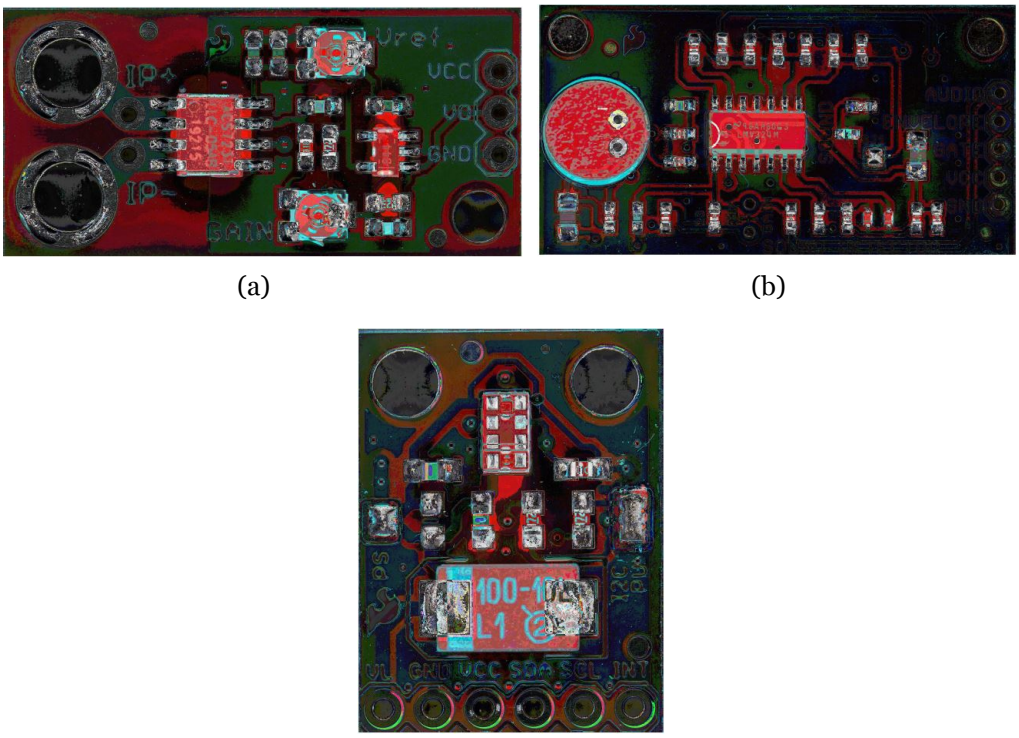
3.4.1 Localization of Image Difference

Component information is the difference between PCB surface image and removal-component image. We can find the location of the component by implementing the XOR operation with these two images. Before the XOR operation, image registration in 3.2.1 is applied to ensure they are pixel-to-pixel mapping alignment. The XOR operation is given in Equation 3.10.

$$dst(I) = src1(I) \oplus src2(I) \tag{3.10}$$

Where $dst(I)$ is the difference result of XOR operation, $src1(I)$, and $src2(I)$ are the

image with components and image without component, respectively. As we can see from Figure 3.4.1, the resulting image highlights color value in the component location. We can then put the resulting image to an unsupervised machine learning cluster for IC package location segmentation.



(a) (b)

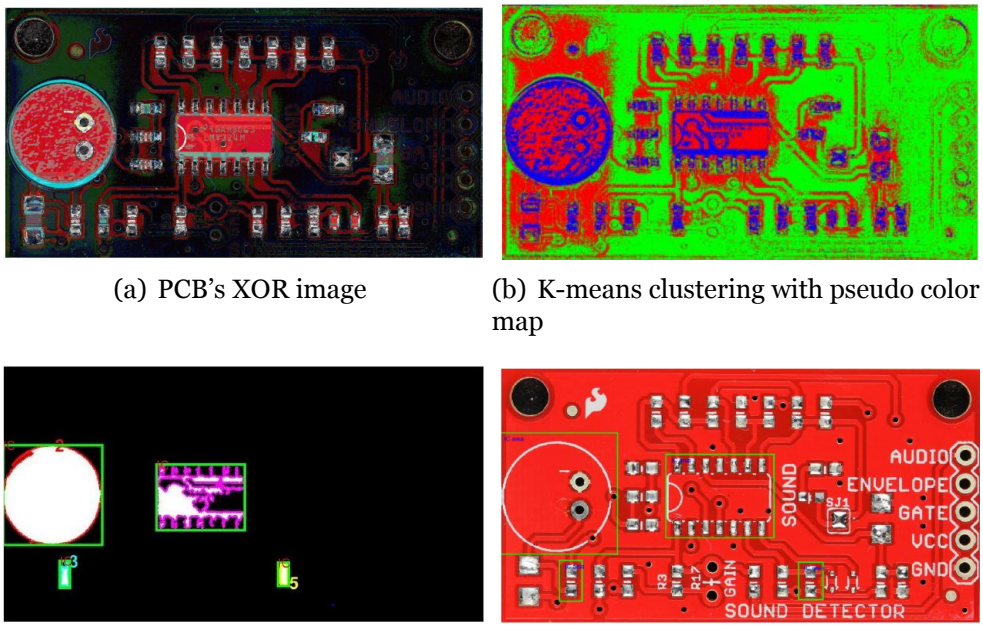
(c)

Figure 3.4.1: XOR Result

3.4.2 Clustering and Segmentation

As discussed in section2.2.1, the K-means clustering is utilized in the XOR image segmentation. K-means clustering would assign every pixel point in an image to precisely one of k possible clusters based on its color value similarity.

There is an example for our IC package location task; we applied K-means to cluster the image into three masks (k=3). As we can see from Figure 3.4.2, these significant areas are clustered with highest mean grey level, then we detect each connected area and crop it out with a square bounding box. These cropped areas are the so-called IC RoI. After we get the complex IC package's location information, we could dive in and identify its metal pins.



(a) PCB's XOR image

(b) K-means clustering with pseudo color map

(c) Component area in binary mask

(d) Component area in component-removed image

Figure 3.4.2: K-means Clustering Detection

3.4.3 Component Pad Detection

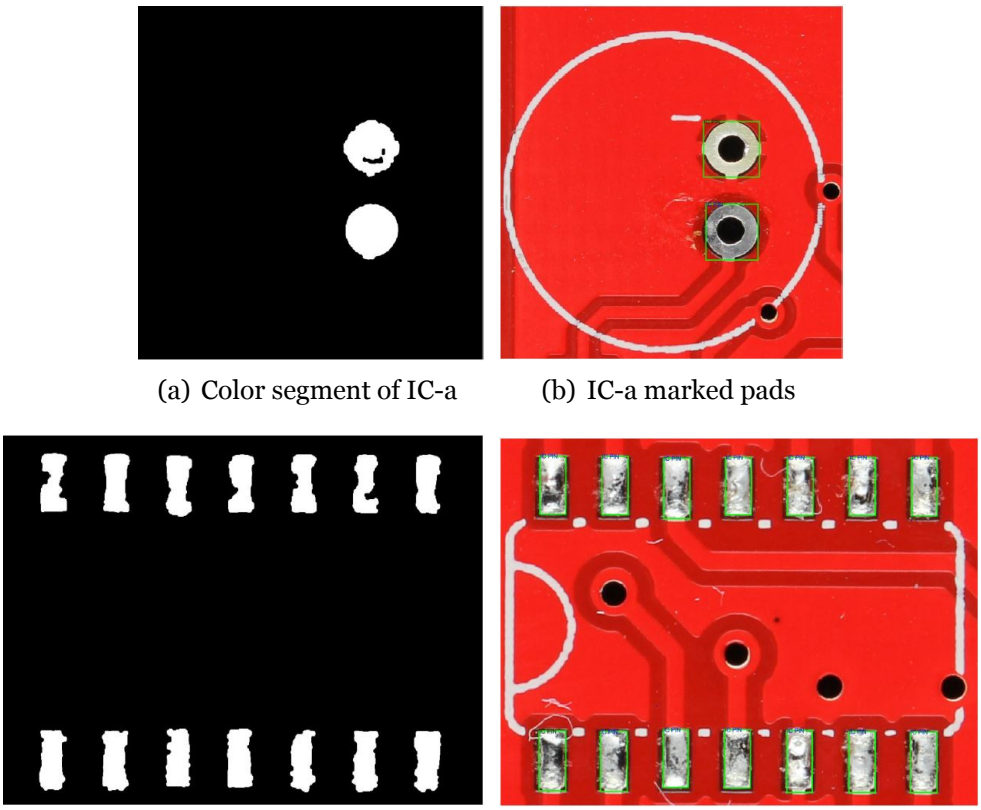
This step is going to extract the information of IC pads, which is an essential connector for IC in the entire circuit. Metal pin identification requires segment every single pad from IC component and finds its location. Once we got the IC package location, we can retrieve the image without component, then crop each IC area out. It is not a tricky task. Simple color space segmentation can remove the background easily. The experienced YCbCr value of metal pad (solder) is presented as Table 3.6.1, thus we set these thresholds in its color space.

Table 3.4.1: Solder Thresholds in YCbCr Color Space

Channel	Threshold Range
Y	[135, 255]
Cb	[120, 255]
Cr	[120, 150]

While the fixed threshold may not be robust and reliable, when the lighting condition and PCB material changed, it is difficult for this color space segmentation to identify metal pins.Besides, white silkscreen is another interference in color feature. Due to the significant difference between pads and background, the K-Means we introduced

above can solve this problem. In Figure 3.4.3, metal pins and other information are divided into two clusters.



(a) Color segment of IC-a (b) IC-a marked pads

(c) Color segment of IC-b (d) IC-b marked pads

Figure 3.4.3: Component Pad Detection

3.5 Component Detection with Template Matching

We presented two machine learning methods as above. Due to the limited dataset and certain image quality, they cannot guarantee a high detection accuracy. For the circuit extraction of real PCB, it needs to detect as many components as possible. Otherwise, the final circuit information is incomplete or even wrong. We apply image template matching to detect other tricky components, which is relay on manual preparation of component templates.

3.5.1 Multiple Template Matching

Firstly, in our project, we manually crop electronic components into a dataset and then make each template with rotations ($90, 180, 270$ degrees) and scale changes (re-scale to $0.5, 0.8, 1.2, 1.5, 2$ time) as data augmentation.

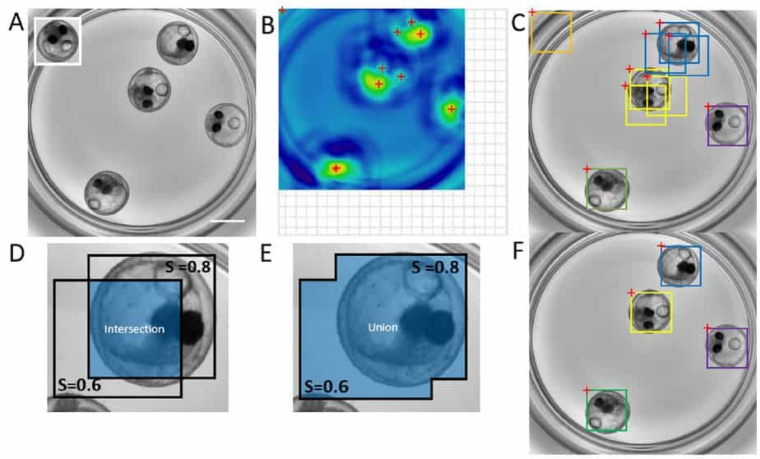


Figure 3.5.1: Overlapping of Multi-template Matching[27]

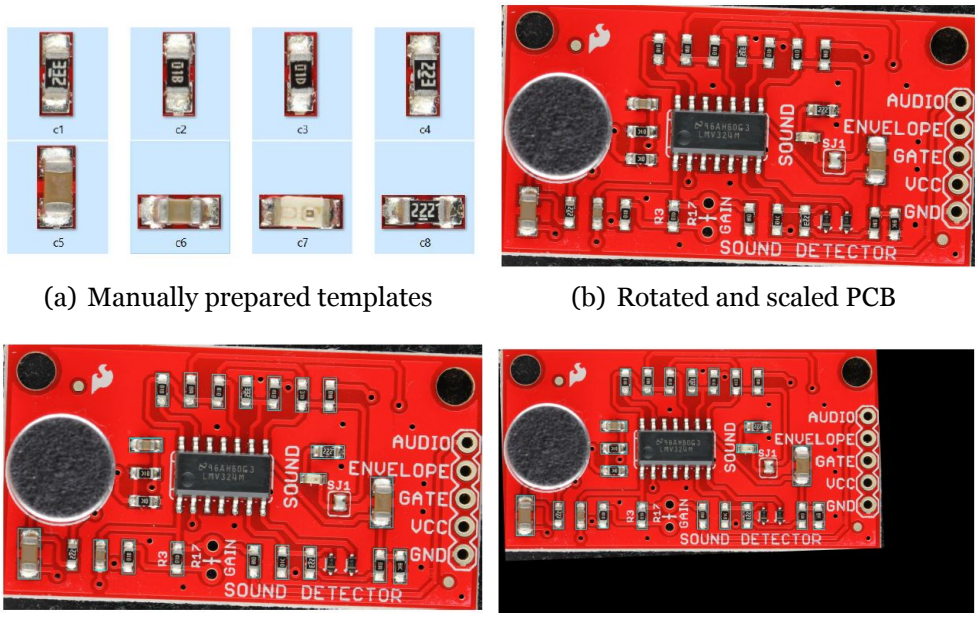
Secondly, it applied an iterative search algorithm with a rectangular sliding widow over the image to compute the correlation map. The similarity score of the candidate area (window) is calculated by comparing with templates in pixel intensity difference and correlations, where templates are the component classes in our project. In order to speed up this search process, the approximate position of component is known as a prior condition.

While the multi-template search result causes an overlapping problem, as Figure 3.5.1 illustrated. For this challenge, Non-Maximum Suppression (NMS) is applied based on a user-defined degree of overlap area based on the degree of overlap between neighboring bounding boxes. NMS computes the ratio between the IoU area for a pair: if they are highly overlap, the ratio will be close to 1; if the overlap is small, the ratio will tend to 0. In this way, overlapping of the same object can be removed efficiently.

3.5.2 Image Correction

For real PCB images, it may not be collected in a standard horizon/ vertical level as our template images. It will cause template matching failed due to a significant difference in the correlation map. We propose a correction method based on the template matching and image registration techniques. Firstly, we apply the multiple template matching to the original test image. The matching result may not good enough, while we only choose the component which got the highest similarity score. Then, we utilize the image registration method in 3.2.1 to compute the homograph transformation matrix H. The original image can be corrected with this matrix H by perspective warp projecting. There is an example Figure3.5.2 for this correction process. When image

corrected, we apply the multiple template matching again, and then it could output more accurate components detection and localization.



(a) Manually prepared templates (b) Rotated and scaled PCB

(c) Matching result of (b) (d) Matching result of corrected image

Figure 3.5.2: Multiple Component Templates Matching

3.6 Metal Segmentation

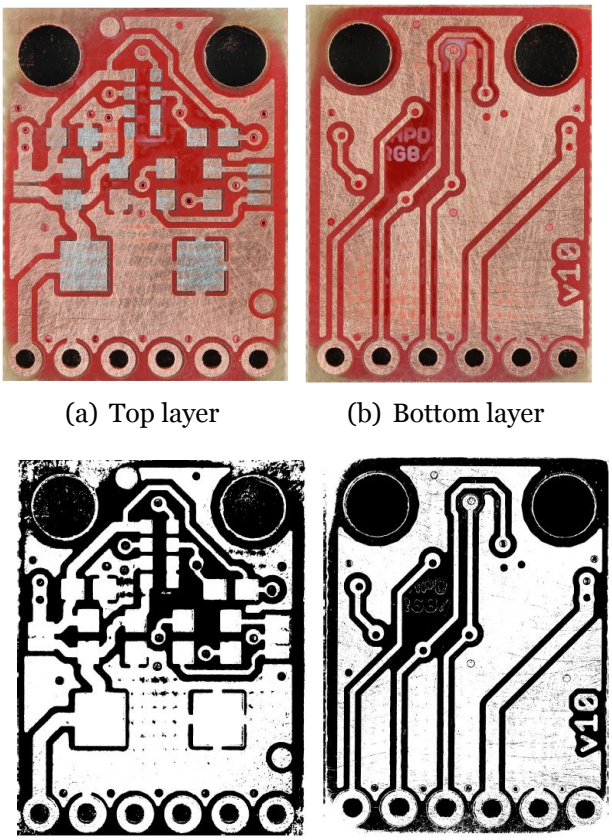
In this section, we focus on the information inside the PCB. For multi-layer PCB, each inner layer is metal printed as the conductive material, like copper. Therefore, metal segmentation is a critical process for circuit analysis and reconstruction. We exploit the colorful de-layer image and tomographic X-ray image to find metal areas.

3.6.1 Color Space Segmentation

Metal has a unique feature in color space, for our specific PCB de-layer images, the copper color with relative high illumination value. We investigate the main color spaces, like RGB, HSV, and YCbCr. Finally, we choose YCbCr color space to segment the metal part (copper) because its features are significant in illumination (Y) and red channel(Cr). The threshold values should be adjusted for special tasks.In this project, we get the experienced threshold setting from experiments, as Table 3.6.1 presented. The segmentation result is illustrated as Figure3.6.1.

Table 3.6.1: Solder Thresholds in YCbCr Color Space

Channel	Threshold Range
Y	[150, 255]
Cb	[0, 255]
Cr	[140, 255]



(a) Top layer (b) Bottom layer

(c) Binary mask of (a) (d) Binary mask of (b)

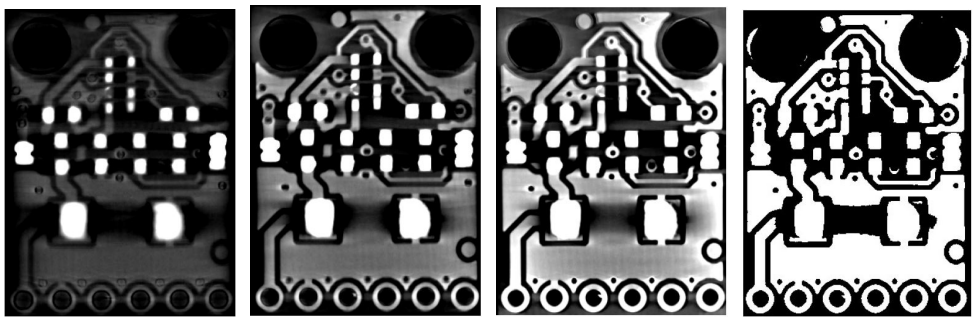
Figure 3.6.1: Color segmentation for Delayered Images

However, it is not perfect after color segmentation, much noise, holes and useless information are in the segmented image. We apply mathematical morphology to denoise and enhance the metal areas, the opening operation and closing operation are introduced in this step to reserve and enhance the metal areas.

3.6.2 Multi-channel Image Enhancement

Slices of CT model should be the ideal and reliable solution. Simultaneously, due to the resolution of CT machine and lossy reconstruction computation, some information is lost in our particular task. From Figure 3.6.2, we present three different depth X-ray

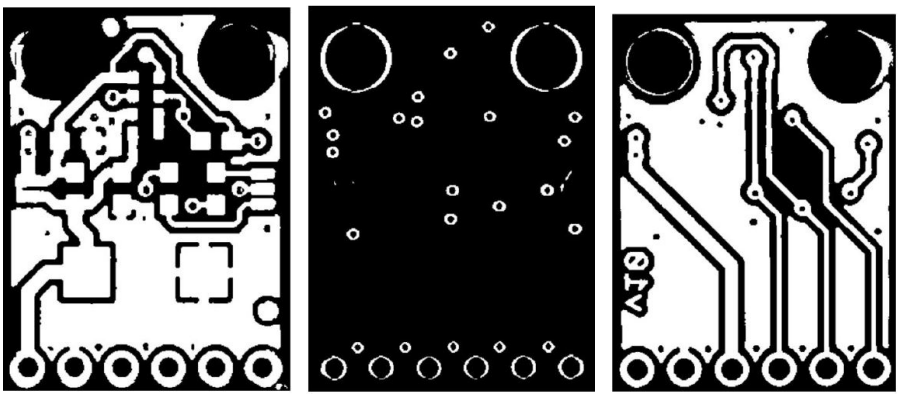
slices for one PCB layer. As you can compare with the manually de-layer image, some metal parts are lost in X-ray slices. Although we can easily find the metal parts in the de-layer image by the human visual system, it is hard for a computer to distinguish these thin lines and dots with limited resolution and noise. Besides, when engineers manually tear down these layers, there is always some scratch, which makes trouble in our image processing for metal segmentation.



(a) (b) (c) (d) Binary Fusion

Figure 3.6.2: X-ray Slice Enhancement

To solve this problem, we enhance the layer image quality by multiple channel fusion. It is a simple operation to utilize the resource we have. We resize the colorful de-layer image and apply color space segmentation. Then, we combine these images (3.6.1and 3.6.2) on a pixel level to enhance the metal region as Figure3.6.3.



(a) Top layer (b) Middle layer (c) Bottom layer

Figure 3.6.3: Multi-channel Enhancement Result

Chapter 4

Implementation

In this chapter, workflow diagrams are introduced to present the implementation of this project. There are two main parts of our work: component detection and inner information extraction. We explain our proposed methods with more detail, like K-means detection.

4.1 PCB Component Detection

Component detection workflow 4.1.1 presents the processes of surface layer detection. It directly explains how different component detection methods work together and enhance the database for machine learning's training dataset and matching templates. As we mentioned before, component detection is the first stage of circuit information extraction, and it is essential because it impacts the following processes. Therefore, we combine various methods to ensure the accuracy of component detection.

In the beginning, it need manually create a labeled component dataset to initiate Faster R-CNN detection. In component detection, the input images include a colorful surface image with component and component-removed surface image for the same PCB. After pre-processing, aligned PCB areas are put into proposed K-means detection. In K-means detection, it mainly finds the IC packages, which have significant features in XOR images), and successfully detected components cropped out and added to the database. Black rectangles will replace the detected components in the with-component image, then the input image for Faster R-CNN. The neural network is trained with prior data in the database, like K-means detection; the successful

detection results will be feedback to improve the database. In order to guarantee the final result, multi-template is applied as the last process, which uses templates from the database and may need to create templates for specific tasks manually.

4.1.1 K-means Detection

We proposed a K-means detection method, which utilizes the difference between images with components and with components removed. As Figure 4.1.2 shown, K-means clustering is applied with XOR images, like Figure 3.4.1 in Section 3.4. After clustering, detected areas are cropped out for further metal pads detection, where K-means clustering would be applied again or using color segmentation for fast detection. For each PCB, the parameter *k* is decided by grid search, and other parameters are also adjusted for specific image condition.

4.1.2 Faster R-CNN Detection

From the image acquisition system, we collected 7 PCB images with 5184×3456 resolution and then processed them as mentioned in 3.2.1.

We manually cropped every component out from original PCB images and then labeled with 4 different tags:*Resistor*, *Capacitor*, *IC* and *Metal Pad*. For each class, there are 30 samples, then divided into 25 and 5 for training and testing, respectively. Three different pre-trained CNN models (VGG-16, Inception V3, and Densent-169) are used separately for transfer learning due to such a limited dataset.

Input images are resized to $224 \times 224 \times 3$ for matching the convolutional layer. Meanwhile, we define the anchor for region searching, three anchor size for component detection are $[32, 64, 128]$, for metal pad detection are $[16, 32, 64]$.set *anchor ratio* as $1:1$, $\sqrt{2}:1$ and $1:\sqrt{2}$. Configure setting is optimized with the experience from [5]. We set 40 epochs and 1000 batches for each epoch.These parameters should adjust for specific image and target.

There are 2 kinds of loss functions: *Log* loss function for classification and *Smooth L1* loss function for bounding box regression in RPN, as Equation 4.1 and 4.2 explained, respectively.

$$L_{cls}(p_i, p_i^*) = -log[p_i^* p_i + (1 - p_i^*)(1 - p_i)] \tag{4.1}$$

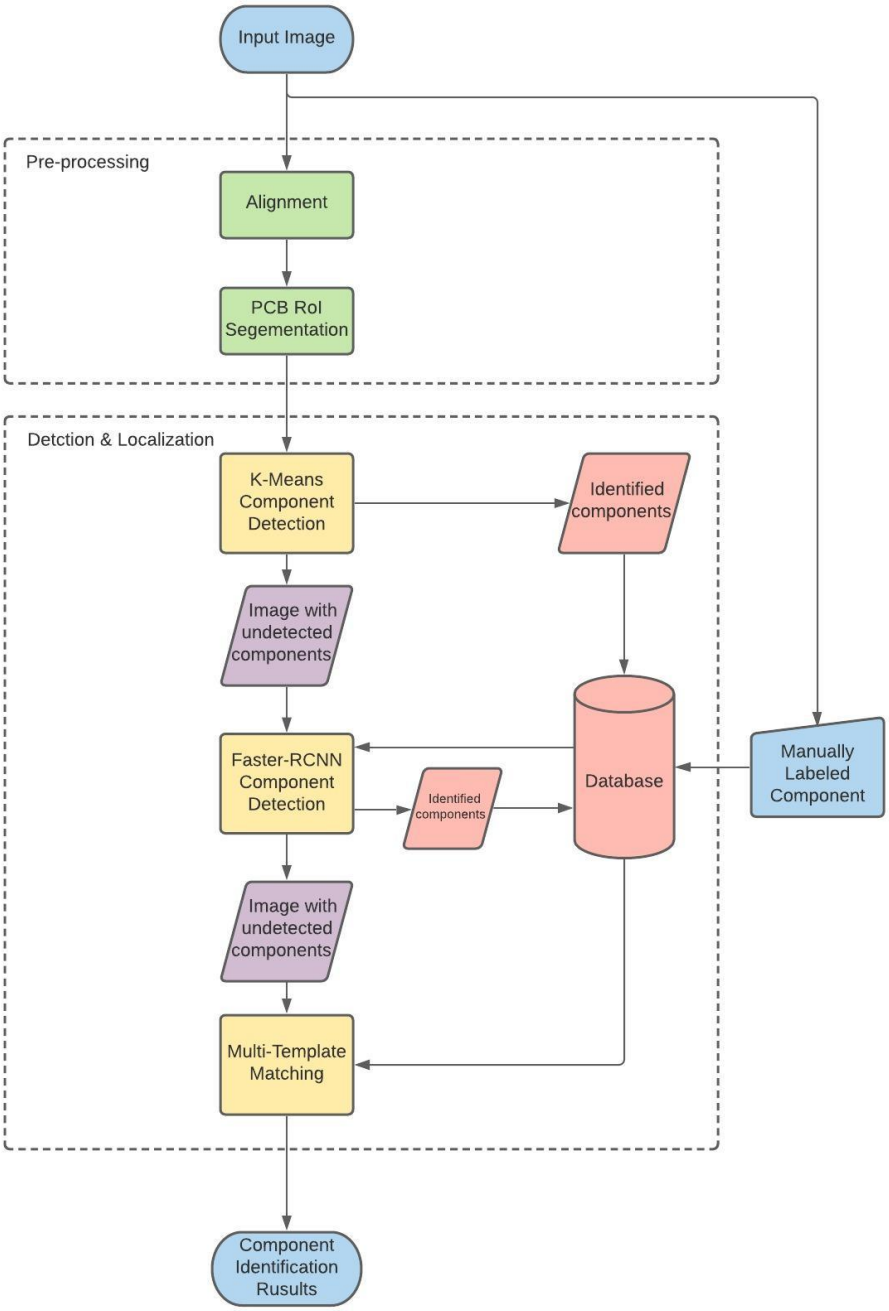


Figure 4.1.1: Component Detection Workflow

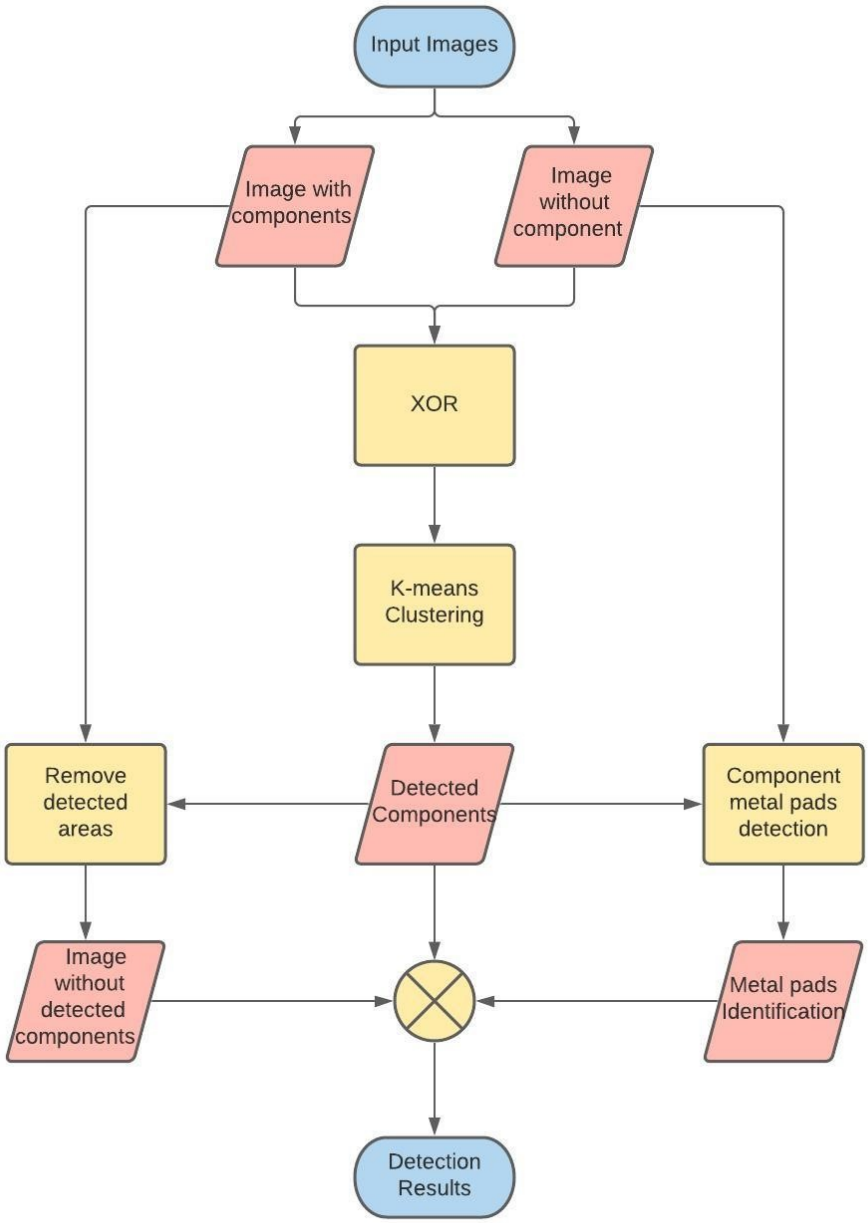


Figure 4.1.2: K-means Detection Workflow

Where, p_i is the target class, while p_i^* is the background class.

$$L_{reg}(t_i, t_i^*) = smooth_{L_1}(t_i - t_i^*) = \begin{cases} 0.5(t_i - t_i^*)^2 \times \frac{1}{\sigma^2} & if|t_i - t_i^*| < \frac{1}{\sigma^2} \\ |t_i - t_i^*| - 0.5 & otherwise \end{cases} \tag{4.2}$$

Where, t_i is a location vector, which represents the anchor's predicted offset in RPN; t_i^* is the same dimensional vector as t_i , and it is the practical offset compared with ground truth.

The total loss function is computed with L_{cls} and L_{reg}, as Equation 4.3 shown.

$$L_{total}(p_i, t_i) = \frac{1}{N_{cls}} \sum_i L_{cls}(p_i, p_i^*) + \lambda \frac{1}{N_{reg}} \sum_i p_i^* L_{reg}(t_i, t_i^*) \tag{4.3}$$

4.1.3 Multi-template Matching

As the final detector, multi-template matching is robust and reliable with manual complement. Workflow 4.2.1 explains how multi-template matching works with a rotated image. We apply the proposed matching method twice: the first matching tries to find one best candidate, which get the highest score of similarity mapping; then this candidate is cropped out to compute the homography transformation matrix as introduced in Section 3.2.1; once we got the matrix, the input image could be corrected to align our templates in same horizon level and same size scale; finally, we apply the multi-template matching again to the corrected image, it could improve the performance of detection. The threshold of correlation score is 0.70, which is a experienced parameter setting for our specific tasks.

4.2 PCB Inner Connection Extraction

Inner information extraction is different from component detection. We utilize image processing techniques to segment conductive copper parts in every inner layer in this stage, then find the connection between adjacent layers.

Workflow 4.2.2 shows three steps of this stage. Firstly, we segment metal part from a colorful delayered image and X-ray slices, respectively, where color space segmentation and adaptive binarization techniques are applied. The majority-voting image fusion combines images from a different channel, which enhances the layer's

metal(conductive) information. Finally, we compute the overlapping of neighboring layers, where parts have overlap means they are connected. In this way, multi-layer connection information could be extracted out.

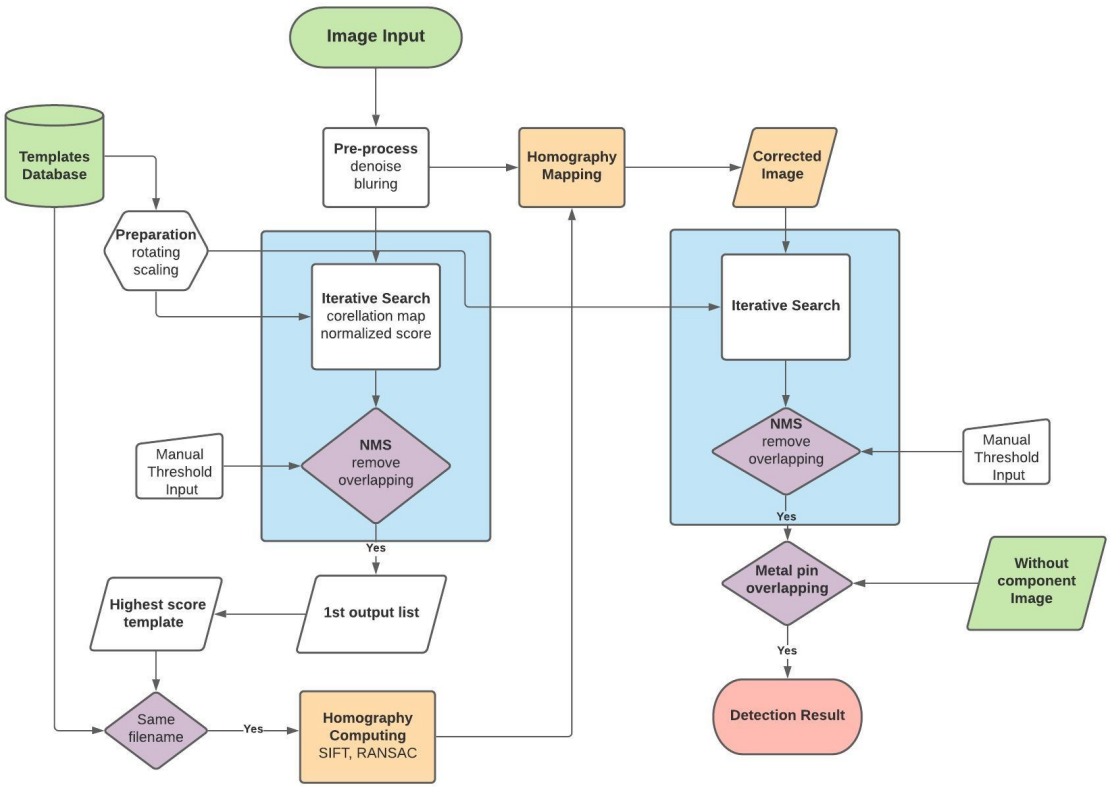


Figure 4.2.1: Multi-template Matching Workflow

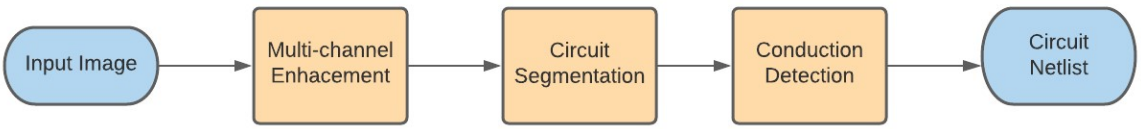


Figure 4.2.2: Inner Connection Extraction Workflow

Chapter 5

Results and Evaluation

In this chapter, the results and evaluation of the proposed methods are presented and discussed. For Faster R-CNN and K-means detection, we illustrate the learning analysis and result discussion. Multi-template matching and inner circuit connections are complements in this project. We present detection images as results of discussing their performance.

5.1 Faster R-CNN Detection

For this supervised machine learning method, we show the analysis of every process and give the performance comparison of different pre-training model.

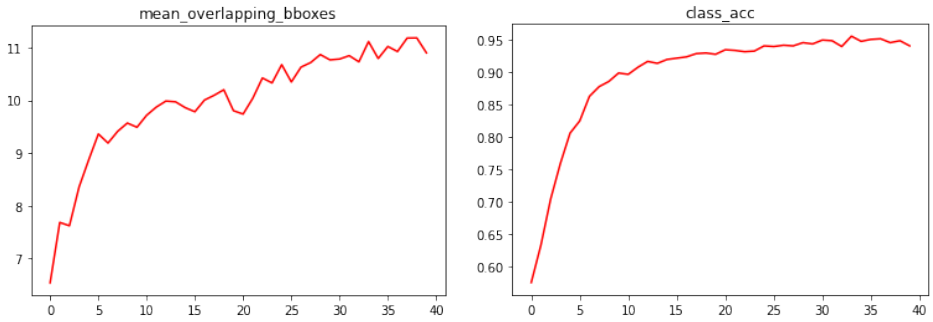
The limited number of PCB images is the biggest challenge for supervised machine learning. As introduced in 3.3.1, we applied data augmentation for 4 classes in this project, augmentation result is shown as Table 5.1.1.

Table 5.1.1: Sample Number of 4 Classes in Training Dataset

Class	Original Number	Augmented Number
Resistor	305	1220
Capacitor	248	992
IC	81	324
Metal Pads	1805	7220

In order to visualize the training process, we plot 3 line charts for totally 40 epochs: the mean number of bounding boxes from RPN overlapping ground truth boxes, classifier accuracy for bounding boxes from RPN (Figure 5.1.1), and loss line chart (Figure

5.1.2), which include: Loss of RPN classifier, Loss of RPN regression, Loss of Detector classifier, Loss Detector regression and Total Loss. We set 40 epochs and 1000 batches for each epoch.



(a) Mean number of overlapping (b) Classifier accuracy from RPN

Figure 5.1.1: Training Analysis for Component

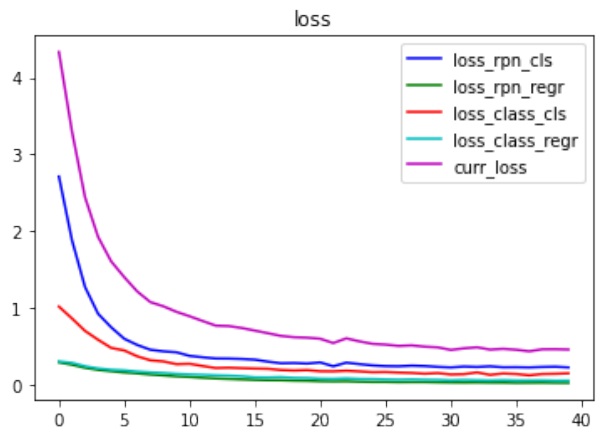
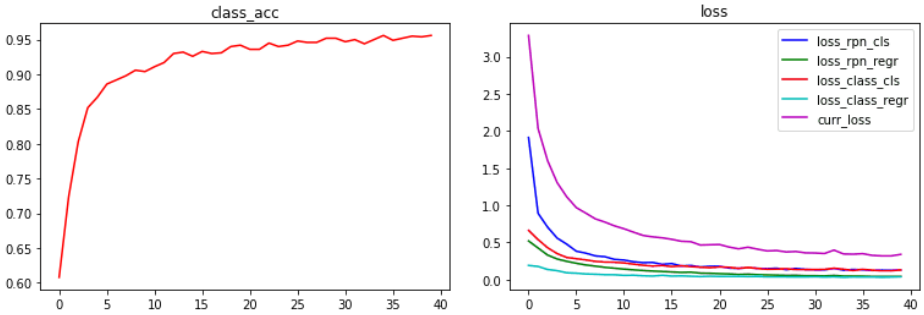


Figure 5.1.2: Total Training Loss for Component

As we can see from the above figures, these losses decreased fast, closing to zero in the first decade epochs. Simultaneously, the classifier's accuracy grows significantly in this period then increases slowly to 95% for the last 30 epochs. For the mean number of the overlapping box between RPN and ground truth, it grew up to 12 in 15 epochs then keeps a gentle fluctuation. The total loss goes down from 6.63 to 0.32, it is significant in the first 20 epochs. It is an acceptable result for limited data with $40k$ batches.

For metal pads detection, the line charts go in similar trends, and we put two key figures here, as Figure 5.1.3 shown.

As for the component localization and recognition, the mAP represents the performance for different models. In our experiment, we used three pre-trained



(a) Classifier Accuracy of Metal Pad (b) Total Loss for Metal Pad

Figure 5.1.3: Training Analysis for Metal Pad

models: *VGG-16*, *Inception V3* and *Densent-169*. These results are listed in Table 5.1.2. From the comparison, the Faster R-CNN algorithm with *VGG-16* is picked up with the best performance both for component data and metal pad data.

Table 5.1.2: Results of Faster R-CNN with Different Pre-trained Model

Dataset	Pre-trained model	mAP
Components	$VGG-16$	71.2%
Components	$InceptionV2$	63.4%
Components	$Densent-169$	62.7%
Metal Pad	$VGG-16$	69.4%
Metal Pad	$InceptionV2$	62.6%
Metal Pad	$Densent-169$	58.7%

The result of test process with *VGG-16* model is shown in the Table5.1.3, from which we can find that, *IC* has the highest AP(94.6%) while the *Capacitor* gets the worst score (52.1%).

Table 5.1.3: Result of Faster R-CNN Object Detection with VGG-16

Class	AP
Resistor	64.0%
Capacitor	52.1%
IC	94.6%
Metal Pad	78.4%

Although *IC* components are significantly different on different PCBs both in shape and size, which caused trouble for its detection, there is usually only one or two *IC* components on a PCB image. Besides, *IC* has a bigger size than other components; its convolutional feature must be quite distinguishable.

The size of *Capacitor* is similar to *Resistor*, and there are some unannotated components like Figure 5.1.6, it causes the low accuracy for our limited dataset. If we had more components in our training and testing sample, it would be improved a lot. The text example is labeled with class and accuracy, such as Figure 5.1.4 and 5.1.5.

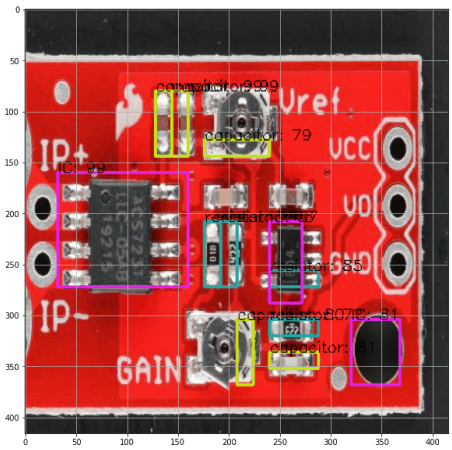


Figure 5.1.4: Example of test result for components

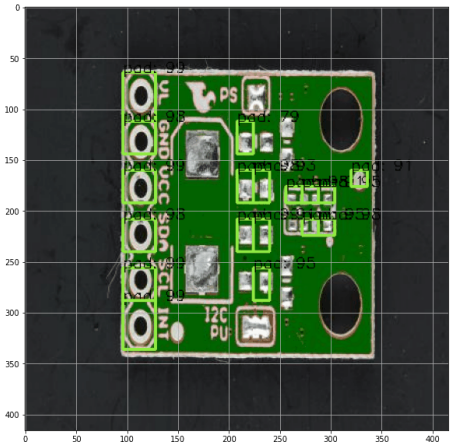


Figure 5.1.5: Example of test result for metal pads

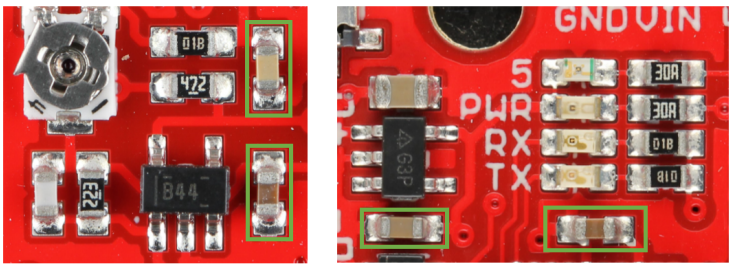
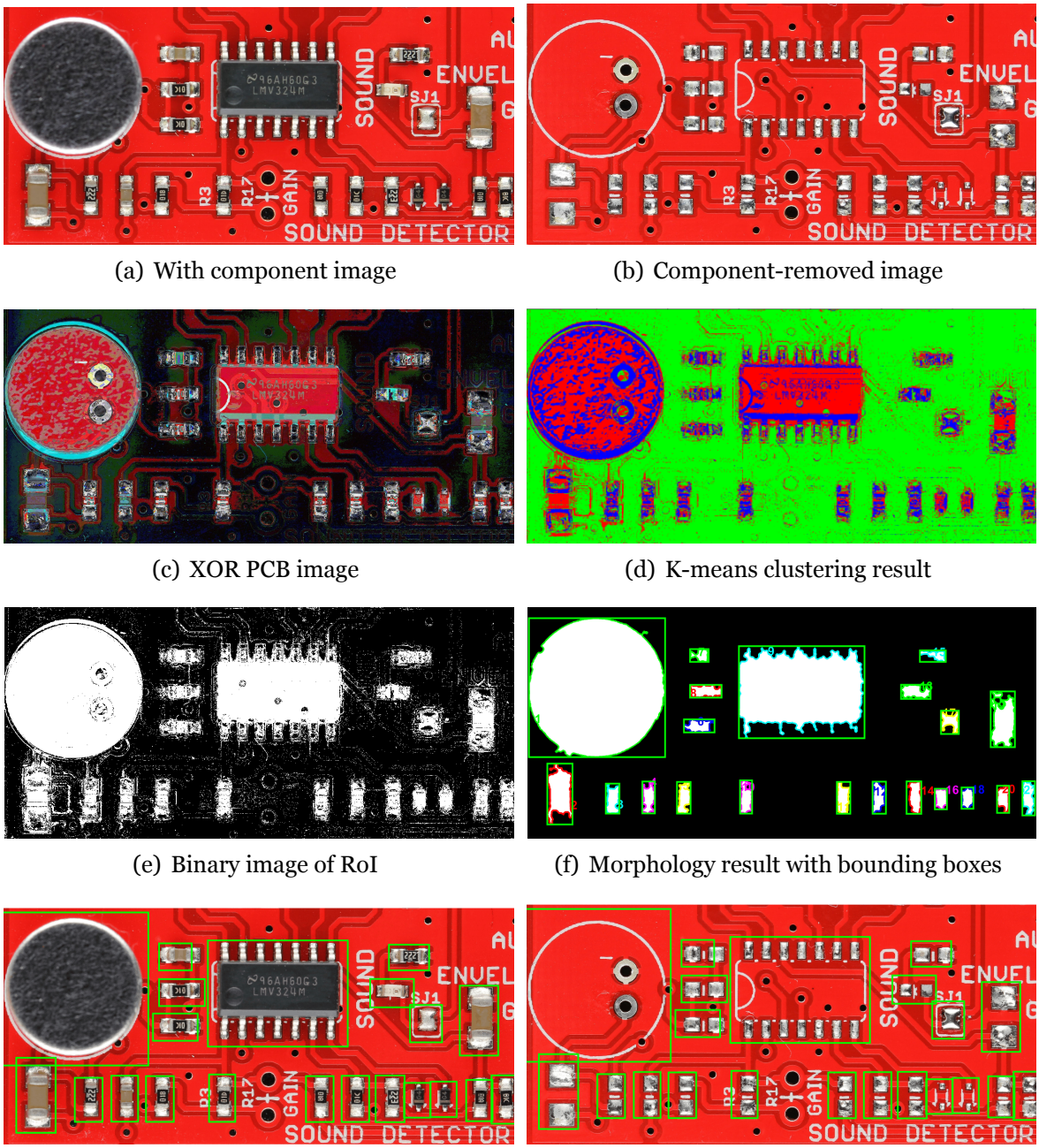


Figure 5.1.6: Unanotated components are similar to Capacitor(green box)

5.2 K-means Detection

K-means clustering for PCB XOR images aims to find out the different regions to represent components. While as introduced in subsection 3.4, the performance relies on the quality of image collection condition and alignment. We applied this method to 7 raw PCB images. The detection results are illustrated below.



(a) With component image

(b) Component-removed image

(c) XOR PCB image

(d) K-means clustering result

(e) Binary image of RoI

(f) Morphology result with bounding boxes

(g) Image (a) with bounding boxes

(h) Image (b) with bounding boxes

Figure 5.2.1: Result of K-means detection

Figure 5.2.1(a) and (b) are collected by normal camera and aligned by our image

registration algorithm. The XOR result is as Figure (c) presented, where we can found that component regions have higher energy and can be distinguished with the background (soldermask). Then K-means clustering is applied to the XOR image. In this example, we set $k = 3$, then use the pseudo color mapping to visualize these clusters as Figure (d).

We choose the first two clusters with the maximum value (red and blue areas in the pseudo color image). Figure (e) is the binary image of target clusters, while it is noisy and dirty. Figure (f) is the de-noise result by using opening and closing morphology operations. Figure 5.2.1 (g) and (h) are the final component area detection results. Figure (h) shows that we can crop the component out to detect the metal pad for the following process.

Table 5.2.1 illustrates the statistic result of 7 PCB models in this project. The false detection consists of wrong detection, missing and incomplete detection. From the result, we can find that the proposed K-means detection works well with good quality XOR images; however, when the lighting condition changes or component shadow cause problem, it could not distinguish neighbouring components, therefore some results are not acceptable like *PCB-2* and *PCB-7*.

Table 5.2.1: Result of K-means Detection

Model	Accuracy	Error Rate
PCB-1	100.0%	11.1%
PCB-2	50.0%	50%
PCB-3	90.0%	10%
PCB-4	100.0%	0%
PCB-5	83.3%	25%
PCB-6	90.4%	9.5%
PCB-7	55.6%	40%

Although K-means detection is fast with low computation cost and high accuracy, it only can find the location of components. We applied a simple CNN as a classifier to detect every cropped component.

To train the CNN classifier, we manually cropped every component out from original PCB images and then labeled with 4 different tags:*Resistor*, *Capacitor*, *IC* and *Metal Pad*. As mentioned before, pre-trained CNN model VGG-16 is utilized for transfer learning. Table 5.2.2 shows the result of our CNN classifier for component detection.

Table 5.2.2: Result of CNN Classification for Cropped Components

Class	Accuracy	Precision
Resistor	97.8%	93.3%
Capacitor	98.4%	94.3%
IC	93.2%	89.4%
Metal Pad	95.7%	90.5%
Average	96.5%	91.9%

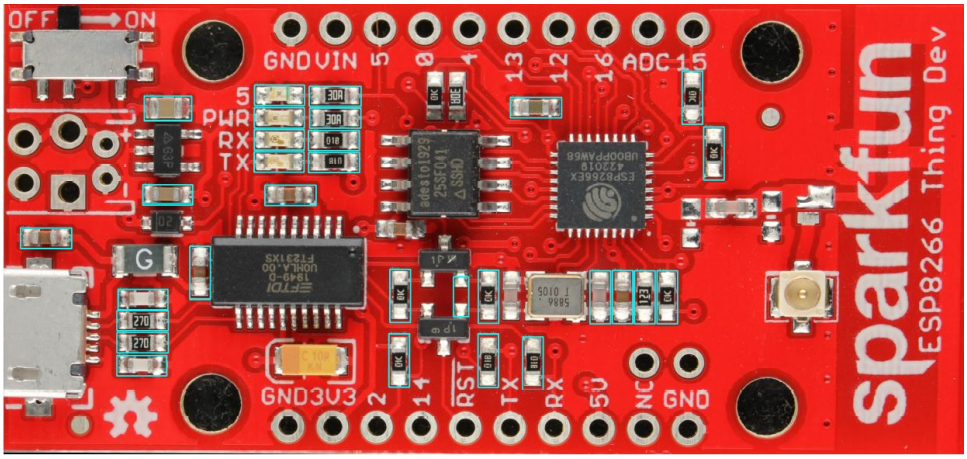


Figure 5.3.1: Example of Multi-template Matching

As Table 5.2.1 and 5.2.2 presented, K-means detection and CNN classification have a higher score in accuracy compared with Faster R-CNN. While for K-means clustering, there are many parameters needed manually assistance for specific tasks, current algorithm could not process with different cases automatically.

5.3 Multi-template Matching

Multi-template Matching is a complement step for previous component detection processes; usually, it gets templates of earlier results. As Figure 5.3.1 shown, this matching method could detect similar components, while for others, it does not work. In particular, the lighting environment may cause a shadow problem. The template correlation mapping is then impaired; besides, two metal pads with a soldermask may be detected as a resistor or capacitor. In this example, we only input resistor and capacitor templates while there is some missed and error detection, as we can see in the picture.

Table 5.3.1 presents the result of multi-template matching for our 7 PCB models. It lists the number of templates, relative components (not including IC package),

correct and false detection (including wrong and missing detection). As we can see from the figures of accuracy and error rates, multi-template matching shows a robust and reliable performance (more than 88% successful rate) on resistor and capacitor detection. These components repeatedly appear on a PCB. While the drawback of this method is that templates are need created and inputted manually. Therefore, multi-template matching is a complement solution for component detection to guarantee the following circuit extraction.

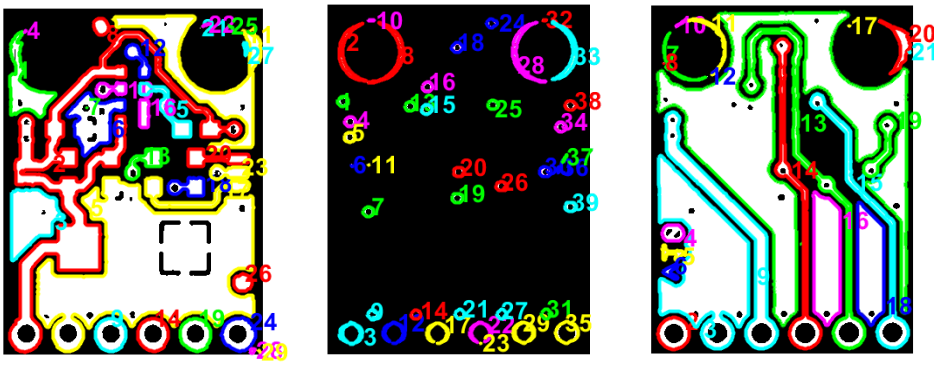
Table 5.3.1: Result of Multi-template Matching

Model	Templates	Accuracy	Error Rate
PCB-1	4	100.0%	0%
PCB-2	4	100.0%	0%
PCB-3	5	90.9%	9.1%
PCB-4	4	89.5%	15.8%
PCB-5	6	86.7%	13.3%
PCB-6	7	88.5%	19.2%
PCB-7	5	93.8%	9.4%

5.4 Inner Circuit Connection

After multi-channel image enhancement and metal segmentation, as described in 3.6, we can utilize every layer's binary image to explore the connection between adjacent layers.

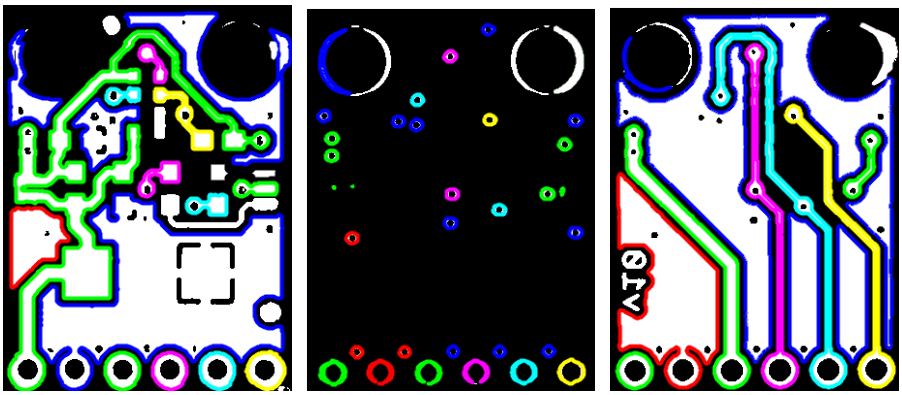
As Figure 5.4.1 illustrated, isolated conductive areas are marked with a label (number) and visualized in different colors. In other images, they have irrelevant labels. It is the so-called "Shape Identification." Then overlapping detection between every adjacent layer's shapes is applied. If overlap exited, it means these two shapes are connected.



(a) Layer 1 (b) Layer 2 (c) Layer 3

Figure 5.4.1: Result of Shape Identification

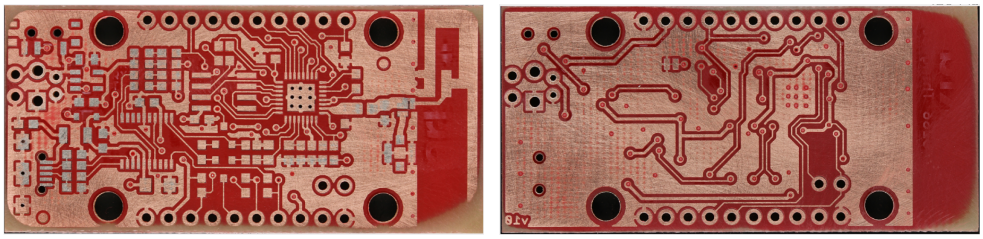
Figure 5.4.2 shows the connection of PCB inner circuit. Connected shapes in all layers are treated as on a circuit net. We group them by detecting every shape's connection. If conductive shapes are connected, they will be labeled with the same group number and color in different layer figures.



(a) Layer 1 (b) Layer 2 (c) Layer 3

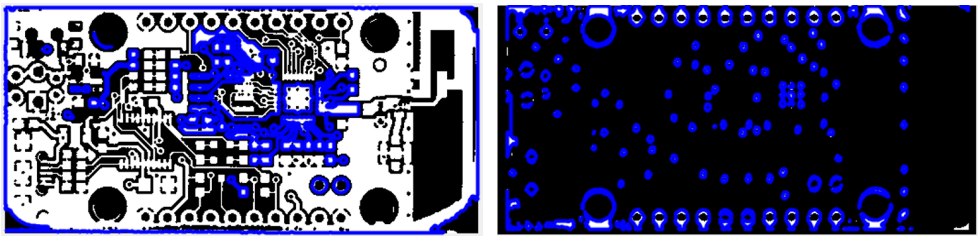
Figure 5.4.2: Result of Multi-layer Connection (connected shapes with same color)

With a clean and complete binary image of every PCB layer, our alignment and connection algorithms work pretty well. In contrast, there is much noise and scratch due to physical de-layer processing, sometimes in practice, which causes wrong binary segmentation and error connection. Figure 5.4.3 illustrates one example that abraded surrounding is similar to target copper circuit. The connection is wrong: all shapes are treated as connected and then grouped into one net, as 5.4.4 shown. It is a challenge for both prototyping and image processing. It asks for a high-accurate metal circuit segmentation to reduce the influence of false targets.

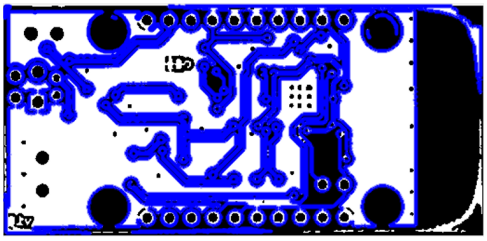


(a) Top Layer (b) Bottom Layer

Figure 5.4.3: Noisy de-layer images (surrounding scratch)



(a) Layer 1 (b) Layer 2



(c) Layer 3

Figure 5.4.4: False connection due to de-layer scratch (all shapes are into one group)

Chapter 6

Conclusions and Future Work

This thesis project aims to extract circuit information from PCB images. We focus on the electronic component detection and inner layers connection in this paper. Advanced object detection methods are applied, a sustainable generative system is proposed in this thesis, which could utilize the result to enhance the training database between two detection modules. This chapter summarizes our work and discusses the valid future work for images based PCB circuit extraction.

6.1 Conclusion

Extracting circuit information from real-world PCB images is a challenge in the electronic manufacturing industry. These images usually contain various components and a lot of scratch by physical composition. This thesis project utilizes state-of-art machine learning methods and classical image processing techniques to build a feasible and reliable analysis system. The work and contributions are in 3 main parts:

1. Design of PCB image processing and data preparation methods.
2. Implementation of Faster R-CNN components detection, XOR-based K-means detection, and multi-template matching.
3. Exploration of a multi-layer PCB inner connection approach.

Instead of simulation data, we use real-world PCB images in this project. Data augmentation is applied to increase more training images for our Faster R-CNN detection and CNN classification. For supervised machine learning, we compared 3

pre-train models and finally chose VGG-16 in our neural networks for transfer learning. The mAP result of Faster R-CNN (71.2%) is not good enough, while compared with other advanced dataset experiment results (78.3%)[22], it is acceptable for our limited data.

In order to improve the detection result and enhance the training dataset, we proposed K-means component detection based on the XOR image. K-means clustering is fast and robust to find the location of components, and then a simple CNN is applied for classification. For the result, the average accuracy of localization is more than 82%, and the accuracy and precision of component classification is 96.5% and 91.9% respectively. In addition, we can add more labeled training data for the Faster R-CNN module and crop component images for the template library.

In this project, we also applied multi-template matching as a complementary solution to ensure all components are taken into circuit analysis, which successes in detecting specific components with 88% accuracy. With the template library enhanced, the missed rate will decrease significantly.

Furthermore, we investigate the inner circuit connection to extract PCB's netlist information. Conductive metal segmentation and multi-channel image fusion are applied to strengthen internal layer information. Then, multi-layer associations are extracted, and it groups these conductive shapes into a netlist.

6.2 Limitation

Our proposed PCB image processing is not automatically adaptive to different image quality, like lighting conditions, background, and component shadow problems. While in real industry cases, it is hard to ensure all the input data are unified in a standard requirement.

There are various components for different PCB manufacture. Currently, we only have limited data for our specific task. The training dataset of machine learning needs to be enhanced with more real-world component images.

In K-means detection and metal segmentation, there are still many parameters requiring manually setting for specific PCB. The performance relies on the XOR image, which needs to collect images in the same condition and accurately pixel-to-pixel

alignment.

6.3 Future Work

Despite the fact that we get some indicative results from our machine learning model, there is no doubt that better data preparation, data augmentation, and model training could provide a better insight into our research. The current dataset size is far from enough for a robust and reliable detection system; Besides *Faster R-CNN*, other advanced object detection algorithms also worth trying, such as *RetineNet* and *YOLOv3*, to make a comparison for the specific PCB detection task.

This thesis is the beginning part of Ericsson's industry project for extracting a complete netlist from PCB images. Component detection results could be added into circuit information, such as the liking component and its pad with a connected inner circuit net. Recognizing the text on a component could extract more detailed information and identify what model it is, enhancing the circuit netlist a lot compared with the general components.

Bibliography

[1] M. Anderberg. *Cluster analysis for applications: probability and mathematical statistics: a series of monographs and textbooks*. Vol. 19. Academic press, 2014.

[2] N. Asadizanjani, M. Tehranipoor, and D. Forte. "PCB reverse engineering using nondestructive X-ray tomography and advanced image processing". In: *IEEE Transactions on Components, Packaging and Manufacturing Technology* 7.2 (2017), pp. 292–299.

[3] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. "Speeded-up robust features (SURF)". In: *Computer vision and image understanding* 110.3 (2008), pp. 346–359.

[4] R. Brunelli. *Template matching techniques in computer vision: theory and practice*. John Wiley & Sons, 2009.

[5] L. K. Cheong, S. A. Suandi, and S. Rahman. "Defects and Components Recognition in Printed Circuit Boards Using Convolutional Neural Network". In: *10th International Conference on Robotics, Vision, Signal Processing and Power Applications*. Springer. 2019, pp. 75–81.

[6] A. Criminisi, I. Reid, and A. Zisserman. "A plane measuring device". In: *Image and Vision Computing* 17.8 (1999), pp. 625–634.

[7] A. T. Erozan, M. Hefenbrock, M. Beigl, J. Aghassi-Hagmann, and M. B. Tahoori. "Reverse engineering of printed electronics circuits: From imaging to netlist extraction". In: *IEEE Transactions on Information Forensics and Security* 15 (2019), pp. 475–486.

[8] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, et al. "A density-based algorithm for discovering clusters in large spatial databases with noise." In: *Kdd*. Vol. 96. 34. 1996, pp. 226–231.

[9] M. A. Fischler and R. C. Bolles. “Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography”. In: *Communications of the ACM* 24.6 (1981), pp. 381–395.

[10] R. Girshick. “Fast r-cnn”. In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 1440–1448.

[11] R. Girshick, J. Donahue, T. Darrell, and J. Malik. “Rich feature hierarchies for accurate object detection and semantic segmentation”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014, pp. 580–587.

[12] G. H. Golub and C. Reinsch. “Singular value decomposition and least squares solutions”. In: *Linear Algebra*. Springer, 1971, pp. 134–151.

[13] J. Grand. “Printed circuit board deconstruction techniques”. In: *8th Workshop on Offensive Technologies*. 2014.

[14] B. Hu and J. Wang. “Detection of PCB Surface Defects With Improved Faster-RCNN and Feature Pyramid Network”. In: *IEEE Access* 8 (2020), pp. 108335–108345.

[15] W. Huang and P. Wei. “A PCB dataset for defects detection and classification”. In: *arXiv preprint arXiv:1901.08204* (2019).

[16] F. Kharfi. *Mathematics and Physics of Computed Tomography (CT): Demonstrations and Practical Examples*. InTech, 2013.

[17] K. Krishna and M. N. Murty. “Genetic K-means algorithm”. In: *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 29.3 (1999), pp. 433–439.

[18] A. Krizhevsky, I. Sutskever, and G. E. Hinton. “Imagenet classification with deep convolutional neural networks”. In: *Advances in neural information processing systems*. 2012, pp. 1097–1105.

[19] T. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár. “Focal loss for dense object detection”. In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 2980–2988.

[20] S. Lloyd. “Least squares quantization in PCM”. In: *IEEE transactions on information theory* 28.2 (1982), pp. 129–137.

[21] D. Lowe. “SIFT flow: dense correspondence across difference scenes”. In: *Proc. ICCV*. 1999, pp. 1150–1157.

[22] G. Mahalingam, K. M. Gay, and K. Ricanek. “PCB-METAL: A PCB Image Dataset for Advanced Computer Vision Machine Learning Component Analysis”. In: *2019 16th International Conference on Machine Vision Applications (MVA)*. IEEE. 2019, pp. 1–5.

[23] N.-D. Nguyen, T. Do, T. D. Ngo, and D.-D. Le. “An Evaluation of Deep Learning Methods for Small Object Detection”. In: *Journal of Electrical and Computer Engineering* 2020 (2020).

[24] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. “You only look once: Unified, real-time object detection”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 779–788.

[25] J. Rey. “Faster R-CNN: Down the rabbit hole of modern object detection”. In: *Tyrolabs* (2018).

[26] S. Tang, F. He, X. Huang, and J. Yang. “Online PCB defect detector on A new PCB defect dataset”. In: *arXiv preprint arXiv:1902.06197* (2019).

[27] L. S. Thomas and J. Gehrig. “Multi-template matching: a versatile tool for object-localization in microscopy images”. In: *BMC bioinformatics* 21.1 (2020), pp. 1–8.

[28] R. Torrance and D. James. “The state-of-the-art in semiconductor reverse engineering”. In: *Proceedings of the 48th Design Automation Conference*. 2011, pp. 333–338.

[29] R. Tran Thanhand Wehrens and L. M. Buydens. “KNN-kernel density-based clustering for high-dimensional multivariate data”. In: *Computational Statistics & Data Analysis* 51.2 (2006), pp. 513–525.

[30] J. R. Uijlings, K. E. Van De Sande, T. Gevers, and A. W. Smeulders. “Selective search for object recognition”. In: *International journal of computer vision* 104.2 (2013), pp. 154–171.

[31] D. Wang. *multilayer pcb manufacturing process*. 2020. URL: `https://www.aipcba.com/pcb/multilayer-pcb-manufacturing-process.html`.

[32] W. Wu, M. J. Wang, and C. Liu. “Automated inspection of printed circuit boards through machine vision”. In: *Computers in industry* 28.2 (1996), pp. 103–111.

[33] Y. Yin, H. Luo, J. Sa, and Q. Zhang. “Study and application of improved level set method with prior graph cut in PCB image segmentation”. In: *Circuit World* (2019).

TRITA-EECS-EX-2020:920

www.kth.se